10 Intrusion Detects and Analysis

Submitted for the IDIC Practical Assignment

June 2000

Allison Miller

Notes on the following detects

Detect 1: Anonymous FTP....3
Analysis of Detect 1....3
Detect 2: Illegal Flag-bits....4
Analysis of Detect 2....4
Detect 3: SYN-FIN scan to SunRPC....5
Analysis of Detect 3....5
Detect 4: Common Trojan probe....7
Analysis of Detect 4....7
Detect 5: False positive....9
Analysis of Detect 5....9
Detect 6: Windows probe, possible “network.vbs”....10
Analysis of Detect 6....12
Detect 7: Probe for vulnerable CGI....14
Analysis of Detect 7....14
Detect 8: IRC proxy scan....15
Analysis of Detect 8....15
Detect 9: SYN scan for 80 (HTTP), 8080 (Wingate)....17
Analysis of Detect 9....17
Detect 10: DOS (Smurf & UDP bomb)....19
Analysis of Detect 10....19

General info

- Except for detect 10, these detects were pulled from my home network. We have a DSL connection shared between several Intel-Linux machines. The servers provide few external services.

- I’ve been running Snort v1.6 for a few weeks and during this time I have been trying to tune my intrusion detection system. There are a lot of false positives, one I shared with you as Detect 5.

- Most of the suspicious traffic comes in the form of scans for open, vulnerable services. Since most of these scans are looking for Windows machines, most of the probers haven’t come back yet. This is good for our network but provides for very few, very boring detects.

- I pulled Detect 10 from the GIAC website so I could demonstrate some understanding of signatures that aren’t portscans!

- All local and friendly-fire traffic has been sanitized/obfuscated. My local network is “snorty.dsl” and we are using .58 thru .62. Our secondary DNS and employer are labeled as such.

Detect 1: Anonymous FTP

```
May 28 11:46:04 w062 ftpd[25850]: FTP LOGIN REFUSED (ftp in /etc/ftpusers) FROM c949617-
a.htfde1.ct.home.com [24.2.145.214], anonymous
May 28 11:46:05 w062 ftpd[25850]: lost connection to c949617-a.htfde1.ct.home.com [24.2.145.214]
May 28 11:46:05 w062 ftpd[25850]: FTP session closed
```

Analysis of Detect 1

<table>
<tr><td>1. Source of trace</td><td colspan="3">My local network:
System: Intel compatible RHLv6.1
Connectivity: DSL line, residential service</td></tr>
<tr><td>2. Detect generated by:</td><td colspan="3">This trace was generated in the /var/log/secure file.</td></tr>
<tr><td>3. Probability of spoofed source address</td><td colspan="3">Very low. The attacker is gathering information on my network. Specifically,I searching for available anonymous FTP services. This detect shows that a TCP connection was established.</td></tr>
<tr><td>4. Description of attack:</td><td colspan="3">The agent is looking for available anonymous FTP services.</td></tr>
<tr><td>5. Attack mechanism:</td><td colspan="3">Establishment of an FTP connection to a specific machine running the FTP service. The agent then tries to logon as “anonymous”. Anonymous users only have to supply an e-mail (non-authenticated) for access to ftp services. Anonymous FTP-services can supply storage space for pirated software or other contraband, and also serve as a jumping off point for further attacks.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-correlations:</b> No other probes logged by this agent. However, I have had other probes for anonymous FTP in the past few months. The administrator of this machine did not realize that anonymous FTP is enabled by default on their install, and so until the log files were checked, people were able to log in as anonymous, although there were no files in the directory, and anonymous had no “write” privileges. Other servers on the network have also been probed in a similar fashion.
<b>Inter-correlations:</b>
It is common for attackers to probe for available anonymous FTP servers. This is commonly reported on the GIAC website.
See CERT advisory, Original issue date: July 14, 1993:
http://www.cert.org/advisories/CA-93.10.anonymous.FTP.activity.html
CVE:
Anonymous FTP is enabled CAN-1999-0497
Inappropriate permissions in anonymous FTP account, CAN-1999-0527</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">Yes. Specific hosts on this network were targeted (servers running FTP).</td></tr>
<tr><td rowspan="5">8. Severity:</td><td>Criticality</td><td>3</td><td>This is a web server, but not one supplying critical services.</td></tr>
<tr><td>Lethality</td><td>3</td><td>An attacker would gain user-level access as “anonymous”.</td></tr>
<tr><td>System Countermeasures</td><td>4</td><td>Updated OS software, limited services available, TCP-wrappers in place.</td></tr>
<tr><td>Network Countermeasures</td><td>1</td><td>No Firewall, but only one route into the network.</td></tr>
<tr><td>Severity</td><td>1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">Keep anonymous FTP disabled, or remove the FTP service entirely, if possible.</td></tr>
<tr><td>10. Multiple choice test question, write a question based on the trace and your analysis with your answer.</td><td colspan="3">Downstream liability can become an issue for hosts of anonymous FTP services. True or False?

Anonymous FTP services are non-authenticated, and if unsupervised, the server can become a storage area for contraband data. Data stored on, or attacks launched from a server may implicate the owners/administrators in</td></tr>
</table>

	illegal activity. In some cases liability is in the form of negligence, in other cases the owners/administrators may be personally implicated. The answer is “True”.

