GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Network Monitoring and Threat Detection In-Depth (Security 503)"
at http://www.giac.org/registration/gcia

Dan Evers
SANS 2000 San Jose, CA

Purpose:

This document is submitted to meet the practical requirements of the LevelTwo GCIA certification as specified by SANS.

Preamble:

The SANS Intrusion Detection training course covers all required steps to: setup, configure, collect, and analyze Internet/intranet based intrusions and attacks. In the course of completing this practical I've put my training to the test by performing the following activities:

1) Acquired a PC suitable for the collection of data and subsequent analysis.
2) Installed & configured Mandrake 7 Linux, ipchains, portsentry, tcpdump, ssh, Ethereal Network Analyzer, and snort on said system. I then developed a collection of tcpdump filters and a perl program to exercise each against various collected data records.
3) System was activated and began collecting ip/tcp/udp traffic data on 05/25/00.

About our network:

The IDS system described above is collecting data from the subnet that hosts our external DNS server and two of our web servers. Because of this routing we are only able to detect attacks within a small address space: a.b.c.17/18/19. To make sense of some of my corrective recommendations keep in mind that we have an "up-stream" router and a "down-stream" state-full firewall and our little subnet is nestled in this DMZ. Also, as I must connect to this subnet remotely I run a copy of the BlackIce Defender (BID) system on my remote administration PC and one of the listed detects are from this IDS. This (I'm sure not unique) situation puts into practice Stephen Northcutt's instruction (my paraphrasing) "to go forth and know many IDS report formats!"

Document Contents:

Enough said…On with the captures!

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #1

```
06:16:51.202031 216.34.196.72.2300 > TargetDNS.33434: udp 36 (ttl 242, id 56634)
06:16:51.203098 216.34.196.72.2301 > TargetDNS.33434: udp 36 (ttl 242, id 56635)
06:16:51.203439 216.34.196.72.2302 > TargetDNS.33434: udp 36 (ttl 242, id 56636)
06:18:45.548838 216.34.196.72.2400 > TargetDNS.33434: udp 36 (ttl 242, id 2703)
06:18:45.549537 216.34.196.72.2401 > TargetDNS.33434: udp 36 (ttl 242, id 2704)
06:18:45.550374 216.34.196.72.2402 > TargetDNS.33434: udp 36 (ttl 242, id 2705)
        .
        .
        .
03:48:09.947829 216.34.196.72.2400 > TargetDNS.33434: udp 36 (ttl 242, id 5270)
03:48:09.948775 216.34.196.72.2401 > TargetDNS.33434: udp 36 (ttl 242, id 5271)
03:48:09.949253 216.34.196.72.2402 > TargetDNS.33434: udp 36 (ttl 242, id 5272)
03:52:03.117997 216.34.196.72.2000 > TargetDNS.33434: udp 36 (ttl 242, id 31201)
03:52:03.118679 216.34.196.72.2001 > TargetDNS.33434: udp 36 (ttl 242, id 31202)
03:52:03.119478 216.34.196.72.2002 > TargetDNS.33434: udp 36 (ttl 242, id 31203)
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program.
   b) Format from tcpdump output:
      [time stamp] [source.port]>[dest.port] [protocol bytes] [(time to live #, id #)]

3) Probability the source address was spoofed:

   a) Low: Probe came from Exodus Communications Inc. (NETBLK-ECI-7) and ISP in Santa Clara, CA.

4) Description of attack:

   a) Many pseudo traceroutes to the same address with target port 33434.

5) Attack mechanism:

   a) Possible load balancing by such products as: F5Labs' 3DNS, GTE Hopscotch, and Resonate Global Dispatch.
   b) The default Van Jacobson traceroute starting port is #33434, which means you should never see it as it should be incremented to one by the time it gets to you. The fact that it's showing up with a constant value dictates that these are crafted packets.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #1 continued:

6) Correlations:

   a) Reference "Becky's" SANS Report (see the 2nd paragraph) of Exodus' explanation of this behavior: http://www.sans.org/y2k/031000.htm
   b) Reference correct Van Jacobson traceroute behavior: http://www.robertgraham.com/pubs/firewall-seen.html#traceroute

7) Evidence of active targeting:

   a) Yes. No other similar activity was directed to other systems on this subnet. This was directed to our DNS server.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	5	This was direct against our external DNS server.
Leth:	1	Unlikely to cause any problems on DNS server.
Sys:	5	Modern system, current patches.
Net:	2	Route allows this activity.
Severity:	-1	Secure…For now!

9) Defensive recommendation:

   a) None. This wasn't an attack, just a rude version of "load balancing".

10) Multiple choice test question:

This an example of the following:
   a) One half of a Trojan communications link
   b) udp based denial of service attack
   c) Load balancing
   d) A Van Jacobson based traceroute

Answer: c

© SANS Institute 2000 - 2002 Author retains full rights.

Detect #2

The original alert from tcpdump (Capture Date: 05/28/2000)::

19:24:28.143338 63.71.60.97.859 > a.b.c.17.137: udp 50 (ttl 111, id 44409)
19:24:28.972749 63.71.60.97.860 > a.b.c.18.137: udp 50 (ttl 111, id 45945)
19:24:30.469650 63.71.60.97.860 > a.b.c.18.137: udp 50 (ttl 111, id 46713)
19:24:31.971784 63.71.60.97.860 > a.b.c.18.137: udp 50 (ttl 111, id 46969)
19:24:33.499054 63.71.60.97.861 > a.b.c.19.137: udp 50 (ttl 111, id 47225)
19:24:34.999750 63.71.60.97.861 > a.b.c.19.137: udp 50 (ttl 111, id 47481)
19:24:36.497015 63.71.60.97.861 > a.b.c.19.137: udp 50 (ttl 111, id 47737)

The extraction from Ethereal Network Analyzer (The big picture):

Begin time: `0.000000 seconds`

63.71.60.97 859	a.b.c.17 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.17 137	63.71.60.97 859	NBNS	Name query response unknown
63.71.60.97 32742	a.b.c.17 139	TCP	[SYN] 35020341:0 8192 0
a.b.c.17 139	63.71.60.97 32742	TCP	[SYN, ACK] 1739704347:35020342 32736 0
63.71.60.97 32742	a.b.c.17 139	TCP	[ACK] 35020342:1739704348 8760 0
63.71.60.97 32742	a.b.c.17 139	NBSS	Session request
a.b.c.17 139	63.71.60.97 32742	TCP	[ACK] 1739704348:35020414 32664 0
a.b.c.17 139	63.71.60.97 32742	NBSS	Positive session response
63.71.60.97 32742	a.b.c.17 139	SMB	SMBnegprot Request
a.b.c.17 139	63.71.60.97 32742	TCP	[ACK] 1739704352:35020572 32736 0
a.b.c.17 139	63.71.60.97 32742	SMB	SMBnegprot Response
63.71.60.97 32742	a.b.c.17 139	SMB	SMBsesssetupX Request
a.b.c.17 139	63.71.60.97 32742	SMB	SMBsesssetupX Response
63.71.60.97 860	a.b.c.18 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.18	63.71.60.97	ICMP	Destination unreachable
63.71.60.97 32742	a.b.c.17 139	TCP	[FIN, ACK] 35020699:1739704474 8634 0
a.b.c.17 139	63.71.60.97 32742	TCP	[ACK] 1739704474:35020700 32735 0
a.b.c.17 139	63.71.60.97 32742	TCP	[FIN, ACK] 1739704474:35020700 32736 0
63.71.60.97 32742	a.b.c.17 139	TCP	[ACK] 35020700:1739704475 8634 0
63.71.60.97 860	a.b.c.18 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.18	63.71.60.97	ICMP	Destination unreachable
63.71.60.97 860	a.b.c.18 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.18	63.71.60.97	ICMP	Destination unreachable
63.71.60.97 861	a.b.c.19 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.19	63.71.60.97	ICMP	Destination unreachable
63.71.60.97 861	a.b.c.19 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.19	63.71.60.97	ICMP	Destination unreachable
63.71.60.97 861	a.b.c.19 137	NBNS	Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
a.b.c.19	63.71.60.97	ICMP	Destination unreachable

End time: `8.353858 seconds`

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #2 continued:

Extract of DNS server SAMBA log:

