Global Information Assurance Certification Paper

Analyst: Donna M. Andert
GIAC Intrusion Detection Curriculum
Practical Assignment for SNAP San Jose

Table of Contents

Detect #1 Checkpoint GUI - Misfire

May 28 04:16:09 router.net 558651: May 28 04:16:08: %SEC-6-IPACCESSLOGP: list 101 denied tcp client.com(65368) -> firewall.253(257), 1 packet
May 28 04:21:35 router.net 558667: May 28 04:21:34: %SEC-6-IPACCESSLOGP: list 101 denied tcp client.com(65368) -> firewall.253(257), 2 packets
May 28 04:56:26 router.net 558755: May 28 04:56:25: %SEC-6-IPACCESSLOGP: list 101 denied tcp client.com(32799) -> firewall.253(257), 1 packet
May 28 05:01:37 router.net 558768: May 28 05:01:36: %SEC-6-IPACCESSLOGP: list 101 denied tcp client.com(32799) -> firewall.253(257), 2 packets
…several days of similar packets continue, intermittently…

1. Source of Trace:
- My network.

2. Detect was generated by:
- Cisco Access Control List syslog to Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* client.com(35641) *source address and port* -> isp.253(257) *destination address and port* , 1 packet

3. Probability the source address was spoofed.
- Low.
- IP address is from a range registered to one of our clients out of Uunet address block.

4. Description of attack:
- Packets blocked at router, and do not reach Client Firewall.
- Client directs all firewall rules on Firewall.
- My ISP controls all access control lists on *router.net*.
- An East Coast customer location is the registered owner of the source IP address.
- Packets are directed to TCP port 257, an unknown port. It was initially unclear from persistent traffic whether there was an automated attack attempt, just waiting for a careless engineer to remove an ACL "too long" during maintenance so a few packets can slip through, or just a misconfigured workstation.
- Follow up with customer support engineer. The source IP is a Firewall-1 box whose final configuration has been delayed for other network issues.

5. Attack mechanism:

- Setup Firewall-1 v3.0 GUI and run fwpolicy, fwlog, and fwstatus Checkpoint GUI tools. Only port 258 is used. This doesn't match the observed packets. Continue research and find www.phoneboy.com/fw1/ web page that describes port 257 as the port for a firewall to send logs to the management station. The remote firewall is misconfigured to send log data to the wrong management station host. The firewall log traffic will be moved from the Internet to the firewall management station on an internal network.

6. Correlation:
- The "Phoneboy Firewall-1 FAQ" describes Checkpoint port uses: http://www.phoneboy.com/fw1/
- Various firewall mailing list archives contain requests for Checkpoint port information and procedures to turn them off. John Lauderdale requests information about port 257 on list: firewall-wizrds@nfr.net, searchable at securityportal.com
- Checkpoint technical support gives detailed screenshots to use GUI to disable "Firewall-1 Control Connections".
- San Jose Intrusion Detection 2.3 page 247 warns about Cisco access control list disabling and reloading. "…when a new access control list is applied to the router, one of the first steps in the process is to disable the current access list. As new statements are applied, blocks are put in place." If this had been unfriendly traffic, there could have been a risk when router ACLs were reloaded. This traffic will be monitored as well as other persistent traffic to this customer firewall – while traffic is safely blocked today, Cisco texts often advise that stored ACL files contain the command the removes the ACL at the start of file. It appears to be a common, though unsafe practice.

7. Evidence of active targeting:
- Yes. Port 257 traffic is only directed at a central Firewall-1 choke point for traffic. Traffic is benign side effect of a misconfigured firewall configuration.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- 1 = (5+3)-(4+3)
- Comment: Lethality 3 chosen because traffic is inappropriately being sent over the Internet. This reveals the firewall product as Checkpoint.

9. Defensive recommendations:
- Report traffic observations to customer support engineer.
- Firewall was reconfigured to send port 257 traffic over an internal network instead of the public Internet.
- No change to rules and logging.

10. Multiple choice test question:

What can be concluded from this trace?
A. spoofed source
B. stealthy reconnaissance
C. destination address could be a broadcast address
D. none of the above
Answer: D

Detect #2 Linuxconf – Hostile Fire

May 31 00:12:21 router.net 571153: May 31 00:12:20: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(3742) -> isp.65.40(98), 1 packet
May 31 00:12:24 router.net 571154: May 31 00:12:23: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(4647) -> isp.65.107(98), 1 packet
May 31 00:12:24 router.net 571155: May 31 00:12:24: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(3855) -> isp.65.153(98), 1 packet

May 31 00:12:27 router.net 571156: May 31 00:12:26: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(4647) -> isp.66.107(98), 1 packet

…snoop - unimportant ethernet header information removed for space…
ETHER: ----- Ether Header -----
ETHER: Packet 1 arrived at 0:16:22.82
IP: ----- IP Header -----
IP: Version = 4
IP: Header length = 20 bytes
IP: Type of service = 0x00
IP: Total length = 60 bytes
IP: Identification = 22476
IP: Flags = 0x4
IP: .1.. .... = do not fragment
IP: ..0. .... = last fragment
IP: Fragment offset = 0 bytes
IP: Time to live = 50 seconds/hops
IP: Protocol = 6 (TCP)
IP: Header checksum = 1864
IP: Source address = 210.112.192.74, 210.112.192.74
IP: Destination address = isp.home.net
IP: No options
TCP: ----- TCP Header -----
TCP: Source port = 2634
TCP: Destination port = 98
TCP: Sequence number = 3123942852
TCP: Acknowledgement number = 0
TCP: Data offset = 40 bytes
TCP: Flags = 0x02
TCP: ..0. .... = No urgent pointer
TCP: ...0 .... = No acknowledgement
TCP: .... 0... = No push
TCP: .... .0.. = No reset
TCP: .... ..1. = Syn
TCP: ....0 = No Fin
TCP: Window = 32120
TCP: Checksum = 0x7856
TCP: Urgent pointer = 0
TCP: Options: (20 bytes)
TCP: - Maximum segment size = 1460 bytes
TCP: - Option 4 (unknown - 0 bytes)
TCP: - Option 8 (unknown - 8 bytes) 043B1ACD00000000
TCP: - No operation
TCP: - Option 3 (unknown - 1 bytes) 00