Detect 2: Illegal Flag-bits

```
May 24 03:31:38 207.200.89.40:80 -> snorty.dsl.61:2793 UNKNOWN *1**R*** RESERVEDBITS
May 30 22:04:26 130.130.68.50:80 -> snorty.dsl.58:1906 UNKNOWN *1**R*** RESERVEDBITS
- - - - - -
[root@snorty.dsl.61]# nslookup 207.200.89.40
Name:    myvip-a.netscape.com
Address:  207.200.89.40
- - - - - -
Output from ARIN WHOIS: http://www.arin.net/whois
University of Wollongong (NET-UOWNET)
Wollongong, New South Wales
AUSTRALIA
  Netname: UOWNET   Netnumber: 130.130.0.0
Coordinator: Cliffe, Steve  (SC143-ARIN)  steve@UOW.EDU.AU
+61 2 4221 3810 (FAX) +61 2 4221 4504
     Domain System inverse mapping provided by:
     WRAITH.CS.UOW.EDU.AU          130.130.64.1
     WYRM.ITS.UOW.EDU.AU           130.130.68.1
```

Analysis of Detect 2

<table>
<tr><td>1. Source of trace</td><td colspan="3">My local network:
System: Intel compatible RHLv6.1
Connectivity: DSL line, residential service</td></tr>
<tr><td>2. Detect generated by:</td><td colspan="3">Snort v1.6, Ruleset dated 3/2000 (Portscan.log)</td></tr>
<tr><td>3. Probability of spoofed source address</td><td colspan="3">Low. This is either an error or a reconnaissance attempt.</td></tr>
<tr><td>4. Description of attack:</td><td colspan="3">Unnatural flag settings in packets coming from a web server. Possibly attempt at OS fingerprinting, or an error coming from a webserver.</td></tr>
<tr><td>5. Attack mechanism:</td><td colspan="3">OS fingerprinting using illegal-flag bits is used in the popular network tools “Queso” and “nmap”, nmap being the more sophisticated tool. Given unnatural flag-bit settings in a packet, specific operating systems will respond predictably.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-correlations:</b> These are the only packets with “Reserved” Illegal Flag-bits captured by my intrusion detection system. No other unusual activity from either of the two addresses.
<b>Inter-correlations:</b>
Illegal flag bits are commonly reported on the GIAC website as being attempts at OS fingerprinting (signatures of NMAP and Queso include illegal flag bits), or as errors (like the strange traffic from Demon.net)</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">Yes. The trace consists of single packets sent to specific machines. The hosts recorded are both web servers, and probably sent these strange packets back to client (browsing) machines.</td></tr>
<tr><td rowspan="3">8. Severity:</td><td>Criticality</td><td>3</td><td>Non-critical UNIX client and small webserver.</td></tr>
<tr><td>Lethality</td><td>2</td><td>Possible reconnaisance.</td></tr>
<tr><td>System Countermeasures</td><td>4</td><td>Patches up to date. Few network services running.</td></tr>
</table>

	Network Countermeasures	1	No Firewall, but only one route into the network.
	Severity	0	Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
9. Defensive recommendation:	None. A firewall can serve to screen out any illegal flag-bit traffic, however this particular traffic (without further correlated activity) does not suggest a threat.		
10. Multiple choice test question, write a question based on the trace and your analysis with your answer.	Which combination of TCP flag-bits is legitimate? A. **SF**** B. *1**R*** C. *****PA* D. ******** A Syn-Fin packet suggests a packet trying to both initiate and end a session, an unnatural phenomenon. Reserved bits are anomalous as they are not used during normal TCP communications. Having no flags set is also anomalous. A Push-Ack is the only naturally occurring traffic.		

Detect 3: SYN-FIN scan to SunRPC

```
Jun  4 21:21:50 213.26.142.2:111 -> snorty.dsl.58:111 SYNFIN **SF****
Jun  4 21:21:51 213.26.142.2:619 -> snorty.dsl.58:111 SYN **S*****
Jun  4 21:21:50 213.26.142.2:111 -> snorty.dsl.59:111 SYNFIN **SF****
Jun  4 21:21:50 213.26.142.2:111 -> snorty.dsl.60:111 SYNFIN **SF****
Jun  4 21:21:50 213.26.142.2:111 -> snorty.dsl.62:111 SYNFIN **SF****
Jun  4 21:21:51 213.26.142.2:620 -> snorty.dsl.62:111 SYN **S*****
- - - - - - -
inetnum: 213.26.142.0 - 213.26.142.63
netname: CIES
descr: CIES
country: IT
admin-c: AI747-RIPE
tech-c: AI747-RIPE
status: ASSIGNED PA
notify: network@cgi.interbusiness.it
mnt-by: INTERB-MNT
changed: network@cgi.interbusiness.it 20000321
source: RIPE
- - - - - - -
person: Antonio Iofrida
address: CIES
address: Contrada Vermicelli, 3/d
address: I- Rende (CS)
address: Italy
phone: +39 984 8314207
fax-no: +39 984 8314223
nic-hdl: AI747-RIPE
changed: domain@cgi.interbusiness.it 20000321
source: RIPE
```