```
-rw-r--r--   root   root        0 Apr 15 13:08 samba-log.celia
-rw-r--r--   root   root        0 Mar 26 04:01 samba-log.clarity
-rw-r--r--   root   root        0 May 25 11:20 samba-log.computer
        .
Many entries later…
        .
-rw-r--r--   root   root        0 Apr 11  1999 samba-log.l232
-rw-r--r--   root   root        0 Apr  6 05:26 samba-log.lozz
-rw-r--r--   root   root        0 May 22 07:47 samba-log.maximus
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program.
   b) Note: Detect actually was much larger, but I've trimmed it down as shown.
   c) Format #1 from tcpdump:
      [timestamp] [source IP.port]>[dest IP.port] [protocol] [size] [(time_to_live #, id #)]
   d) Format #2 from Ethereal Network Analyzer and my alignment formatting:
      [source_IP] [source_port] [dest IP] [dest_port] [protocol] [flags]
      [beg_seq:end_seq] [window_size] [len_size]
      OR
      [source_IP] [source_port] [dest IP] [dest_port] [protocol_bytes]
      [Command_message_display]
   e) Format #3 from DNS samba.log extraction:
      [permissions] [owner] [group] [file_size] [date & time] [log_name]

3) Probability the source address was spoofed:

   a) Low: This is an allocated IP address from an ISP: UUNET Technologies, Inc. (NETBLK-UUNET63) in Fairfax, Virginia.
   b) To be useful to the attacker a response would be needed.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #2 continued:

4) Description of attack:

   a) Multiple sequential udp probes on NetBIOS port 137.
   b) Due to the duration of 8 seconds I'd say this is an automated reconnaissance probe.
   c) If a non-ICPM Destination Unreachable response is given the attack attempts to establish a NetBIOS "datagram" communications link with target.
   d) If contact is established then NetBIOS "session" communications is attempted.

5) Attack mechanism:

   a) Scripted reconnaissance and attack probe: Time between the 1st and 2nd packet is 830ms, all subsequent times average 1.5 seconds. Much too precise for keyboard activity. The total elapse time was just over 8 seconds.
   b) In the following discussion packets of high interest are shown in RED.
   c) The tcpdump filter shown in *The original alert from tcpdump (Capture Date: 05/28/2000)* at top caught my eye in that there was only a single udp packet sent to our DNS server at a.b.c.17 but there was the "what you'd expect" (i.e. 3 udp retries) number of packets sent to the web servers at a.b.c.18 and a.b.c.19. This would indicate that the attacker actually got a response from the DNS server, so I took a closer look at the raw captured data for 05/28/00 and extracted all I/O involving the attackers IP address. The results of which are shown in the section labeled *The extraction from Ethereal Network Analyzer (The big picture).*
   d) We now see that the attacker was able to establish NetBIOS "datagram" communications to port 137 and begins establishing NetBIOS "session" communications. Here's a quick look at two of the NetBIOS commands sent and acknowledged:

      i) SMB SNBnegprot: This is a request to negotiate a protocol variant that will be used the session. The client sends a list of all of the variants that it can speak to the server.
      ii) SMBsesssetupX: Part of the SAMBA "Autologin" process.

   e) Further inspection of the DNS system showed that this wasn't the 1st time a successful attack to port 137 has occurred. As "small" sample of the /var/log/samba.log is shown in the section labeled: *Extract of DNS server SAMBA log*. Here we see that this has been going on for sometime.
   f) A false feeling of security: Even though SAMBA was running on this server it was never configured (i.e. it was in "out of box" default configuration) which, while allowing establishment of a connection the attacker wasn't presented with any juicy file access, and even though the logs show "0" bytes I still think it's possible that the system has been further compromised. Sure, we could argue that if anyone got root access they'd be able to delete or zero-out the files. But, what the heck, I'm erring on the side of paranoia here!