May 31 11:34:58 router.net 572850: May 31 11:34:57: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(4737) -> isp.65.11(98), 1 packet
May 31 11:35:00 router.net 572851: May 31 11:34:59: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(1752) -> isp.66.106(98), 1 packet
May 31 11:35:00 router.net 572852: May 31 11:35:00: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(4753) -> isp.65.27(98), 1 packet
May 31 11:35:03 router.net 572854: May 31 11:35:02: %SEC-6-IPACCESSLOGP: list 101 denied tcp 210.112.192.74(1751) -> isp.66.105(98), 1 packet

1. Source of Trace:
- My network.

2. Detect was generated by:
- Cisco Access Control List logged to syslog on Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* client.com(35641) *source address and port* -> isp.253(257) *destination address and port* , 1 packet

- Black Ice Defender on Windows98, converted through Ethereal and displayed with Sun snoop (out of home DSL out of another class C within my same ISP)

3. Probability the source address was spoofed.
- Low. Attacker needs to receive reconnaissance results.
- Attacking host may be compromised.
- IP address owners LG Internet, in Korea, are beyond reasonable intervention.

4. Description of attack:
- Attacker probed port 98 TCP, known as linuxconf, at several IP addresses, two sets into my ISP's engineering network, and one into my ISP's work-from-home network.
- Technique: The technique is either a TCP Connect scan or a TCP SYN scan.
- Passive Fingerprinting Analysis: SYN, window size 32120, don't fragment, TTL 50, traceroute to attacker is 13 hops. Therefore, if this data is accurate, a window size of 32120 and an original TTL of (50+13) 63 suggests the attacker was using the Linux OS. http://www.enteract.com/%7Elspitz/traces.txt
- Passive Fingerprinting Analysis: SackOK set is consistent with Linux operating system. http://www.enteract.com/%7Elspitz/finger.html

5. Attack mechanism:
- Reconnaissance - The SYN scan sends a single packet with the SYN bit sent. If a process is listening on that port, a SYN-ACK would be returned to the source IP, confirming the server and port. If a Reset is returned then a process is not listening on that port.
- The reconnaissance scan technique could be either TCP Connect or TCP SYN, both send an initial SYN.
- Once a host listening on port 98 is identified, the attack can be attempted. Linuxconf is a web-enabled administration utility.

Several Linuxconf security issues:
- A somewhat recent report in the Linux Weekly News contains the Linuxconf author's response to rumors of a security issue. Jacques Gelinas reports that if Linuxconf has been configured to allow a network or host access, then perhaps there could be a problem parsing an http request, but only by the allowed network or hosts. He says he cannot do anything useful with the security weakness. http://lwn.net/1999/1223/a/linuxconfresponse.html
- Robert Graham's Firewall Forensics FAQ refers to a buffer overflow in the LANG environment variable. http://www.robertgraham.com/pubs/firewall-seen.html
- One 1998 version was setuid root. http://lwn.net/980604/a/linuxconf/html

6. Correlation:
- Exact Correlation Hit - June 1, 2000 email to incidents@securityfocus.com by infrastructure@narellan.net describe a similar linuxconf scan, from the SAME IP on the same day. They did not post their Black Ice evidence data, however.
- Exact Correlation Hit - Many GIAC analysts reported linuxconf scans with the exact same source IP on their networks – this is just one. http://www.sans.org/y2k/060300.htm

7. Evidence of active targeting:
- Yes. A single packet was sent to each address. The IPs showed some knowledge of internal servers, beyond coincidence for 4 random packets.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- -1 = (5+1)-(4+3)

9. Defensive recommendations:
- Periodically check for recent activity by this IP address.
- Continue periodic vulnerability testing of packet filtering rules.
- Begin periodic vulnerability testing of Linux servers.
- Begin periodic scans to identify new servers brought online within engineering.

10. Multiple choice test question:

What operating system is the attacker hoping to exploit?
A. SGI
B. Solaris
C. FreeBSD
D. Linux
Answer: D

Detect #3 SubSeven or Backdoor-G – Hostile Fire

Jun 3 03:15:52 router.net 582506: Jun 3 03:15:51: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.6.34.49(21059) -> firewall.253(27374), 1 packet
Jun 3 03:15:55 router.net 582507: Jun 3 03:15:54: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.6.34.49(21062) -> firewall.253(6700), 1 packet
Jun 3 03:20:10 router.net 582520: Jun 3 03:20:09: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.6.34.49(21062) -> firewall.253(6700), 2 packets
Jun 3 03:20:58 router.net 582521: Jun 3 03:20:57: %SEC-6-IPACCESSLOGP: list 101 denied tcp 216.6.34.49(21063) -> firewall.253(1243), 2 packets

1. Source of Trace:
- My network.

2. Detect was generated by:
- Cisco Access Control List syslog to Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* 10.8.31.251(35641) *source address and port* -> isp.253(257) *destination address and port* , 1 packet

3. Probability the source address was spoofed.
- Low. Attacker needs to see reconnaissance results.
- Attacking host may be compromised.

4. Description of attack:
- The Attacker knows the firewall server IP address already, and is probing for the SubSeven trojan applications, old and new.
- Since packet 2 and 3 are separated by 5 minutes and yet have the same source port, I conclude they are part of a crafted and targeted port scan.
- There were several other SubSeven probes from other IP addresses, but each probed a different set of ports, destination IP addresses, and different "extra" non-SubSeven ports so these packets are judged to be separate attackers.