Analysis of Detect 3

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000

<table>
<tr><td>3. Probability of spoofed source address</td><td colspan="3">Low. The scan appears to be gathering information on my local subnet.</td></tr>
<tr><td>4. Description of attack:</td><td colspan="3">SYN-FIN scan against the hosts on our subnet. Each machine scanned received one SYN-FIN packet from the probing system's port 111 (portmapper). Two hosts then received another packet (a SYN) from iterating ports on the probing machines. It looks like the prober got some kind of response back from hosts .58 and .62, and then tried to establish a connection. This scan is somewhat fast. I'm not sure why the time-stamps are out of order, probably just different routing.</td></tr>
<tr><td>5. Attack mechanism:</td><td colspan="3">Syn/Fin packets are unnatural, suggesting they were crafted. Crafted Syn/Fins are commonly used in stack analysis/OS fingerprinting. Probes to Port 111 are also common, SunRPC is a commonly targeted service. It is interesting that to see the SF scans to port 111, this may be considered a new signature or code branch. The classic SF scans use SRC port 0, but that has mutated over the past few months with a reported trend of SRC port = DST port, i.e. POPII (109 to 109) and Squid Proxy (3128 to 3128).</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-correlations:</b> Unique activity on this subnet since monitoring began. No other suspicious activity from this agent logged.
<b>Inter-correlations:</b>
See SANS advisories on SunRPC activity:
• CA-99-16, Buffer Overflow in Sun Solstice AdminSuite Daemon sadmind http://www.cert.org/advisories/CA-99-16-sadmind.html
• CA-99-12, Buffer overflow in amd http://www.cert.org/advisories/CA-99-12-amd.html
• CA-99-08, Buffer overflow in rpc.cmsd http://www.cert.org/advisories/CA-99-08-cmsd.html
• CA-99-05, Vulnerability in statd exposes vulnerability in automountd http://www.cert.org/advisories/CA-99-05-statd-automountd.html
• CA-98.12, Remotely Exploitable Buffer Overflow Vulnerability in mountd http://www.cert.org/advisories/CA-98.12.mountd.html
• CA-98.11, Vulnerability in ToolTalk RPC service http://www.cert.org/advisories/CA-98.11.tooltalk.html
CVE:
rpc.ttdbserverd CVE-1999-0687, CVE-1999-0003, CVE-1999-0693
rpc.cmsd CVE-1999-0696
rpc.statd CVE-1999-0018, CVE-1999-0019</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">No, scan runs through subnet. However, subnet scanned and then connections initiated to specific hosts.</td></tr>
<tr><td rowspan="5">8. Severity:</td><td>Criticality</td><td>3</td><td>Scan runs through network indiscriminately, few services being run from residential subnet.</td></tr>
<tr><td>Lethality</td><td>2</td><td>May provide recon information on our hosts. Also, attacks targeted at port 111 are potentially lethal exploits and setups for rpc.statd, although there are no Sun machines on this network.</td></tr>
<tr><td>System Countermeasures</td><td>3</td><td>Patches mostly up to date. Few network services running. Some systems are better protected than others are.</td></tr>
<tr><td>Network Countermeasures</td><td>1</td><td>No Firewall, but only one route into the network.</td></tr>
<tr><td>Severity</td><td>1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">Time for a firewall! This activity may provide recon on networked systems. However, there are no Sun boxes on our network. This appears to be a non</td></tr>
</table>

	targeted probe.
10. Multiple choice test question, write a question based on the trace and your analysis with your answer.	An attacker targeting Solaris hosts would be most interested in activity on which of the following ports? A. TCP: 1 B. UDP: 139 C. TCP: 111 D. UDP: 22 TCP: 1 can be used to ID SGI Irix or SCO Unix systems. UDP: 139 is Netbios Session service, found on Microsoft systems. UDP: 22 is the default port for PCAnywhere. The answer is C. TCP: 111 is the SunRPC port. Other systems may use services on this port, but it would be one of the ports of interest to attackers targeting Solaris machines.

Detect 4: Common Trojan probe

```
[**] Back Orifice [**]
05/24-20:36:14.454078 212.159.68.118:1025 -> snorty.dsl.58:31337
UDP TTL:114 TOS:0x0 ID:19419
Len: 27

[**] Back Orifice [**]
05/24-20:36:14.468730 212.159.68.118:1025 -> snorty.dsl.60:31337
UDP TTL:114 TOS:0x0 ID:19931
Len: 27

[**] Back Orifice [**]
05/24-20:36:14.471019 212.159.68.118:1025 -> snorty.dsl.59:31337
UDP TTL:113 TOS:0x0 ID:19675
Len: 27

[**] Back Orifice [**]
05/24-20:36:14.487350 212.159.68.118:1025 -> snorty.dsl.62:31337
UDP TTL:114 TOS:0x0 ID:20443
Len: 27
- - - - - - - -
[root@snort]# nslookup 212.159.68.118
Name:    118.01-02.quay.dial.plus.net.uk
Address:  212.159.68.118
```

Analysis of Detect 4

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000
3. Probability of spoofed source address	Low. The scan appears to be gathering information on my local subnet.
4. Description of attack:	Scanning for Back Orifice, a very common Windows Trojan. This scan is very fast.
5. Attack mechanism:	It appears to be an automated probe, as the source port is constant and it is a quick scan. The packet IDs are out of order. Perhaps the out-of-order packet took a different route, which would explain the reduced TTL. If the packet IDs are read in order, the scan is taking place sequentially over the subnet. If the packet IDs are correct, the prober may be a somewhat busy networked

	machine.		
6. Correlations:	**Intra-Correlations:** Surprisingly, this is the only real trojan probe received by my network. This is also the only activity from the probing system. **Inter-Correlations:** See CERT Vulnerability Note, Friday, October 2, 1998: http://www.cert.org/vul_notes/VN-98.07.backorifice.html CVE: A hacker utility is installed on a system CAN-1999-0660		
7. Evidence of active targeting:	No.		
8. Severity:	Criticality	3	Scan runs through network indiscriminately, few services being run from residential subnet.
	Lethality	0	Back Orifice gives the client remote access of some Windows based machines, no Windows on this network.
	System Countermeasures	3	Patches mostly up to date. Few network services running. Some systems are better protected than others are.
	Network Countermeasures	1	No Firewall, but only one route into the network.
	Severity	-1	Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
9. Defensive recommendation:	None needed.		
10. Multiple choice test question, write a question based on the trace and your analysis with your answer.	Which application does not support encrypted communication? A. Loki B. BO2K C. Telnet D. PCAnywhere Loki and BO2K are hacker tools that can use encryption to disguise their communications. PCAnywhere is a remote administrator's tool that uses encryption to secure communications over networks. (Yes, it has been argued that BO2K is also a remote administrator's tool, however, let's just say the marketing is the message, with apologies to Marshall McLuhan). Telnet is a commonly used TCP protocol for establishing shell accounts on remote machines. All communication happens "in the clear". Ssh is a much safer protocol that provides the same functionality. Answer = C		

Detect 5: False positive

```
[**] ICQ Trojan [**]
05/26-17:00:17.589994 smtp.our-telecommutable-employer.com:53 -> snorty.dsl.58:4950
UDP TTL:51 TOS:0x0 ID:42615
Len: 242

[**] ICQ Trojan [**]
05/26-17:00:25.315321 admin.our-secondary-dns.com:53 -> snorty.dsl.58:4950
UDP TTL:55 TOS:0x0 ID:15741
Len: 177

[**] ICQ Trojan [**]
05/26-17:00:32.380164 admin.our-secondary-dns.com:53 -> snorty.dsl.58:4950
UDP TTL:21 TOS:0x0 ID:30520
Len: 138
- - -  - - -
[root@lilith snort]# nslookup 208.210.124.36
Name:    admin.our-secondary-dns.com
Address:  obfuscated.obfuscated.36
- - - - - - -
[root@lilith snort]# nslookup 207.8.203.101
Name:    smtp.our-telecommutable-employer.com
Address: obfuscated.obfuscated.101
```