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #2 continued:

6) Correlations:

   a) SANS Text from San Jose 2000: Network-Based Intrusion Detection Analysis (2.4) pages 292 - 293 "Information Gathering: Scanning for In-Use Services".
   b) For a general description of what NetBIOS is go here: http://www.whatis.com/netbios.htm
   c) For a look at the NetBIOS RFC1001 check this out: http://www.cis.ohio-state.edu/htbin/rfc/rfc1001.html
   d) Reference CVE regarding NetBIOS/SHARES (under review): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0520
   e) Reference CVE regarding NetBIOS/SMB shares in default configuration (under review): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0519

7) Evidence of active targeting:

   a) Yes. Appears that the attacker was probing our entire range of hosts IPs.
   b) When attacker found our misconfigured DNS system the attack was modified to take advantage of the situation.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity
   b)

Category:	Rating:	Comments:
Crit:	5	This is our DNS server.
Leth:	3	While SAMBA was running it was still in the "out of the box" default configuration and wasn't allowing much access to the system.
Sys:	0	While this is a modern system with current patches a misconfiguration occurred due to not removing the SAMBA daemons from the startup script.
Net:	2	Could tighten-up the up-stream router (or replace with a state-full firewall) to block access to port 137 (and others) which have no business on this subnet.
Severity:	6	This is bad…This is very bad!

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #2 continued:

9) Defensive recommendation (or, in this case actual actions!):

   a) None for the 2 web servers at a.b.c.18 and a.b.c.19
   b) When I found this I inspected the DNS server and found that it had SAMBA running on it! I've discussed this with the System Administrator and he has since disabled SMB and NMD (SAMBA daemons).
   c) This system is now scheduled for a complete rebuild from scratch ASAP.
   d) Will recommend that up-stream router be replaced with a statefull firewall and block all non-used ports routed to this subnet, or at least, audit the router's ACLs.
   e) Point of interest: A reverse lookup found this to be a class A DSL from Pac Bell address block. As DSL IP addresses are often very long lived it may be possible to back trace this to the attacker.

10) Multiple choice test question:

The trace above is caused by:
   a) nmap OS determination scan on udp ports
   b) Port scan for NetBIOS vulnerabilities
   c) Correct NetBIOS communications
   c) both b) and c)
Answer: d