5. Attack mechanism:

- SubSeven has multiple components, some with configurable names, as well as configurable ports. It is initially started from the Win.ini file or the System.ini file. Once the attacker makes contact with the Trojan, it can be controlled through several configurable ports. From the SubSeven author's feature descriptions, it can be used to perform a range of silly to destructive pranks as well as industrial espionage. If it could be installed on the desktop of a key marketing person, imagine how much information could be extracted from AOL Instant Messenger Spy, Yahoo Messenger Spy, Microsoft Messenger Spy, IRC Connection Notify, ICQ Spy, Full Screen Capture, Download Files, and List visible windows.
- Unknown port 6700 – DAQV research application used somewhere within University of Oregon. http://www.cs.uoregon.edu/~hacks/research/daqv/doc/proc.html
- Unknown port 6700 - A server application called "Carracho". http://www.tracker-tracker.com/ubb/Forum5/HTML/000007.html
- Unknown port 6700 – Possibly a configurable backdoor port, or the newest flavor of SubSeven in beta test…
- A description of the SubSeven application with screen shots of the user interface can be found at: http://www.commodon.com/threat/threat-sub7.htm
- The author, mobman, evidently has his own website where he "advertises" the features of SubSeven, and brags about soon to be released features. http://subseven.slak.org

6. Correlation:
- In the last two weeks, SubSeven scans to my ISP have increased, including the accompanying scan for port 6700.
- SubSeven scan reported to GIAC. http://www.sans.org/y2k/052900.htm
- SubSeven probes from probable cable modems to GIAC. http://www.sans.org/y2k/053000.htm
- Port 6700 probe on 12/31/99 at GIAC. http://www.sans.org/y2k/123199-1220.htm
- 7/6/99 ISS Security Alert titled "Windows Backdoor Update III" warns about SubSeven and describes it as very configurable. http://www.infowar.com
- 2/25/00 report of SubSeven scan with log excerpt. http://www.health.ufl.edu/wss/mail-archives/unix-sec/2000//02//msg00007.html
- 2/29/00 report in www.geocrawler.com security FW-1 mailing list of port 27374 scans. http://www.geocrawler.com
- 3/21/00 online summary of presentation by Terry Maher to Carlsbad Chamber of Commerce describing security threats, firewall logs, trojan horse probes, and a description of SubSeven. http://www.carlsbad.org/tech-security.htm

7. Evidence of active targeting:
- Yes. Only 4 packets are sent, to a firewall host. In general this host receives more individually targeted packets to trojan ports, dns, ftp, or others, so I think it's known as a firewall on some non-legitimate website. But this SubSeven attempt means that they mistakenly think it might be a firewall on NT, which it is not.
- There were NO SubSeven probes seen with BlackIce on my home computer's subnet, another indication that the attacker had a narrow target.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- -2 = (5+0)-(4+3)

9. Defensive recommendations:
- Begin periodic vulnerability testing of Windows servers in this subnet and subnets with trust relationships to this subnet.
- Begin periodic scans to identify new Windows servers brought online within engineering.

10. Multiple choice test question:

What can be concluded from the trace?
A. The attacker's source computer is a shared multiuser operating system.

B. The attacker's source IP address is probably spoofed.
C. The attacker is planning a future denial of service attack.
D. Reconnaissance of destination host was unsuccessful this time.
Answer: D

Detect #4 NetBus – Hostile Fire

May 26 21:24:05 router.net 554058: May 26 21:24:04: %SEC-6-IPACCESSLOGP: list 101 denied tcp 207.40.92.160(1385) -> isp.1(12345), 1 packet
May 26 21:24:07 router.net 554059: May 26 21:24:06: %SEC-6-IPACCESSLOGP: list 101 denied tcp 207.40.92.160(1388) -> isp.2(12345), 1 packet
May 26 21:29:39 router.net 554068: May 26 21:29:38: %SEC-6-IPACCESSLOGP: list 101 denied tcp 207.40.92.160(1395) -> isp.3(80), 1 packet
…24 additional packets not shown…

1. Source of Trace:
- My network.

2. Detect was generated by:
- Cisco Access Control List syslog to Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* 10.8.31.251(35641) *source address and port* -> isp.net(257) *destination address and port* , 1 packet

3. Probability the source address was spoofed.
- Low. Attacker needs to see the reconnaissance results.
- Attacking host may be compromised.

4. Description of attack:
- 27 IP addresses were targeted.
- Packets arrived within 1-2 seconds.
- The fourth octet of the destination IP address was randomized, but only contained IP addresses of real hosts on the subnet.
- One desktop host was probed with port 80 instead of 12345.
- This is a targeted scan for trojan applications on known live hosts.

5. Attack mechanism:
- NetBus is a "remote admin" trojan that runs on Windows NT or Win95/98. The server runs on the victim host, and the client runs on the attacker controlled host. The NetBus server program is usually installed by being sent to the victim or being published on a website as some kind of useful application. It can also be included as part of the setup package for an actual application program. The attacker knows they have placed it on a server, or fooled a victim to download the program, or they spend time mapping and searching for an infected host using the default ports and passwords. Once the NetBus trojan is installed, any attacker who knows the listening port number and password can remotely control the host. Some features of NetBus include: shutdown Windows, get a screenshot, show windows, and download and delete any file.
- In this case, we had an concurrent issue with NetBus ongoing. A coincidental vulnerability assessment that zealously scanned the customer's address space located the NetBus trojan. The customer had reported the departure of a staff member. It is suspicious that a targeted scan was done for the recently found NetBus.

6. Correlation:
There are frequent reports of port scans for the popular trojan applications, including NetBus, at GIAC. Just a few are:
- NetBus scan reported to GIAC. http://www.sans.org/y2k/052900.htm

- NetBus probe reported to GIAC. http://www.sans.org/y2k/053000.htm
- NetBus probes reported to GIAC. http://www.sans.org/y2k/060600.htm
- The NetBus detect is well known. Additional technical details about the NetBus trojan can be researched at: http://www.nwi.net/~pchelp/nb/netbus.htm
- http://www.datafellows.com/v-descs/netbus.htm
- http://www.pspl.com/trojan_info/win32/netbus.htm
- http://www.irchelp.org/irchelp/security/netbus.html

7. Evidence of active targeting:
- Yes. This is a highly targeted scan looking for the Trojan NetBus.
- Scary Fact #1 – The subnet must have been previously mapped because unused IP addresses are skipped in this scan.
- Scary Fact #2 – NetBus was recently, April 2000, found on a client's network that has some shared network connectivity with our network.
- There were NO NetBus probes seen with BlackIce on my home computer's subnet, another indication that the attacker had a narrow target.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- -2 = (5+0)-(4+3)

9. Defensive recommendations:
- Audit for expected host mapping vulnerabilities from Internet into firewall, and close them if possible.
- Begin periodic vulnerability testing of servers of all platform types in this subnet and subnets with trust relationships to this subnet, including work from home subnet.
- Institute security procedures and training for all engineering personnel when installing or reinstalling internal servers.