Analysis of Detect 5

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service		
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000		
3. Probability of spoofed source address	Low. This is legitimate traffic that set off the Snort filter.		
4. Description of attack:	This is legitimate traffic that set off the Snort filter's "ICQ Trojan" alert. UDP traffic to port 4950 is a signature for a Trojan-horse program.		
5. Attack mechanism:	This traffic is actually an exchange of DNS information between one of our host machines and 1) the mail server of an employer we frequently telecommute with (mail is often forwarded from their mail servers into our personal mail servers) and 2) our secondary DNS provider (there are domain names associated with that host).		
6. Correlations:	**Intra-correlations:** This is normal traffic for this network. **Inter-correlations:** Re: ICQ Trojan Bugtraq: BlueBoar@THIEVCO.COM posted a summary regarding the ICQ Trojan, from the vuln-dev mailing list to Bugtraq on 1999, Nov 06. "ICQ 2000 trojan/worm (VD#5)" CVE: A hacker utility is installed on a system CAN-1999-0660		
7. Evidence of active targeting:	Yes. The DNS traffic is aimed at determining information for specific hosts.		
8. Severity:	Criticality	4	Light mail and web traffic come to this server. Also used to communicate with employer.
	Lethality	0	Normal DNS traffic.
	System Countermeasures	3	Patches mostly up to date. Few network services running. Some systems are better protected than others are.

	Network Countermeasures	1	No Firewall, but only one route into the network.
	Severity	0	Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
9. Defensive recommendation:	Edit filters to screen out “friendly-fire”. Since this is a home network that uses an external DNS server and is often used to telecommute, there is a lot of traffic that is probably normal even if it is not part of the local network.		
10. Multiple choice test question, write a question based on the trace and your analysis with your answer.	False positives can be useful for administrators because: A. False positives help administrators to tune their intrusion detection system. B. False positives help administrators identify misconfigured network device. C. False positives educate administrators about the type of traffic on their networks. D. All of the above. False positive detects are legitimate traffic patterns that for some reason set off the alarms on intrusion detection systems. Most traffic on intrusion detection systems (at least initally) consists of false positives. Filters on intrusion detection systems will often need to be tuned to your particular network; false positives are misdiagnosed traffic. On the other hand, it is possible that the traffic is the result of a misconfiguration of a network device. In these cases the intrusion detection system acts as a network diagnostic tool. False positives are a sample of network traffic, and upon examination, teach administrators the ins and outs of their network. The answer is D, all of the above.		

Detect 6: Windows probe, possible “network.vbs”

```
[**] SMB Name Wildcard [**]
06/07-16:36:33.185959 208.193.119.167:137 -> snorty.dsl.58:137
UDP TTL:118 TOS:0x0 ID:25388
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:34.672419 208.193.119.167:137 -> snorty.dsl.58:137
UDP TTL:118 TOS:0x0 ID:25644
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:43.473170 208.193.119.167:137 -> snorty.dsl.59:137
UDP TTL:118 TOS:0x0 ID:26924
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:44.970492 208.193.119.167:137 -> snorty.dsl.59:137
UDP TTL:118 TOS:0x0 ID:27180
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:46.470002 208.193.119.167:137 -> snorty.dsl.59:137
UDP TTL:118 TOS:0x0 ID:27436
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:53.745595 208.193.119.167:137 -> snorty.dsl.60:137
DP TTL:118 TOS:0x0 ID:28460
```

```
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:55.241935 208.193.119.167:137 -> snorty.dsl.60:137
UDP TTL:118 TOS:0x0 ID:28716
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:36:56.743786 208.193.119.167:137 -> snorty.dsl.60:137
UDP TTL:118 TOS:0x0 ID:28972
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:37:04.023635 208.193.119.167:137 -> snorty.dsl.61:137
UDP TTL:118 TOS:0x0 ID:29996
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:37:10.439100 208.193.119.167:137 -> snorty.dsl.62:137
UDP TTL:118 TOS:0x0 ID:33068
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:37:11.937017 208.193.119.167:137 -> snorty.dsl.62:137
UDP TTL:118 TOS:0x0 ID:33324
Len: 58

[**] SMB Name Wildcard [**]
06/07-16:37:13.436611 208.193.119.167:137 -> snorty.dsl.62:137
UDP TTL:118 TOS:0x0 ID:33580
Len: 58

- - - - - - - - -
[**] SMB Name Wildcard [**]
06/07-16:36:33.185959 208.193.119.167:137 -> snorty.dsl.58:137
UDP TTL:118 TOS:0x0 ID:25388
Len: 58
33 F8 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  3........... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

[**] SMB Name Wildcard [**]
06/07-16:36:34.672419 208.193.119.167:137 -> snorty.dsl.58:137
UDP TTL:118 TOS:0x0 ID:25644
Len: 58
33 FA 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  3........... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01

[**] SMB Name Wildcard [**]
06/07-16:36:36.171933 208.193.119.167:137 -> snorty.dsl.58:137
UDP TTL:118 TOS:0x0 ID:25900
Len: 58
33 FC 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  3........... CKA
```

```
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01                                            ..

[**] SMB Name Wildcard [**]
06/07-16:36:43.473170 208.193.119.167:137 -> snorty.dsl.59:137
UDP TTL:118 TOS:0x0 ID:26924
Len: 58
34 02 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  4........... CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01
- - - - - -
[**] SMB C access [**]
06/07-16:37:04.464747 208.193.119.167:2539 -> snorty.dsl.61:139
TCP TTL:118 TOS:0x0 ID:31276  DF
*****PA* Seq: 0xD96A34   Ack: 0x461A31E3   Win: 0x21E3
UDP TTL:118 TOS:0x0 ID:29996
Len: 58

[**] SMB C access [**]
06/07-16:37:04.464747 208.193.119.167:2539 -> snorty.dsl.61:139
TCP TTL:118 TOS:0x0 ID:31276  DF
*****PA* Seq: 0xD96A34   Ack: 0x461A31E3   Win: 0x21E3

- - - - - - - - - -
[**] SMB C access [**]
06/07-16:37:04.464747 208.193.119.167:2539 -> snorty.dsl.61:139
TCP TTL:118 TOS:0x0 ID:31276  DF
*****PA* Seq: 0xD96A34   Ack: 0x461A31E3   Win: 0x21E3
00 00 00 7B FF 53 4D 42 73 00 00 00 00 10 00 00  ...{.SMBs.......
00 00 00 00 00 00 00 00 00 00 00 00 00 00 DD 14  ................
01 00 81 9F 0D 75 00 63 00 68 0B 32 00 00 00 49  .....u.c.h.2...I
03 00 00 00 00 00 00 00 00 00 00 01 00 00 00 26  ...............&
00 4B 4C 50 00 57 4F 52 4B 47 52 4F 55 50 00 57  .KLP.WORKGROUP.W
69 6E 64 6F 77 73 20 34 2E 30 00 57 69 6E 64 6F  indows 4.0.Windo
77 73 20 34 2E 30 00 04 FF 00 00 00 02 00 01 00  ws 4.0..........
0D 00 00 5C 5C 57 30 36 31 5C 43 00 41 3A 00     ...\\W061\C.A:.
- - - -
```