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #3

```
22:15:23.602113 209.162.39.178.55970 > TargetDNS.80: ack 0 win 1024 (ttl 28, id 42038)
22:17:17.073091 209.162.39.178.58269 > TargetWeb.80: ack 0 win 3072 (ttl 22, id 2603)
22:18:32.603647 209.162.39.178.35098 > TargetWeb.80: ack 0 win 4096 (ttl 35, id 65491)
        .
Many attacks here! All to TargetWeb.19.
        .
22:20:15.944692 209.162.39.178.36316 > TargetWeb.80: ack 0 win 3072 (ttl 26, id 6828)
22:20:21.994838 209.162.39.178.36317 > TargetWeb.80: ack 0 win 3072 (ttl 26, id 25992)
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) Tcpdump and perl filtering program.
   b) Format from tcpdump output:
      [time stamp] [source.port]>[dest.port] [ack seq] [win size] [(time to live #, id #)]

3) Probability the source address was spoofed:

   a) Low: This address belongs to "TheGrid (NETBLK-THEGRID-BLK)" which is a local ISP in San Luis Obispo, CA.
   b) The attacker would want to see the responses to his attack.

4) Description of attack:

   a) Attacker is searching subnet for web servers on port 80 and then begins attack when a victim is located.

5) Attack mechanism:

   a) This behavior has been reported as an nmap ACK scan but alternatively this could be interpreted as an automated DoS attack both in behavior and timing characteristics. Notice that the 1st instance is pointed to our DNS server (TargetDNS.80) but finding no receptive port the host value is incremented and all following activity is directed against our web server (TargetWeb.80). Now that the attacker has the TargetWeb's attention a series of malformed ACK's with 0 payload are sent. This has the effect of "slowing down" a web server. Depending on the type of web server being attacked the slow down may continue for sometime after the attack stops as the buffers clear. Current versions of Apache are not effected past the duration of the attack.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #3 continued:

6) Correlations:
   a) Igor Gashinsky discusses the nmap ACK Scan aspect at: http://www.sans.org/y2k/practical/Igor_Gashinsky.doc
   b) CVE OS Determination (candidate): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0454
   c) My Systems Administrator mentor, Paul Castro, pointed out that he's seen this same behavior being used as a DoS attack as described above. Thanks, Paul!

7) Evidence of active targeting:

   a) Medium probability of active targeting. I believe this was a scripted reconnaissance and DoS probe, but, see detect #4 below for an interesting development!

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity
   b)

Category:	Rating:	Comments:
Crit:	5	A DoS on our Web server would be a "critical situation".
Leth:	4	Some web servers could be DoS'd for extended periods of time.
Sys:	5	Modern web server with latest patches.
Net:	2	Same recommendations at detect #2 above. But we must allow access to this service.
Severity:	2	This is an issue to be addressed ASAP!

9) Defensive recommendation:

   a) None for the target: Web server was not compromised by attack in any way.

10) Multiple choice test question:

This trace is indicative to:
   a) nmap OS determination scan
   b) DoS ACK attack
   c) One side of hidden payload communications
   d) both a) and b)

Answer: d

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #4

23:15:52.947776 209.162.39.178.55950 > TargetDNS.514: udp 0 (ttl 43, id 13521)
23:15:52.804067 209.162.39.178.55951 > TargetDNS.514: udp 0 (ttl 43, id 49128)

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program.
   b) Format from tcpdump output:
   [time stamp] [source.port]>[dest.port] [protocol bytes] [(time to live #, id #)]

3) Probability the source address was spoofed:

   a) Low: This is from TheGrid (NETBLK-THEGRID-BLK) an ISP in San Luis Obispo, CA. Same IP as detect #3.

4) Description of attack:

   a) Probes to port 514 with 0 sized payload looking for a Cisco IOS Syslog vulnerability.
   b) If not for the correlation of the previous detect #3 it would be plausible to attribute this to a WhatsUp probe as this utility can generate 0 byte payloads.
   c) Might this be part of an nmap udp port can? The 0 sized payload is indicative of nmap's behavior…

5) Attack mechanism:

   a) This is part of a series of attacks (see detect #3 above) from the same IP and may part of a scripted attack. This particular attack was captured from a tcpdump filter designed to trigger on the Cisco IOS Syslog vulnerability.
   b) The attacker doesn't yet know what he's attacking as our DNS box is unlikely to have the Cisco IOS vulnerability.

6) Correlations:

   a) Reference Hal Palmeranz at SANS, San Jose, 2000 discussing class text "2.2 intrusion Detection and Packer Filtering: How It Really Works" page108.
   b) CVE Cisco IOS Syslog vulnerability:
   http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0063
   c) Cisco Internal Memo: http://msgs.SecurePoint.com/bugtraq/

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #4 continued:

7) Evidence of active targeting:

   a) Yes. This is 1 of 2 attacks from this specific IP address. See detect #3 above!

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	5	This was directed against our DNS server.
Leth:	1	Not an issue for a non-Cisco system.
Sys:	5	Can't crack what's not there!
Net:	2	Up-stream router will pass this through.
Severity:	-1	Not much a threat.

9) Defensive recommendation:

   a) None: Attack was misdirected and not applicable to the platform being targeted.

10) Multiple choice test question:

This trace shows:
   a) The use of WhatsUp utility.
   b) Probe against the CISCO IOS Syslog vulnerability.
   c) Incorrectly configured router.
   d) udp traceroute signature.
Answer: b

© SANS Institute 2000 - 2002 Author retains full rights.

Detect #5

```
May 29 11:53:59 SourceDNS:53 -> FireWall:56783 UDP
May 29 11:53:59 SourceDNS:53 -> FireWall:56782 UDP
May 29 11:53:59 SourceDNS:53 -> FireWall:56781 UDP
      .
      .
      .
May 29 11:57:30 SourceDNS:53 -> FireWall:56688 UDP
May 29 11:57:30 SourceDNS:53 -> FireWall:56687 UDP
May 29 11:57:30 SourceDNS:53 -> FireWall:56686 UDP
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program AND snort caught this as an ssp_portscan
   b) Format from tcpdump output:
   [time stamp][source:port]>[dest:port][protocol]

3) Probability the source address was spoofed:

   a) None. This is our DNS server!

4) Description of attack:

   a) At fist glance this made my blood run cold! Could it be that our DNS server has been compromised and is being used to probe our firewall? Look at those sequential ports values (also, consider detect #2 for a insight to my "emotional" context)!

5) Attack mechanism:

   a) None as this was a false positive. This is a normal response to our FireWall that is doing NAT and allowing a DNS Zone transfer from our external DNS system to our internal DNS system. The usage of very high ports is symptomatic of our Novell Boarder Manager firewall (and, I'm told also of ipchains and several other firewalls). These port values are generated from a "pool" of available ports.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #5 continued:

6) Correlations:

   a) Regarding port NAT assignment: Read-up on Novell Board Manager behavior and you will notice that while the Dynamic mode of operation is specified as "Random" the Static mode of operation is not specified, and it is possible to run in both modes concurrently. After much discussion with the firewall administrator I have allowed him to convince me that this is "normal behavior". But, if you want to build-up (or tear-down) your own conclusions visit: http://support.novell.com and do a search on NAT and Boarder Manager.
   b) This above trace corresponds with further investigation of communications between these 2 systems via tcpdump.

7) Evidence of active targeting:

   a) Not applicable.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	5	Actions involving firewall and DNS server
Leth:	1	Appeared to be a port scan of fire wall. But wasn't!
Sys:	5	Normal allowable traffic by DNS server.
Net:	5	Normal behavior by down-stream firewall.
Severity:	-4	Not a problem!

9) Defensive recommendation:

   a) None for defensive, however, I will continue to monitor port 53 activity from our DNS to our FireWall server.
   b) Also, recommended to the firewall system administrator to modify configuration to allow for random port assignment as predictability is a liability.