10. Multiple choice test question:
What is happening in this trace?
A. Attacker is looking for a vulnerable web server.
B. Attacker is mapping hosts with destination port 12345
C. Attacker is probing for Trojans.
D. None of the above.
Answer: C

Detect #5 POP2 – Hostile Fire

May 23 07:14:23 router.net 537040: May 23 07:14:22: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.3(109), 1 packet
May 23 08:41:05 router.net 537349: May 23 08:41:04: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.7(109), 1 packet
May 23 09:02:46 router.net 537432: May 23 09:02:45: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.8(109), 1 packet
……almost every IP address……
May 24 15:23:48 router.net 544075: May 24 15:23:47: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.92(109), 1 packet
May 24 15:45:29 router.net 544148: May 24 15:45:28: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.93(109), 1 packet
May 24 16:07:09 router.net 544227: May 24 16:07:08: %SEC-6-IPACCESSLOGP: list 101 denied tcp 21.118.8.50(0) -> isp.94(109), 1 packet
……mapper stops suddenly around the time when an undisguised traceroute is executed from one of these hosts.

1. Source of Trace:
- My network.

2. Detect was generated by:
- Cisco Access Control List syslog to Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* 10.8.31.251(35641) *source address and port* -> isp.net(257) *destination address and port* , 1 packet

3. Probability the source address was spoofed.
- Low. Attacker needs to see the reconnaissance results.
- Source IP address is either the attacker's real host or a host (s)he prefers to continue using since the mapping activity stopped when a traceroute was done. The source address space is military.

4. Description of attack:
- Most IP addresses in the subnet were targeted, sequentially.
- The packets arrived at random intervals more than 15 minutes apart so they were not visibly obvious on the 10 minute syslog reports.
- The source port number of 0 is clearly crafted.
- When an undisguised traceroute to the attacker was performed from the same targeted subnet, the attacker stopped his reconnaissance.
- The attacker source IP address is allocated to the military.

5. Attack mechanism:
- Most problems with pop2 are buffer overflow vulnerabilities.
- Phrack issue 49 describes how buffer overflows work. A vulnerable application creates dynamic variables at runtime and writes trusting code that doesn't check the bounds very well. Using some normal data input method, the attacker creates data that the bounds checking, if any, can't handle. The program writes too much data into the buffer variable so that the return address of a procedure call is overwritten with the attacker's assembly code. Now, instead of using the normal return address, the program is executing the attacker's code. The attacker's code usually spawns a shell process. The spawned shell process will be owned by the same user id that owns this process, preferably root. And that's why the security recommendations for Unix systems are always trying to find a way for a process to run as anything but root, to contain future compromises as narrowly as possible. The gory programming details of developing a new Linux buffer overflow exploit are described in: http://phrack.infonexus.com/search.phtml?view&article=p49-14
- Here's my assessment of the attacker skill level and profile. If an exploit is published, then a person with moderate system level programming skills can read, understand, and execute the exploit on a correctly targeted host. The skills are knowledge of the operating system, C programming, and a little internetworking experience – assembly language isn't really needed to use a published exploit. To create a "new" exploit, the successful exploit author has expert level knowledge of the operating details, especially the virtual memory map of a process, and the native assembly language. The exploit author only needs moderate C programming skills, but possesses deep understanding of what's happening at the assembly language level of the simple code he creates. The exploit author must be highly motivated to create the attack and has their own operating system setup at home to try this out, hour after hour. If there is a buffer overflow vulnerability, it isn't hard to corrupt the stack and cause the program to crash, creating a denial of service. But the exploit author really wants to find a way to achieve root privilege without detection. That's the real goal.

6. Correlation:
- Exact Correlation Hit – Reported to GIAC with a detect date within 24 hours of the detect on our network. http://www.sans.org/y2k/052700.htm

- A recent June 7, 2000 post by Fredrik Ostergren to incidents@securityfocus.com states that the recent rise in pop2 scans are looking for popv4.46 and v3.44 that shipped with Red Hat 5.2, but I could not find the information on the security focus web site.
- The pop2 port is rarely used anymore. But if it's found, it's probably a poorly maintained site with few trained staff members. Problems with pop servers are #9 on the SANS Top Ten list. http://www.sans.org/topten.htm
- In the BUGTRAQ Admin forum, a message makes recommendations and posts code examples to protect against bad strings being sent to an open port before login and password authentication is done. http://msgs.securepoint.com/cgi-bin/get/bugtraq9904/162.html
- An old rootshell message warns against versions of popper and qpopper from Qualcomm that can be manipulated to allow the attacker to read someone else's email, a confidentiality issue only. http://rootshell.com/archive-j457nxiqi3gq59dv/199709/popper.txt.html
- In 1998, another user posts code that gives root access through version 2 qpopper using a buffer exploit. http://rootshell.com/archive-j457nxiqi3gq59dv/199806/qpop.c.html

7. Evidence of active targeting:
- No. The scanner tried almost sequential IP addresses indicating little knowledge of the network. With the GIAC report from the exact same source IP, also to port 109, it is clear the attacker is scanning widely for port 109. He hasn't chosen a target "yet".

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- -2 = (5+0)-(3+4)

9. Defensive recommendations:
- Scan was blocked successfully.
- Improve periodic vulnerability testing and training to prevent future buffer overflows attacks.
- Add host based intrusion detection on key servers.