Output from ARIN WHOIS
http://www.arin.net/whois

```
UUNET Technologies, Inc. (NETBLK-UUNET1996B) UUNET1996B
                                        208.192.0.0 - 208.249.255.255
Auto-Graphics (NETBLK-UU-208-193-119) UU-208-193-119
                                        208.193.119.0 - 208.193.119.255
```

Analysis of Detect 6

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000
3. Probability of spoofed source address	Low. This scan is an attempt at network recon.
4. Description of attack:	The agent is scanning for Windows machines with open NetBios SMB Services. This is an approximate description of the attack: three packets are

<table>
<tr><td></td><td colspan="3">sent to each machine's UDP port 137. The ID numbers increment by 256. The TTL stays constant, although this is a fast scan so might not be a good diagnostic. Upon some criteria, probably a response from a host, a connection is initiated with TCP 139. This consistent signature suggests both crafted packets and automated activity.</td></tr>
<tr><td>5. Attack mechanism:</td><td colspan="3">This traffic is indicative of the "network.vbs" worm, which looks for unprotected NetBios SMB servers using the NBTSTAT –A command for Windows (UDP:137), which enumerates valuable information about the network being targeted. If the worm gets a response from any of the packets sent to port 137, it will follow up by initiating a TCP connection to port 139 (trying to mount a share named "C").
* Note: I think it is strange I even get the follow-up SMB C Access attempt as there are no services running on the port 137. Oh well.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-Correlations:</b> This is the first scan detected, since this it has been followed by a veritable flood of "SMB Name Wildcard", although the "SMB C Access" is less common. This is the only activity detected from this host.
<b>Inter-correlations:</b>
CERT:
• IN-2000-02, Exploitation of Unprotected Windows Networking Shares http://www.cert.org/incident_notes/IN-2000-02.html
• IN-2000-03, 911Worm, http://www.cert.org/incident_notes/IN-2000-03.html
SANS alert http://www.sans.org/newlook/alerts/911worm.htm,
SANS GIAC detect http://www.sans.org/y2k/honeypot_catch.htm provided by Bryce Alexander
CVE Entries:
SMB shares with poor access control - CAN-1999-0520
NFS exports to the world - CAN-1999-0554</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">No. This traffic covers our subnet. Also, if the source of this traffic is the "network.vbs" worm, the worm picks a random subnet to probe.</td></tr>
<tr><td rowspan="5">8. Severity:</td><td>Criticality</td><td>3</td><td>Scan runs through network indiscriminately, few services being run from residential subnet.</td></tr>
<tr><td>Lethality</td><td>0</td><td>This particular vulnerability allows attackers to enumerate network informationon some Windows based machines. No Windows on this network.</td></tr>
<tr><td>System Countermeasures</td><td>3</td><td>Patches mostly up to date. Few network services running. Some systems are better protected than others are.</td></tr>
<tr><td>Network Countermeasures</td><td>1</td><td>No Firewall, but only one route into the network.</td></tr>
<tr><td>Severity</td><td>-1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">None. We don't do Windows. However, a firewall of properly configured network device can filter for this traffic.</td></tr>
<tr><td>10. Multiple choice test question, write a question based on the trace and your analysis with your answer.</td><td colspan="3">Which of the following malicious programs will use the Internet to replicate?
A. Melissa virus
B. ILOVEYOU worm
C. "network.vbs" worm
D. All of the above

The Melissa virus and the ILOVEYOU worm will both send itself to addresses in the local MS Exchange address book. The "network.vbs" worm jumps from machine to machine across the Internet by exploiting open Windows shares. The answer is D, all of the above.</td></tr>
</table>