10) Multiple choice test question:

The above trace is caused by:
   a) Port scan by compromised DNS server
   b) Normal NAT address assignment by certain firewalls
   c) Symptom of GrandWazoo Trojan activity
   d) None of the above

Answer: b

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #6

The Trigger (caught by a Doom666 tcpdump filter):
(Reference Format #1)

19:22:49.735667 216.224.158.199.1403 > a.b.c.**17**.666: S 3015241679:3015241679(0) win 32428 <mss 1474,sackOK,timestamp 1505617[|tcp]> (DF) (ttl 48, id 12902)
.
.
.
19:39:47.636096 216.224.158.199.4099 > a.b.c.**17**.666: S 4104233400:4104233400(0) win 32428 <mss 1474,sackOK,timestamp 1607377[|tcp]> (DF) (ttl 48, id 23638)

The Bigger Picture (brought to you by the ethereal viewer):
(Reference Format #2)

```
19:22:38.0059   216.224.158.199 53412 a.b.c.17 80  TCP [ACK]
19:22:43.6818   216.224.158.199 1110  a.b.c.17 315 TCP [SYN]
19:22:43.6967   216.224.158.199 1111  a.b.c.17 79  TCP [SYN]
.
.
.
19:23:06.7652   216.224.158.199 2595  a.b.c.17 143 TCP [ACK]
19:23:06.7664   216.224.158.199 2595  a.b.c.17 143 TCP [FIN, ACK]
19:23:06.9967   216.224.158.199 53412 a.b.c.18 80  TCP [ACK]
19:23:08.8221   216.224.158.199 2364  a.b.c.17 109 TCP [ACK]
19:23:10.4402   216.224.158.199 2474  a.b.c.17 513 TCP [ACK]
19:23:12.0421   216.224.158.199 2595  a.b.c.17 143 TCP [ACK]
19:23:12.6080   216.224.158.199 2623  a.b.c.18 315 TCP [SYN]
19:23:12.6159   216.224.158.199 2624  a.b.c.18 79  TCP [SYN]
19:23:12.6818   216.224.158.199 2625  a.b.c.18 708 TCP [SYN]
19:23:12.6825   216.224.158.199 2626  a.b.c.18 701 TCP [SYN]
.
.
.
19:48:42.4963   216.224.158.199  ---  a.b.c.17     ICMP  Echo (ping) request
19:54:09.2584   216.224.158.199 2960  a.b.c.19  0  TCP  [FIN, SYN]
.
.
.
19:54:13.2170   216.224.158.199 2964  a.b.c.19  0   TCP [FIN, SYN]
19:54:45.0759   216.224.158.199 2335  a.b.c.18  0   TCP [FIN, SYN]
.
.
.
19:54:49.0749   216.224.158.199 2339  a.b.c.18  0    TCP   [FIN, SYN]
19:55:06.7152   216.224.158.199 2724  a.b.c.17  0    TCP   [FIN, SYN]
.
.
.
19:56:44.4501   216.224.158.199 1619  a.b.c.17  0    TCP   []
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #6 continued:

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program and Ethereal Network Analyzer.
   b) Format #1 tcpdump filter output:
      [time stamp] [source.port]>[dest.port] [flags] [beg seq:end seq(size)]
      [window size] [<additional payload info[|protocol]> [flag] [(time to live #, id #)]
   c) Format #2 Ethereal Network Analyzer filter output:
      [time stamp] [source IP] [source port] [dest. IP] [dest. Port] [protocol] [flags]
   d) NOTE: This is a **drastically truncated** sample both in the number of lines shown and the length of the lines (We're talking 11929 lines here!).
   e) Addition information extracted using ethereal filtering for the attacker's IP address using Ethereal Network Analyzer.

3) Probability the source address was spoofed:

   a) Low. This IP is from a service provider
   b) The attacker will want to see any responses to this attack.

4) Description of attack:

   a) Originally this was detected using a tcpdump filter configured to trigger on Doom at port 666 activity and this is shown in the 1st trace sample (there where 10 lines).
   b) This led me to test for all activity by this IP address. A very truncated sample of the results is shown in the 2nd trace sample. This sample shows that over a time period of 34 minutes and 6 seconds a total of 11922 packets where sent to all of the systems on this subnet. This included a variety of ports and flag combinations to each system.

5) Attack Mechanism:

   a) This is truly a weird signature! On one level this seems reminiscent of an nmap OS detection probe, but given the that this was directed to **all systems** on our subnet, the variety of ports tested, and the bouncing around of different hosts being probed, I wouldn't be surprised if this was a "home made" script of indeterminate origin…Looks like someone said "Say, I know all these possible exploits, why not put them all together, add a random generator, and a bracketing range value for the IP addresses and…Whala!"
   b) The famous SYN/FIN attack is nicely represented.
   c) For clarity in the 2nd sample transitions between hosts shown are in **bold.**

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #6 continued:

6) Correlations:

   a) Vicki Irwin at SANS San Jose 2000 Referencing training text "2.2 Intrusion Detection and Packet Filtering: How It Really Works" pages 172-176 "nmap OS Determination"
   b) Reference CVE for SYN/FIN attacks: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0116

7) Evidence of active targeting:

   a) Yes! This is directed against every host on our subnet. It was a thoroughly comprehensive reconnaissance probe.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	3	Indiscriminate attack against our entire subnet.
Leth:	1	Unlikely to succeed in a system compromise.
Sys:	5	Target systems are current and patched.
Net:	2	Recommend auditing the up-stream router ACLs with the goal of tightening access to only services used on this subnet.
Severity:	-3	Not an issue.

9) Defensive recommendation:

   a) None, except for verifying with the system administrator that all no essential services are shutdown.
   b) Take note of the attacker's IP and keep an eye on this joker!

10) Multiple choice test question:

This trace is typical of:
   a) Normal user behavior
   b) nmap OS Determination
   c) Scripted reconnaissance and attack probe.
   d) Traceroute reverse scan load balancing

Answer: c

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #7

The Friendly Greeting (Reference Format #1):
19:40:02.201109 216.224.158.199.61714 > TargetDNS.21: S 3163148033:3163148033(0) win 3072 <wscale 10,nop,mss 265,timestamp 1061109567[|tcp]> (ttl 38, id 19896)

19:40:02.201865 TargetDNS.21 > 216.224.158.199.61714: S 647653060:647653060(0) ack 3163148034 win 32736 <mss 265> (ttl 64, id 58618)