10. Multiple choice test question:

What is happening in this trace?
A. Denial of Service on port 109
B. Use of old style broadcast 0 to crash destination IP
C. Port scan using port 109
D. Host scan using tcp pop2 port
Answer: D

Detect #6 SNMP – Misfire/Unknown

…initial packet from this source IP address…full trace is 27 packets over 14 minutes
May 14 15:59:46 irv1-bbr1.intelenet.net 500603: May 14 15:59:45: %SEC-6-IPACCESSLOGP: list 101 denied udp 63.210.114.55(1240) -> 207.38.65.192(161), 1 packet
…initial packet from this source IP address…full trace is 72 packets over 31 minutes
May 14 18:33:50 irv1-bbr1.intelenet.net 500990: May 14 18:33:49: %SEC-6-IPACCESSLOGP: list 101 denied udp 209.245.68.87(1945) -> 207.38.65.192(161), 1 packet
…initial packet from this source IP address…full trace is 1954 packets over 11 hours
May 14 19:56:59 irv1-bbr1.intelenet.net 501260: May 14 19:56:58: %SEC-6-IPACCESSLOGP: list 101 denied udp 63.208.243.140(2344) -> 207.38.65.192(161), 1 packet

1. Source of Trace:
My network.

2. Detect was generated by:

- Cisco Access Control List syslog to Unix server.

May 17 15:30:13 *timestamp* router.net *fully qualified host name* 514475 *packet number with respect to the router* :
May 17 15:30:12: %SEC-6-IPACCESSLOGP: list 101 *router ACL that logged packet* denied *ACL filter action* tcp *protocol* 10.8.31.251(35641) *source address and port* -> isp.net(257) *destination address and port* , 1 packet

3. Probability the source address was spoofed.
- Low if payload is normal SNMP or a misconfigured host.
- High if the actual payload contains lethal one way payload aimed at HP 5M/5N printers.

4. Description of attack:
- Since April, there have been unsubstantiated reports that the HP printer is broken. The "broken" reports could just be bad postscript or a power event. The HP printer was purchased in 1997 or earlier.
- UDP SNMP traffic is targeted at an IP address that is an HP printer. There is no logging or packet sniffing available yet to confirm stimulus response. And the payload is unknown.
- The source IP address changes.
- All apparent source IP addresses are Los Angeles Level 3 addresses.

5. Attack mechanism:
- A September 1998 message to BUGTRAQ describes a way to crash HP printers with a single legitimate SNMP packet. http://www.attrition.org/security/denial/w/hp5-snmp.dos.html
- The following snmpgetnext command can direct a packet that crashed HP printers in 1998.

$ snmpget printername public 43.15.1.1.4.1.1
…or
$ npadmin –languages printername
…or
$ npadmin –protocol printername

- The printer is often left with error message 79 but is still pingable. It may accept one more print job but never print it. Sometimes the printer is left unpingable. The problem is a bug in the printer firmware, called the formwatter.
- A persistent attack, or one that needs only a few delivered packets to succeed, can be successful during the small time window when a trusting engineer removes a Cisco ACL and reloads it. In fact, the attacker's first goal could be to induce an engineer to add his annoying source address to the Cisco ACL. And previously ignored annoying traffic with destructive payload could slip through.

6. Correlation:
- Issue 50 of Phrack warns against SNMPv1 insecurities such as easily guessable community strings and in the clear community strings that can be packet sniffed. It may be possible to learn some information about the target network by examining SNMP data. http://www.2600.net/phrack/p50-07.html
- San Jose SANS Intrusion Detection 2.4/2.5 page 287 describes using SNMP to scan for printers and other information.
- San Jose SANS Intrusion Detection 2.3 page 247 warns about access control list disabling and reloading.
- A user warns about similar problems with SNMP for APC products, a vendor of backup power products. http://www.rewted.org/mailing-lists/bugtraq/11.98/0023.html

7. Evidence of active targeting:
- Yes. Whether hostile or not, the traffic is targeted at a single address.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- -4 = (1+2)-(2+5)

9. Defensive recommendations:
- Check the community string on the printer and make it difficult to guess.

- Modify router log monitoring to send an alarm when SNMP traffic is sent to the printer from the outside.
- Setup equipment to get a packet trace of this event and confirm attack or configuration issue.
- Check whether there are firmware upgrades that should be performed for this printer.

10. Multiple choice test question:

Is the source IP address spoofed?
A. Definitely yes.
B. Definitely no.
C. Need to look at the entire packet payload.
D. It is not possible to know if a source IP address in a packet is spoofed, by looking at the packet data.
Answer: D

Detect #7 statd – Hostile Fire

Host: isp.net
Date: Tue May 30 14:15:05 2000
File: /var/adm/messages; 28 new line(s), 26 filtered
May 30 14:08:07 isp.net ftpd[21340]: FTP LOGIN REFUSED (bad shell) FROM ppp220002.fx.ro [193.231.220.2], root
May 30 14:13:52 isp.net statd[583]: statd: attempt to create "/var/statmon/sm//../../../tmp/CyberCop.rpc.statd.vulnerability"

Additional messages in /var/adm/messages that scripts did not identify.
May 30 14:02:01 irv1-web1.intelenet.net sshd[19233]: fatal: Local: Sorry, you are not allowed to connect.
May 30 14:04:48 irv1-web1.intelenet.net sshd[20260]: fatal: Local: Sorry, you are not allowed to connect.

1. Source of Trace:
- My network.
- Host outside perimeter defenses.

2. Detect was generated by:
- syslog messages in /var/adm/messages on Unix server.

3. Probability the source address was spoofed.
- Low. Attacker is attempting to compromise server.
- Source IP address is probably a previously compromised host.
- Logged port 111 scans on other subnets were from other IP addresses, with nothing from the .ro domain. Attackers are known to use multiple hosts to evade detection. This makes meaningful fusion of logs more difficult.