Detect 7: Probe for vulnerable CGI

```
[**] TEST-CGI probe! [**]
06/07-18:29:59.214627 24.0.199.135:4148 -> snorty.dsl.62:80
TCP TTL:117 TOS:0x0 ID:9725  DF
*****PA* Seq: 0x383E0F1   Ack: 0xEF1AC058   Win: 0x2238
48 45 41 44 20 2F 63 67 69 2F 74 65 73 74 2D 63  HEAD /cgi/test-c
67 69 2E 74 63 6C 20 48 54 54 50 2F 31 2E 30 0D  gi.tcl HTTP/1.0.
0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A  .User-Agent: Moz
69 6C 6C 61 2F 34 2E 37 20 5B 65 6E 5D 20 28 57  illa/4.7 [en] (W
69 6E 39 35 3B 20 55 29 0D  65                    in95; U)..

[**] Classifieds CGI access attempt [**]
06/07-18:27:53.580841 24.0.199.135:3831 -> snorty.dsl.62:80
TCP TTL:117 TOS:0x0 ID:22004  DF
*****PA* Seq: 0x381DF19   Ack: 0xE871C6DD   Win: 0x2238
48 45 41 44 20 2F 63 67 69 62 69 6E 2F 63 6C 61  HEAD /cgibin/cla
73 73 69 66 69 65 64 73 2E 63 67 69 20 48 54 54  ssifieds.cgi HTT
50 2F 31 2E 30 0D 0A 55 73 65 72 2D 41 67 65 6E  P/1.0..User-Agen
74 3A 20 4D 6F 7A 69 6C 6C 61 2F 34 2E 37 20 5B  t: Mozilla/4.7 [
65 6E 5D 20 28 57 69 6E 39 35 3B 20 55 29 0D 0A  en] (Win95; U)..

[Etc. etc.]
```

Analysis of Detect 7

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000
3. Probability of spoofed source address	Low. This is an attempt at recon on my webserver.
4. Description of attack:	Probe for webserver vulnerabilities via exploitable CGI scripts and related web applications.
5. Attack mechanism:	The attacker connected to the webserver 1372 times in a very short time period (about 10 seconds), suggesting an automated attack from a host with a high-bandwidth connection. The source ports were increasing but sometimes skipped a few ports, suggesting that (given the high speed of the attack), the machine is very busy. With the source port and ID (see above) different on all packets, these are probably not crafted packets. The attacker went through a laundry list of potential vulnerabilities. This is a very loud , network intensive scan.
6. Correlations:	**Intra-correlations:** Other webservers have been probed for vulnerable CGI scripts and web applications . This is the only suspicious traffic from this host logged. **Inter-correlations:** CERT: http://www.cert.org/advisories/CA-97.07.nph-test-cgi_script.html http://www.cert.org/advisories/CA-96.06.cgi_example_code.html http://www.cert.org/advisories/CA-97.12.webdist.html L0pht Security Advisory re: the "test-cgi" problem, posted in 1996 CVE: CAN-1999-0736 CVE-1999-0067 CVE-1999-0068

	CVE-1999-0270 CVE-1999-0346 CVE-2000-0207		
7. Evidence of active targeting:	Yes. Only one webserver was probed.		
8. Severity:	Criticality	3	Non-critical UNIX client and small webserver.
	Lethality	5	Many of the vulnerabilities scanned for can be exploited to give the attacker root access to the system across the Internet.
	System Countermeasures	4	Patches mostly up to date. Few network services running.
	Network Countermeasures	1	No Firewall, but only one route into the network.
	Severity	3	Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)
9. Defensive recommendation:	Only use CGI scripts and web application code as necessary for operation of webserver. Keep up to date on vulnerabilities in common scripts used by webserver.		
10. Multiple choice test question, write a question based on the trace and your analysis with your answer.	Our intrusion detection system will do content analysis. We set up a filter that looks for the combination of “GET”, “cgi-bin”, and “/etc/passwd”. This filter is best described as: A. An effective meta-filter B. An exploit-specific filter C. Reliable bounds checking D. Covert channel filtering A: The filter will act as an effective meta-filter, it will catch a number of common CGI-BIN exploits, including phf, php, and aglimpse. This filter does not target a specific expoit. It has nothing to do with bounds-checking (that happens during application development, anyway) and is not looking for covert channel activity.		

Detect 8: IRC proxy scan

```
[**] WinGate 1080 Attempt [**]
06/07-19:18:43.066856 207.114.4.46:3301 -> snorty.dsl.61:1080
TCP TTL:55 TOS:0x0 ID:61991  DF
**S***** Seq: 0x4A049ED6   Ack: 0x0   Win: 0x4000
TCP Options => MSS: 1460 NOP WS: 0 NOP NOP TS: 2151395 0
- - - - -
46.4.114.207.in-addr.arpa        name = ProxyScan.MD.US.Undernet.Org
4.114.207.in-addr.arpa   nameserver = ns1.abs.net
4.114.207.in-addr.arpa   nameserver = ns2.abs.net
```

Analysis of Detect 8

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000
3. Probability of spoofed source address	Very low. This is a information gathering network probe. Also, this traffic has been recorded from this server by others. (See GIAC website)
4. Description of attack:	This is a network recon probe, searching for vulnerable WinGate servers.
5. Attack mechanism:	The WinGate application allows a LAN to share a network connection.

<table>
<tr><td></td><td colspan="3">Unfortunately, the default configuration does not log traffic, and allows intruders to use the WinGate service as a proxy server to launder their hostile network traffic. Sites running vulnerable WinGate servers may be implicated in a security incident.
In this case, when IRC connections are initiated with this server, the IRC clients are immediately scanned. It is unknown what the intent of this probe is.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-Correlations:</b> This is the first WinGate probe I received, since then I've received several similar scans. I returned to the IRC server (md.us.undernet.org) and was automatically probed again.
<b>Inter-Correlations:</b>
SANS GIAC notice: http://www.sans.org/y2k/IRC.htm , reported by Tim White ** Tim's description of this activity matched the signature I received perfectly, and prompted me to self-correlate by returning to the offending IRC server.
See CERT vendor note: http://www.cert.org/vul_notes/VN-98.03.WinGate.html
CVE: CVE-1999-0290</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">Yes. An IRC client was specifically probed upon establishing a connection.</td></tr>
<tr><td>8. Severity:</td><td>Criticality</td><td>2</td><td>Non-critical UNIX desktop.</td></tr>
<tr><td></td><td>Lethality</td><td>2</td><td>Allows attacker to use site's services to proxy their network traffic.</td></tr>
<tr><td></td><td>System Countermeasures</td><td>4</td><td>Patches mostly up to date. Few network services running.</td></tr>
<tr><td></td><td>Network Countermeasures</td><td>1</td><td>No Firewall, but only one route into the network.</td></tr>
<tr><td></td><td>Severity</td><td>-1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">Filter for traffic to 1080 at boundary network device.</td></tr>
<tr><td>10. Multiple choice test question, write a question based on the trace and your analysis with your answer.</td><td colspan="3">The detect shown suggests:
A. False positive: An IRC connection being established on one of the client's ephemeral ports.
B. False positive: IRC connection in progress through a WinGate proxy server.
C. True detect: IRC client using WinGate proxy server to establish a connection to an IRC server.
D. True detect: IRC server scanning for vulnerable WinGate server.