4. Description of attack:
- Attacker probably scanned our subnets for hosts with an open port 111, but there is no network traffic logging on this subnet. Scans for port 111 occur weekly or daily on other subnets with logging.
- Attacker attempts to login with sshd, twice, from the same host. Source host IP address is not on the allowed host list.
- Attacker tries to ftp, possibly to an account that is setup without any login or ftp access which generates the “bad shell” log message.
- From a different host with Cybercop (available over the Internet for trial use) loaded, attacker checks for statd vulnerability.
- /var/statmon/sm has a modify time of May 30 14:09.

- Logging anomaly was noted and investigated by system administrator. Multiple on-host and off-host logging archives show no alteration of logfiles or other anomalies. System administrator turns off statd and takes further preventive measures on host. And plans to put additional hosts behind a firewall are accelerated.

5. Attack mechanism:

- The "portmap" attack on port 111 is aimed at Solaris Unix systems from Sun Microsystems. In the 80s, remote procedure call mechanisms were the primary means of building client server systems. Most of these systems were isolated networks within aerospace or large corporations, and they did NOT have Internet access. The architecture reasoning goes like this. A server process could be started up at any time, and listen on any port. The client prefers to ask for the service by name or number, and depends on the "portmap" service to report the port that the server is listening on or forward the request to the appropriate port. The "portmap" service can be queried on a Solaris Unix system to find out which ports key server processes are listening on. But even if "portmap' is not running, the attacker always has the option to port scan to locate the processes. Chapman and Zwicky's "Building Internet Firewalls", has a nice summary of RPC issues within the context of firewalls, page 148.
- The goal of exploiting the rpc.statd service is to be able to execute an arbitrary command with the same privileges as the owner of the rpc.statd process, which is often root or some other semi-privileged user. The general buffer overflow technique is described in Detect #5.
- A persistent attacker will leverage the ability to execute commands as any non-root user and combine this with other user level vulnerabilities to eventually gain root access. Bruce Schneier of Counterpane Internet Security uses the term "attack trees" to describe a risk analysis methodology that can be used to focus $$$ on the valuable assets or to model a chain of vulnerabilities that result in compromised data. http://www.counterpane.com/attacktrees-ddj-ft.html

6. Correlation:

- This site has periodic scans for port 111 on other monitored subnets.
- This attack is #3 on the SANS Top Ten list. A sampling of portmap correlations are listed below. http://www.sans.org/topten
- Several portmap scans reported to GIAC. http://www.sans.org/y2k/051800.htm
- Portmap scan with SYN reported to GIAC. http://www.sans.org/y2k/051900.htm
- Portmap scan with SYN/FIN reported to GIAC. http://www.sans.org/y2k/052000.htm
- Portmap scans with PUSH/ACK reported to GIAC. http://www.sans.org/y2k/052600.htm
- Portmap scan and connect to dump() reported to GIAC. http://www.sans.org/y2k/052800.htm
- Portmap attempts with getport and dump() reported to GIAC. http://www.sans.org/y2k/053000.htm

7. Evidence of active targeting:

- Yes. There have been periodic scans for port 111 on the two subnets with network logging. And a portmap attack is a good choice for our Solaris centric infrastructure.

8. Severity:

- Severity = (critical + lethal) – (system + net countermeasures)
- 3 = (3+5)-(4+1)

9. Defensive recommendations:

- Although the host complies with Cert Advisory CA-99-05, a further minor patch revision from 106592-02 to –03 is preferable. The current recommended set of security patches from Sun Microsystems should be installed.
- Increase installation frequency of security patches.
- Add logging firewall between DMZ hosts and the Internet.
- Add host based intrusion detection system to DMZ hosts.
- Increase frequency of vulnerability testing of DMZ hosts.
- Add periodic fusion of data from DMZ host based logs and network logs on other subnets, contiguous and discontiguous. Look for differences in scans that target contiguous subnets as well as the discontiguous subnet.

These imply that the attacker is attempting to break into our site specifically and has used public sources to footprint our IP address space allocation.

10. Multiple choice test question:
Which service logged the most serious intrusion attempt?
A. ftp
B. ssh
C. statd
D. These are all normal messages, and no further action should be taken.
Answer: C

Detect #8 Zone Transfer – Hostile Fire

May 31 07:24:53 ns.isp.net named[22009]: unapproved AXFR from [211.38.204.93].3646 for "domain.com" (not auth)
…unapproved zone transfer for 9 other domain of the same customer…
Jun 2 05:48:16 ns.isp.net named[22009]: unapproved AXFR from [203.126.241.50].3047 for "domain.com" (not auth)
…unapproved zone transfer for 9 other domains of the same customer…

1. Source of Trace:
- My network.

2. Detect was generated by:
- syslog messages in /var/adm/messages file on DNS server.

3. Probability the source address was spoofed.
- Low. Attacker needs to see the reconnaissance results.
- Source IP address is probably a previously compromised host.

4. Description of attack:
- Attacker requests a DNS zone transfer of 10 domain names that they do not own. Attacker's source IP address is owned by "Soosiro PC Bang" out of Korea. Requests span 16 hours, so it looks "low and slow". The logged denials don't really jump out at the analyst until all of the messages are pulled from the log as opposed to seeing one an hour.
- The zone transfer requests are allowed through the border routers but are not authorized by the DNS configuration of the name server. The illegal request is logged.
- Two days later, the exact 10 domain names, in the exact same order, are requested from an IP address owned by NTUC Thrift and Loan out of Singapore. Requests span 12 hours, "low and slow".
- My guess is that they have an automated script making the zone transfer requests. They looked at the results of May 31st, made some changes to the destination IP address list, tried the shorter list again on June 2nd, and the attempt completed 4 hours sooner.
- It's probably the same attacker working from two compromised hosts in Asia.

5. Attack mechanism:
- A successful zone transfer can contain a large amount of highly accurate information about internal server names and IP addresses.
- The easiest way to request a zone transfer is to use the widely available tool on Unix and NT, nslookup.
- According to the authors of "Hacking Exposed", page 22, the best tool is "axfr", available from ftp://ftp.trinux.org/pub/trinux/tools/netmap/axfr-0.5.2.tar.gz
- When a zone transfer fails, an attacker will return to traditional methods of reconnaissance that are much slower such as scanning for hosts and ports from previously compromised hosts.