IRC clients initiate the IRC sessions, not servers, so you shouldn't see (unsolicited) SYNs from the server to a client's ephemeral port. SYNs are also not used once the connection has been established, ruling out the second option. The third option makes no sense, since the connection only shows activity from the IRC server and doesn't appear to actually use the WinGate service. This detect is typical for WinGate probes, thus the answer must be D. Also, check out the resolved name of the server: "ProxyScan.MD.US.Undernet.Org". A little clue, there.</td></tr>
</table>

Detect 9: SYN scan for 80 (HTTP), 8080 (Wingate)

```
Jun  8 21:35:36 216.53.151.3:1421 -> snorty.dsl.58:80 SYN **S*****
Jun  8 21:35:38 216.53.151.3:1422 -> snorty.dsl.58:8080 SYN **S*****
Jun  8 21:35:39 216.53.151.3:1423 -> snorty.dsl.59:80 SYN **S*****
Jun  8 21:35:40 216.53.151.3:1424 -> snorty.dsl.59:8080 SYN **S*****
Jun  8 21:35:40 216.53.151.3:1425 -> snorty.dsl.60:80 SYN **S*****
Jun  8 21:35:40 216.53.151.3:1426 -> snorty.dsl.60:8080 SYN **S*****
Jun  8 21:35:38 216.53.151.3:1427 -> snorty.dsl.61:80 SYN **S*****
Jun  8 21:35:40 216.53.151.3:1428 -> snorty.dsl.61:8080 SYN **S*****
Jun  8 21:35:40 216.53.151.3:1429 -> snorty.dsl.62:80 SYN **S*****
Jun  8 21:35:41 216.53.151.3:1430 -> snorty.dsl.62:8080 SYN **S*****
Jun  8 21:35:41 216.53.151.3:1428 -> snorty.dsl.61:8080 SYN **S*****
Jun  8 21:35:42 216.53.151.3:1430 -> snorty.dsl.62:8080 SYN **S*****

Name:    216-53-151-003.ppp.mpinet.net
Address:  216.53.151.3

[**] WinGate 8080 Attempt [**]
06/08-21:35:36.212318 216.53.151.3:1422 -> snorty.dsl.58:8080
TCP TTL:112 TOS:0x0 ID:21104  DF
**S***** Seq: 0x186BEDF   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 536 NOP NOP SackOK

06/08-21:35:36.212318 216.53.151.3:1422 -> snorty.dsl.58:8080
TCP TTL:112 TOS:0x0 ID:21104  DF
**S***** Seq: 0x186BEDF   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 536 NOP NOP SackOK

[**] WinGate 8080 Attempt [**]
06/08-21:35:37.020042 216.53.151.3:1422 -> snorty.dsl.58:8080
TCP TTL:112 TOS:0x0 ID:22384  DF
**S***** Seq: 0x186BEDF   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 536 NOP NOP SackOK

[**] WinGate 8080 Attempt [**]
06/08-21:35:37.762474 216.53.151.3:1422 -> snorty.dsl.58:8080
TCP TTL:112 TOS:0x0 ID:24176  DF
**S***** Seq: 0x186BEDF   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 536 NOP NOP SackOK

[**] WinGate 8080 Attempt [**]
06/08-21:35:38.497100 216.53.151.3:1422 -> snorty.dsl.58:8080
TCP TTL:112 TOS:0x0 ID:29552  DF
**S***** Seq: 0x186BEDF   Ack: 0x0   Win: 0x2000
TCP Options => MSS: 536 NOP NOP SackOK
```

Analysis of Detect 9

1. Source of trace	My local network: System: Intel compatible RHLv6.1 Connectivity: DSL line, residential service
2. Detect generated by:	Snort v1.6, Ruleset dated 3/2000
3. Probability of spoofed source address	Low. This is a information gathering network probe.
4. Description of attack:	This is a network recon probe, searching for vulnerable webservers and

<table>
<tr><td></td><td colspan="3">(apparently) WinGate servers.</td></tr>
<tr><td>5. Attack mechanism:</td><td colspan="3">This is an automated probe that sends a SYN to port 80 and port 8080 on each of the systems in our subnet. The scan is fast, 12 packets over 8 seconds. The source ports are incrementing, except for the repeated ports at the end of the scan, and the IDs are increasing normally. This suggests that the packets are not crafted.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-correlations:</b> I have had several scans across my network searching for traffic to 8080. Scans for webservers are harder to detect since they tend to blend in with legitimate web traffic. The network was also scanned by an IRC server, probing port 1080, another port associated with proxy servers, see Detect 8.
<b>Inter-correlations:</b>
See CERT vendor note: http://www.cert.org/vul_notes/VN-98.03.WinGate.html
CVE: CVE-1999-0290</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">No. Traffic runs indiscriminately through network.</td></tr>
<tr><td rowspan="5">8. Severity:</td><td>Criticality</td><td>3</td><td>Scan runs through network indiscriminately, few services being run from residential subnet.</td></tr>
<tr><td>Lethality</td><td>2</td><td>Network mapping for active webservers. Vulnerable proxy services allow attackers to use site’s services to launder their network traffic.</td></tr>
<tr><td>System Countermeasures</td><td>3</td><td>Patches mostly up to date. Few network services running. Some systems are better protected than others are.</td></tr>
<tr><td>Network Countermeasures</td><td>1</td><td>No Firewall, but only one route into the network.</td></tr>
<tr><td>Severity</td><td>1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">Filter for traffic to 1080, 8080, (WinGate) and 3128 (Squid Proxy services) at boundary network device.</td></tr>
<tr><td>10. Multiple choice test question, write a question based on the trace and your analysis with your answer.</td><td colspan="3">True or False: TCP:8080 is associated with the WinGate Trojan Horse.

False. TCP:8080 is associated with WinGate, but WinGate is not a Trojan Horse, it is a legitimate service that comes with a vulnerable default configuration. It requires some configuration before it can be used safely. In its vulnerable state WinGate can be used by hostile agents to disguise the source of their attacks on other systems.</td></tr>
</table>

Detect 10: DOS (Smurf & UDP bomb)

http://www.sans.org/y2k/052000.htm – posted by Mike Black

```
- - - - - - - -
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 > 204.17.222.255: icmp: echo request
03:58:15.371826 phwww.netcast.nl.41056 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:18.039447 phwww.netcast.nl.2933 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:20.007210 phwww.netcast.nl.21668 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:22.162075 phwww.netcast.nl.54301 > 204.17.222.255.echo: udp 1024
03:58:23.432190 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 > 204.17.222.255: icmp: echo request
03:58:15.371826 phwww.netcast.nl.41056 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:18.039447 phwww.netcast.nl.2933 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:20.007210 phwww.netcast.nl.21668 > 204.17.222.255.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 > 204.17.222.255: icmp: echo request
03:58:22.162075 phwww.netcast.nl.54301 > 204.17.222.255.echo: udp 1024
03:58:23.432190 phwww.netcast.nl > 204.x.x.0: icmp: echo request
03:58:23.435480 phwww.netcast.nl.23701 > 204.x.x.0.echo: udp 1024
** etc. etc. ***
```

Analysis of Detect 10

1. Source of trace	From the SANS GIAC website: http://www.sans.org/y2k/052000.htm – posted by Mike Black
2. Detect generated by:	Looks like Snort portscan.log or TCPdump
3. Probability of spoofed source address	High. This trace indicates a third-party attack on the apparent source host, netcast.nl.
4. Description of attack:	This attack uses ICMP echo requests to broadcast addresses (Smurf) interleavened with broadcasted UDP:echo traffic (UDP bomb) from a spoofed address to effect a DOS. Alternative diagnosis: Aggressive mapping. From M. Black: "A ping flood DOS attack from phwww.netcast.nl -- been going continuously for 12+ hours now. Random UDP ports."
5. Attack mechanism:	An ICMP echo request or UDP:echo traffic to broadcast addresses can result in all the responsive servers on that subnet responding en masse. Enough traffic sent to the spoofed source address (the intended victim) can lock up systems by overloading the host (killing the server), or jamming their network with garbage traffic (killing the network capabilities). Accurate

<table>
<tr><td></td><td colspan="3">configuration removes this vulnerability.
The attacker alternately sends packets to both broadcast addresses .0 (old BSD and some UNIX broadcast) and .255 (most other Oses). The attacker is also interleavening UDP:echo packets with ICMP echo requests. This increases the likelihood of a system responding to the spoofed echo requests.</td></tr>
<tr><td>6. Correlations:</td><td colspan="3"><b>Intra-correlations:</b> No similar traffic posted from M. Black. No similar traffic on my local network.
<b>Inter-correlations:</b>
CERT:
http://www.cert.org/advisories/CA-98.01.smurf.html
http://www.cert.org/advisories/CA-99-17-denial-of-service-tools.html
CVE:
• CVE-1999-0513: ICMP messages to broadcast addresses are allowed, allowing for a Smurf attack that can cause a denial of service.
• CVE-1999-0103: Echo and chargen, or other combinations of UDP services, can be used in tandem to flood the server, a.k.a. UDP bomb or UDP packet storm.
• CAN-1999-0523: ** CANDIDATE (under review) ** ICMP echo (ping) is allowed from arbitrary hosts.
• CAN-1999-0635: ** CANDIDATE (under review) ** The echo service is running.</td></tr>
<tr><td>7. Evidence of active targeting:</td><td colspan="3">No. There is no evidence that “our” network is being actively targeted. However, the victim of this attack is being very actively targeted.</td></tr>
<tr><td>8. Severity:</td><td>Criticality</td><td>4</td><td>Indiscriminate scan across a network. An array of systems probably exist on this segment.</td></tr>
<tr><td></td><td>Lethality</td><td>3</td><td>This attack is not a large threat to the resources of the intermediary system, but is a large threat to the victim system.</td></tr>
<tr><td></td><td>System Countermeasures</td><td>4</td><td>Mike is keeping track of the activity on his systems, so I’m guessing he’s patching his systems to a reasonable level.</td></tr>
<tr><td></td><td>Network Countermeasures</td><td>4</td><td>Mike is keeping track of the activity on his systems, so I’m guessing he’s actively administering his Inter-networked connections.</td></tr>
<tr><td></td><td>Severity</td><td>1</td><td>Severity = (Criticality + Lethality) – (System Countermeasures + Network Countermeasures)</td></tr>
<tr><td>9. Defensive recommendation:</td><td colspan="3">Take the appropriate countermeasures! This attack can be prevented. ICMP should be properly blocked. Packet filters, firewalls, and routers can all be used to provide protection from these types of attacks. Small UDP services can be disabled and traffic can be filtered to prevent becoming the intermediary in dDOS attacks, too.</td></tr>
<tr><td>10. Multiple choice test question, write a question based on the trace and your analysis with your answer.</td><td colspan="3">This traffic shows the signature of:
A. Ping of Death
B. Smurf Attack
C. Teardrop Attack
D. Echo-Chargen Loop
Ping of death involve fragmented packets, which, when reassembled, are larger than the maximum datagram size, i.e. 65535 bytes. The teardrop attack uses pathological offset of UDP packet fragments (fragments that overlap upon reassembly) which can crash certain systems. Echo-Chargen looping sets two UDP small services into an infinite exchange of Random characters: Echo back. The Smurf Attack uses an intermediary to amplify an</td></tr>
</table>

	attack, shown here. This is the type of traffic reflected by the trace above.