6. Correlation:

- Attempting an unauthorized zone transfer seldom works, since most sites prevent it. However, it is often attempted because it can deliver highly accurate information about an organization's internal network. Even a previously well maintained site can become unexpectedly vulnerable if experienced staff leave and new engineers don't know how to secure DNS. The book, "Hacking Exposed", page 19, describes this technique as: Popularity=9 out of a possible 10.
- GIAC report of unapproved zone transfer on May 17, 2000. http://www.sans.org/y2k/052100.htm
- GIAC reports zone transfers are still popular on May 30, 2000. http://www.sans.org/y2k/053000.htm

7. Evidence of active targeting:

- Yes. Two reconnaissance attempts were attempted to identical series of domain names. The domain is a high profile retailer. No other logged traffic was found from these two attacker IP addresses in the router syslog data.

8. Severity:

- Severity = (critical + lethal) – (system + net countermeasures)
- 1 = (5+3)-(3+4)
- Comment: A compromise of this retailer could be a news event.

9. Defensive recommendations:

- Block zone transfers at the perimeter instead of the host.
- Confirm that the xfernets directive is used in all name server configurations.
- Add logging firewall between DMZ hosts and the Internet.
- Add host based intrusion detection system to DMZ hosts.
- Increase frequency of vulnerability testing of DMZ hosts.

10. Multiple choice test question:

What is happening in this trace?

A. Port scan to locate a host running named.
B. Host scan of isp.net network.
C. DNS denial of service attack in progress.
D. Unauthorized zone transfer.

Answer: D

Detect #9 CGI – Hostile Fire

National Telemedia Corp. a UUnet customer.

```
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:
- GIAC report by Erik Fichtner. http://www.sans.org/y2k/052800-1130.htm

2. Detect was generated by:
- tcpdump except for the extra "05/24" at the beginning. That might have been added. –vv –x options for more data.

05/24 21:52:21.614162 *timestamp* 208.201.208.71.2382 *source.port* > 10.0.0.3.80 *destination.port* : P *tcp flags* 3190942928:3190942960 *start and end sequence #* (32) *bytes* ack 2272436265 *acknowledgement #* win 32120 *options* (DF) *fragment info* (ttl 48, id 42496) *ttl and IP identification address: hex data*

3. Probability the source address was spoofed.
- Low. Attacker needs to see reconnaissance results.

4. Description of attack:
- The attacker probably ran earlier undetected port scans to find a web server. Now he makes actual connections to the web server to see if each vulnerability is present.
- Three different cgi attack probes are sent to the same web server.
- The attacker is appears to be learning how to perform cgi attacks because the "test-cgi" probe ought to go ahead and have the "?*" appended to actually list the files and directories.
- All three attacks have a newline appended, followed by what appears to be identical nonsense values. If executed, they will not return anything meaningful.

5. Attack mechanism:
- CGI is the Common Gateway Interface, an interface specification that communicates information between the client program and the server. CGI scripts can be written in any language, but Perl is the most popular. The attacker-to-be uses the "Get" method to retrieve information from the server to the client.
- In a typical CGI scripting vulnerability, the attacker would add a newline and then the attack command, such as "cat /etc/passwd". The attack works if the server CGI program is careless about parsing the client input data. A defensively written CGI program will find the newline or newline and blank, and truncate the client's input or refuse it. A vulnerable CGI server program will execute the command.

6. Correlation:
- R. Jesus Garcia reports a series of CGI probes that are the exact same three as Erik Fichtner, two weeks later. http://www.sans.org/y2k/053000.htm
- It's a coincidence, but the three vulnerability probes can all be found within 3 pages of each other in McClure/Scambray/Kurtz' "Hacking Exposed" on pages 409-411.
- CGI vulnerabilities are #2 on the SANS Top Ten list. http://www.sans.org/topten.htm

7. Evidence of active targeting:

- Yes. The attacker has probably done previous reconnaissance to confirm that there is a web server listening on port 80. Now he comes back to see if the web server is vulnerable to several exploits. The three-way handshake has already been completed.
- The attacker is probably more of a web programmer than a networking savvy person. A networking or system administrator type attacker would already know what type of operating system and web server was running, and target that. This attacker-to-be might be working his way through the "Hacking Exposed" book.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- 0 = (3+5)-(5+3)
- Actual network issues are unknown. The reporting analyst is unconcerned, posts frequently, and has probably closed up vulnerabilities long ago. Analysis data is host based. Network defenses are unknown, especially for port 80, which is almost always left open.

9. Defensive recommendations:
- Confirm that web server is NOT NCSA HTTPD v 1.5A-Export or earlier, or Apache HTTPD v 1.0.3.
- Remove PHF script from server.
- If the host is IRIX, delete the vulnerable scripts, and/or apply the vendor supplied patch. IRIX have other TCP vulnerabilities so it should be replaced with a more secure host.
- Remove the vulnerable test-cgi script.

10. Multiple choice test question:
What is happening in this trace?
A. Normal web server traffic.
B. Reconnaissance to determine web server vulnerability to an attack.
C. This is the attack.
D. None of the above.
Answer: B

Detect #10 Denial of Service – Spoofing/Hostile Fire/Denial of Service

03:58:15.235672 phwww.netcast.nl > 204.x.x.0: icmp: echo request
03:58:15.239141 phwww.netcast.nl.2356 > 204.x.x.0.echo: udp 1024
03:58:15.368527 phwww.netcast.nl > 204.x.x.255: icmp: echo request
03:58:15.371826 phwww.netcast.nl.41056 > 204.x.x.255.echo: udp 1024
03:58:17.902494 phwww.netcast.nl > 204.x.x.0: icmp: echo request
03:58:17.906341 phwww.netcast.nl.3471 > 204.x.x.0.echo: udp 1024
03:58:18.035617 phwww.netcast.nl > 204.x.x.255: icmp: echo request
03:58:18.039447 phwww.netcast.nl.2933 > 204.x.x.255.echo: udp 1024
03:58:19.870268 phwww.netcast.nl > 204.x.x.0: icmp: echo request
03:58:19.874172 phwww.netcast.nl.42557 > 204.x.x.0.echo: udp 1024
03:58:20.003372 phwww.netcast.nl > 204.x.x.255: icmp: echo request
03:58:20.007210 phwww.netcast.nl.21668 > 204.x.x.255.echo: udp 1024
03:58:21.896327 phwww.netcast.nl > 204.x.x.0: icmp: echo request
03:58:22.028786 phwww.netcast.nl.11873 > 204.x.x.0.echo: udp 1024
03:58:22.030896 phwww.netcast.nl > 204.x.x.255: icmp: echo request
03:58:22.162075 phwww.netcast.nl.54301 > 204.x.x.255.echo: udp 1024
…additional 12+ hours of traffic not shown…

1. Source of Trace:
- GIAC report by Mike Black. http://www.sans.org/y2k/052000.htm

2. Detect was generated by:

- tcpdump format.

03:58:15.235672 *timestamp* phwww.netcast.nl *source* > 204.x.x.0 *destination* : icmp: echo request *icmp message*
03:58:15.239141 *timestamp* phwww.netcast.nl.2356 *source.port* > 204.x.x.0.echo *destination.port* : udp *protocol* 1024 *bytes*

3. Probability the source address was spoofed.
- High probability that the source address is spoofed.
- The attacker is sending the packets with the victim's IP address as the source IP address.

4. Description of attack:
- Unknown Attacker sends spoofed icmp echo request packets to broadcast addresses of this intermediary, commonly known as the Amplifier. Attack #1 commonly known as "smurf".
- Unknown Attacker sends spoofed packets to the UDP echo port of the Amplifier site. Attacker makes sure it's a healthy sized 1024 byte packet each time, so the Analyst should label this denial of service, not aggressive mapping. Attack #2 commonly known as "fraggle".
- The attacker uses new and old style broadcast addresses that end in .0 and .255
- The attacker's source port number is random, so he is probably crafting packets.
- This report has been made by the Amplifier site, not the victim site. The victim site is phwww.netcast.nl
- The Amplifier site does not have defenses in place to prevent being used like this. See Defensive Recommendations.

5. Attack mechanism:
- Both of these Smurf and Fraggle attacks look very classic.
- There are three involved sites: Attacker, Amplifier/Intermediary, and Victim.
- With the Smurf attack, the Attacker sends spoofed icmp echo requests to one or more broadcast addresses of the Amplifier site.
- With the Fraggle attack, the Attacker sends spoofed packets to the UDP echo port of the Amplifier site.
- Because of the destination broadcast addresses, all listening hosts on each network will attempt to respond to the source IP address, the Victim site.
- With the Smurf attack packet, all live hosts at the Amplifier site will send an icmp echo reply to the Victim.
- With the Fraggle attack packet, all live hosts at the Amplifier site will send a UDP reply from the echo port, with the data.
- If, for example, the Amplifier site has 100 or more live hosts on their network, the Attacker can create an attack 100 times or more larger than the traffic he is sending. That's where the name "Amplifier" comes from.

6. Correlation:
- This denial of service attack has been highly publicized. But victims don't like to talk about it for legal and business reasons.
- SANS San Jose Intrusion Detection 2.4/2.5 page 250 a shows sample smurf trace.
- One of our system administrators reports that when the Smurf attack first became popular, our location was hit.

7. Evidence of active targeting:
- Yes. The attacker has probably done earlier reconnaissance to determine that the destination IP can be used as an amplifier, and has enough hosts to generate enough traffic for the attack.
- He manipulates the destination subnet to deluge the source IP address with icmp and udp traffic.
- Netcast appears to be a foreign ISP or Web Hosting company whose services would be interrupted by a bandwidth oriented denial of service like this one.

8. Severity:
- Severity = (critical + lethal) – (system + net countermeasures)
- 3 = (3+4)-(3+1) Amplifier's point of view
- From the Amplifier's point of view, the spoofed traffic is constant and may or may cause some host based denial of service due to host logs filling up.

- 2 = (5+5)-(3+5) Victim's point of view
- Well, the victim got a lower severity, ironically, because there was no mention that they had telephoned about the problem by 12 hours so I deduce the following likely facts: they have redundant Internet connectivity and they quickly configured their routers to drop the packets from the destination IP. The victim's web site promotes video content delivery so they probably have generous bandwidth. They would be vulnerable to a distributed denial of service attack. But perhaps they were busy calling other amplifier sites?

9. Defensive recommendations:

- Network administrators are being urged to take defensive measures by worldwide security leaders such as SANS, Netscan, CERT, and many others. http://www.sans.org, http://netscan.org/broadcast/solutions.html, ftp://info.cert.org/pub/cert_advisories/CA-96.01.UDP_service_denial, http://users.quadrunner.com/chuegen/smurf.cgi

Recommendations for the Victim:

- Investigate router features such as committed access rate to limit bandwidth utilized for icmp traffic.
- Filter the traffic out of your network as near the edge as possible, or request your ISP to do so.
- Consider turning off some logging during an attack.

Recommendations for Amplifiers/Intermediaries:

- Configure your edge routers to ignore these broadcast requests with the "no ip directed-broadcast" command. Or use the appropriate configuration for your non-Cisco routers.

Recommendations for Defending Against Internal Attackers:

- Add filtering at the edge of the network to eliminate source address spoofed packets from entering the network from downstream or leaving upstream.
- Investigate the Cisco feature called "ip verify unicast reverse-path". This feature ensures that packet's return path is through the same router interface it came in on.

10. Multiple choice test question:

What is happening in this trace?

A. Stealthy reconnaissance.
B. Aggressive network mapping.
C. Denial of Service from Amplifier's point of view.
D. Denial of Service from Victim's point of view.

Answer: C