The Stab In The Back:
19:40:02.256209 216.224.158.199.61716 > TargetDNS.21: SFP 3163148033:3163148033(0) win 3072 urg 0 <wscale 10,nop,mss 265,timestamp 1061109567[|tcp]> (ttl 38, id 40257)

19:40:02.315102 216.224.158.199.61720 > TargetDNS.35170: FP 3163148033:3163148033(0) win 3072 urg 0 <wscale 10,nop,mss 265,timestamp 1061109567[|tcp]> (ttl 38, id 13557)

A little delay then 20 @ Bogus Packets (Reference Format #2):
19:56:25.461007 216.224.158.199.1600 > TargetDNS.0: . win 512 (ttl 48, id 15910)
19:56:26.468905 216.224.158.199.1601 > TargetDNS.0: . win 512 (ttl 48, id 5765)
19:56:27.489327 216.224.158.199.1602 > TargetDNS.0: . win 512 (ttl 48, id 51873)
19:56:28.518029 216.224.158.199.1603 > TargetDNS.0: . win 512 (ttl 48, id 64258)
19:56:29.484554 216.224.158.199.1604 > TargetDNS.0: . win 512 (ttl 48, id 245)
19:56:30.455353 216.224.158.199.1605 > TargetDNS.0: . win 512 (ttl 48, id 44265)
19:56:31.448033 216.224.158.199.1606 > TargetDNS.0: . win 512 (ttl 48, id 39468)
19:56:32.451832 216.224.158.199.1607 > TargetDNS.0: . win 512 (ttl 48, id 27458)
19:56:33.491793 216.224.158.199.1608 > TargetDNS.0: . win 512 (ttl 48, id 10654)
19:56:34.520920 216.224.158.199.1609 > TargetDNS.0: . win 512 (ttl 48, id 15247)
19:56:35.486329 216.224.158.199.1610 > TargetDNS.0: . win 512 (ttl 48, id 32821)
19:56:36.469112 216.224.158.199.1611 > TargetDNS.0: . win 512 (ttl 48, id 50771)
19:56:37.449619 216.224.158.199.1612 > TargetDNS.0: . win 512 (ttl 48, id 27435)
19:56:38.450643 216.224.158.199.1613 > TargetDNS.0: . win 512 (ttl 48, id 20358)
19:56:39.486764 216.224.158.199.1614 > TargetDNS.0: . win 512 (ttl 48, id 3791)
19:56:40.500476 216.224.158.199.1615 > TargetDNS.0: . win 512 (ttl 48, id 42646)
19:56:41.494993 216.224.158.199.1616 > TargetDNS.0: . win 512 (ttl 48, id 59867)
19:56:42.450480 216.224.158.199.1617 > TargetDNS.0: . win 512 (ttl 48, id 32810)
19:56:43.453886 216.224.158.199.1618 > TargetDNS.0: . win 512 (ttl 48, id 16083)
19:56:44.450136 216.224.158.199.1619 > TargetDNS.0: . win 512 (ttl 48, id 35477)

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program.
   b) Format #1 from tcpdump output:
      [time stamp] [source.port]>[dest.port] [flags] [beg seq:end seq(size)]
      [window bytes] [flags] [data size] [<additional payload info[|protocol]>
      [(time to live #, id #)]
   c) Format #2 from Ethereal Network Analyzer:
      [time stamp] [source.port]>[dest.port] [window bytes] [(time to live #, id #)]

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #7 continued:

3) Probability the source address was spoofed:

   a) Low. More activity from our friends at TheGrid (NETBLK-THEGRID-BLK2) down in San Luis Obispo. But, hey, this time it's a different IP address (Reference detects #3, and #4)!

4) Description of attack:

   a) A friendly ftp request followed by a few impossible flags and a series of 20 bogus packets with no flags set to port 0.

5) Attack mechanism:

   a) The goal of this attack is to initiate a correct protocol (i.e. ftp) session then confuse the system by sending the SYN/FIN/PSH/URG packet to cause an ftp buffer overrun so that the attacker will be given access privileges. Keep in mind that the "PSH" flag tells the receiver NOT to buffer the data before passing it on to the application! The attack starts out with an acceptable FTP control channel (Port 21) request that is replied to from TargetDNS with an ACK. But instead of finishing the initial handshake the culprit turns around with a SYN/FIN/PSH/URG packet to port 21 and then quickly sends Fin/Push/Urgent packet to the high valued ethereal port #35170. Then, a few minutes later, this is followed by 20 bogus packets with no flags set to port 0.
   b) While I can't say that the 20 packets sent to port 0 follow in a timely manner it is possible that they are related in that the 1st set of crafted packets might contain code that, if executed, would open services to port 0. OR
   c) It could be that the attacker is trying to figure out what kind of system he's attacking. On some systems port 0 is "invalid" and may produce odd results which may facilitate O/S determination.

6) Correlations:

   a) Reference Hal Pomeranz speaking of the SANS class text "2.2 Intrusion Detection and Packet Filtering: How It Really Works" page 18 TCP Segment Flags".
   b) Reference port 0 probes: http://www.robertgraham.com/pubs/firewall-seen.html#1.1
   c) Reference ftp buffer overruns used to gain access to attacker specified ports: http://www.securiteam.com/exploits/Extending_the_FTP__ALG__vulnerability_to_any_FTP_client.html

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #7 continued:

7) Evidence of active targeting:

   a) Yes! This was directed at our DNS box. A subsequent search of this source IP shows no other attacks to our subnet.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	5	Attack was directed against our DNS server.
Leth:	5	A successful attack on our DNS would give attacker access to our system from a trusted host.
Sys:	5	DNS server is current and patched. Portsentry blocked IP access via ipchains.
Net:	2	Router is passing this through.
Severity:	3	While this attack was not successful the ability to do OS detection is partially due to the up-stream router's current configuration. Again, recommend installing a statefull firewall at that location.

9) Defensive recommendation:

   a) For the target: None as box was not compromised by attack, but will recommend to Systems Administrator to audit box for unneeded services.
   b) Recommend auditing the up-stream router ACLs with the goal of tightening access to only services used on this subnet.
   c) For IDS: Take note of this IP and keep an eye on it!

10) Multiple choice test question:

This trace is typical of:
   a) An ftp session gone horribly awry.
   b) Load balancing.
   c) Normal FTP "back channel" activity.
   d) An FTP buffer overrun attack.

Answer: d

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #8

Original Capture:

```
20:16:52.5959   24.5.193.65  TargetBID UDP  1376 > 18753
20:16:53.1009   24.5.193.65  TargetBID UDP  1376 > 18753
20:19:47.3389   24.5.193.65  TargetBID UDP  1376 > 18753
20:19:47.4989   24.5.193.65  TargetBID UDP  1376 > 18753
20:19:55.5110   24.5.193.65  TargetBID UDP  1376 > 34555
20:19:55.5800   24.5.193.65  TargetBID UDP  1376 > 34555
20:20:00.5650   24.5.193.65  TargetBID UDP  1376 > 34555
20:20:09.6019   24.5.193.65  TargetBID UDP  1376 > 27444
20:20:09.6019   24.5.193.65  TargetBID UDP  1376 > 27444
20:20:14.6419   24.5.193.65  TargetBID UDP  1376 > 27444
20:20:14.7469   24.5.193.65  TargetBID TCP  36824 > 27665 [SYN] 114957151  0  1024  0
20:20:29.3229   24.5.193.65  TargetBID TCP  56985 > 16660 [SYN] 874683303  0  1024  0
20:20:35.8079   24.5.193.65  TargetBID TCP  37566 > 65000 [SYN] 253032809  0   2048 0
20:20:45.1849   24.5.193.65  TargetBID ICMP  Echo (ping) reply
20:20:53.2740   24.5.193.65  TargetBID ICMP  Echo (ping) reply
20:12:35.1649   24.5.193.65  TargetBID ICMP  Echo (ping) reply
20:12:43.2340   24.5.193.65  TargetBID ICMP  Echo (ping) reply
```

Data Payload Close Up:

```
20:12:35.1649   24.5.193.65  TargetBID ICMP  Echo (ping) reply

  0  4445 5354 0000 2053 5243 0000 0800 4500   DEST.. SRC....E.
 10  0036 c30e 0000 3501 7f53 1805 c141 d8e0   .6....5..S...A..
 20  913e 0000 b413 029a 0000 0000 0000 0000   .>..............
 30  0000 0000 0000 0000 0000 0000 0000 736b   ..............sk
 40  696c 6c7a                                 illz

20:12:43.2340   24.5.193.65  TargetBID ICMP  Echo (ping) reply

  0  4445 5354 0000 2053 5243 0000 0800 4500   DEST.. SRC....E.
 10  003b c317 0000 3501 7f45 1805 c141 d8e0   .;....5..E...A..
 20  913e 0000 c14a 029c 0000 0000 0000 0000   .>...J..........
 30  0000 0000 0000 0000 0000 0000 0000 6765   ..............ge
 40  7375 6e64 6865 6974 21                    sundheit!
```

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #8 continued:

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) BlackIce Defender.
   b) Formats from Ethereal Network Analyzer:
      UDP: [time stamp] [source IP] [dest IP Name] [protocol] [src port]>[dst port]
      TCP: [time stamp] [source IP] [dest IP Name] [protocol] [src port]>
      [dst port] [flags] [beg seq] [end seq] [bytes] [window size] [len size]
      ICMP: [time stamp] [source IP] [dest IP Name] [protocol] [service description]

3) Probability the source address was spoofed:

   a) Low. This is an IP from @Home Network (NETBLK-RDC1-SFBA-1) an ISP located in Redwood City, CA.
   b) The attacker would need to receive the responses to the different Trin00 probes to be useful.

4) Description of attack:

   a) A series of Trin00 probes using different protocols followed by a couple of "signature" echo replies with their altered payload.

5) Attack mechanism:

   a) I believe this is not an automated script attack because the average time between packets is totally inconsistent and would allow enough time to manually enter information from a keyboard between attack sets.
   b) The attack is completely oriented around Trin00 and some of its variants.
   c) The ending two packets are unsolicited echo replies that contain the signature "skillz" and "gesundheit!" payloads typical of some of the Trin00 variants.
   d) Quick Overview: A distributed DoS attack relies on daemons that are installed on many compromised hosts. A client is used to identify a target to the daemons and each compromised system then can launch a DoS attack (usually using flood-like attacks i.e. UDP, ICMP, SYN) which can cripple a targeted network or host.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #8 continued:

6) Correlations:

   a) Payload "skillz" and "gesundheit!" and Master Trin00 Activity, Reference: http://www.sans.org/y2k/stacheldraht.htm
   b) Port #18753, Reference "Shaft handler to Agent": http://www.doshelp.com/Documents/shaft.txt
   c) Port #34555, Reference: http://www.sans.org/y2k/030100.htm
   d) Port #27444, Reference "Trin00/TFN2K/UDP": http://nethog.net/feeds/niteryder/trojans.htm
   e) CVE (Candidate) Trin00: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0138

7) Evidence of active targeting:

   a) Yes! This was one of several directed attacks against my remote administration system from a specific IP address over a period of several days.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	4	This was directed against my Win 95 remote administration system. I give it a 4 due to the potential for damage.
Leth:	5	A compromised system would allow attacks to other systems and possible compromise of our entire network.
Sys:	5	System is current with latest MS security patches.
Net:	5	The BlackIce Defender software works very well.
Severity:	-1	All is quiet on the western front!

9) Defensive recommendation:

   a) None. This target system caught and blocked the attacker.

10) Multiple choice test question:

This trace is typical of:
   a) One half of MS-Mail communications.
   b) Trin00 Master activity.
   c) Trolling for Trojans.
   d) Normal communications diagnostic probes.

Answer: b

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #9

Sample of beginning of complete capture:

```
0.000000  64.7.9.34  a.b.c.17  TCP  3062 > 111 [SYN]          3075749085:0                 Win=32120 Len=0
0.000668  a.b.c.17 64.7.9.34   TCP  111 > 3062 [RST, ACK] 0:3075749086                  Win=0     Len=0
0.071001  64.7.9.34   a.b.c.18 TCP  3063 > 111 [SYN]          3066476159:0                 Win=32120 Len=0
0.000363  a.b.c.18  64.7.9.34  TCP  111>3063   [SYN, ACK] 2799118744:3066476160 Win=32120 Len=0
0.008295  64.7.9.34   a.b.c.19 TCP  3064 > 111 [SYN]          3064758821:0                 Win=32120 Len=0
```

*Extract of targeted web server at a.b.c.**18**:*

```
0           64.7.9.34  a.b.c.18 TCP 3063 > 111  [SYN]       3066476159:0                 Win=32120  Len=0
0.000363  a.b.c.18  64.7.9.34 TCP 111 > 3063  [SYN,ACK]  2799118744:3066476160  Win=32120  Len=0
0.133858  64.7.9.34  a.b.c.18 TCP 3076 > 111  [SYN]       3067028907:0                 Win=32120  Len=0
0.134016  a.b.c.18  64.7.9.34 TCP 111 > 3076  [SYN,ACK]  2803520230:3067028908  Win=32120  Len=0
0.274708  64.7.9.34  a.b.c.18 TCP 3076 > 111  [ACK]       3067028908:2803520231  Win=32120  Len=0
0.280218  64.7.9.34  a.b.c.18 TCP 3076 > 111  [PSH,ACK]  3067028908:2803520231  Win=32120  Len=44
0.280372  a.b.c.18  64.7.9.34 TCP 111 > 3076  [ACK]       2803520231:3067028952  Win=32120  Len=0
3.355321  a.b.c.18  64.7.9.34 TCP 111 > 3063  [SYN,ACK]  2799118744:3066476160  Win=32120  Len=0
3.491311  64.7.9.34  a.b.c.18 TCP 3063 > 111  [ACK]       3066476160:2799118745  Win=32120  Len=0
5.292478  a.b.c.18  64.7.9.34 TCP 111 > 3076  [RST,ACK]  2803520231:3067028952  Win=32120  Len=0
5.343715  a.b.c.18  64.7.9.34 TCP 111 > 3063  [FIN,ACK]  2799118745:3066476160  Win=32120  Len=0
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   a) tcpdump and perl filtering program.
   b) Format, from Ethereal Network Analyzer and my alignment:
      [time from beginning of capture] [source IP] [dest. IP] [protocol]
      [source.port > dest.port] [flags] [beg. seq:end seq] [window size] [len. size]

3) Probability the source address was spoofed:

   a) Low. This address belongs to MegaPath Networks Inc. (NETBLK-MEGAPATH-BLK-1) local ISP located in Pleasanton, CA.
   b) The attacker would need to get responses back in order to use the attack results.

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #9 continues:

4) Description of attack:

   a) Each host on our subnet (a.b.c.17, 18, 19) where probed at the SunRPC services on port 111. When a response was stimulated then a series of packets where sent to the responsive ports.
   b) The our DNS system at a.b.c.17 promptly sent back a RST and the attacker lost all interest in it and started concentrating packets to our web servers at a.b.c.18 and a.b.c.19.
   c) I've extracted only the a.b.c.18 information for clarity, but it's analogous to the a.b.c.19 activity.

5) Attack mechanism:

   a) The low time between the initial packets indicates that this is an automated attack. Notice that our system is "slowing" down the transactions. This in normal behavior by our system when attack behavior is detected.
   b) Once the attacker has identified responsive systems he begins sending many packets to port 111 in hopes go gain access to the SunRPC services.
   d) Upon closer inspection we see that the attacker is sending many SYN packets with different initial sequence numbers and our system is responding to each as a different thread. I've isolated a single thread initiated from the attacker as shown in RED in the *Extract of targeted web server at a.b.c.18* sample. We see a normal SYN followed by a SYN/ACK from our system, but the attacker keeps using the same sequence number then sends a packet with the PSH/ACK flags set. This is also the only packet with a payload (it is 44 bytes long), our system ACK's this weird packet, and about 5 seconds later we see the connection RST by our system.
   e) It took a while for our system to respond to the malicious packet of Len = 44. So, in one sense this could be considered a DoS attack, but what I think was really going on is that the attacker was hoping to cause an RPC buffer overrun and gain root access to the RPC services. I also would bet that the payload of Len = 44 packet contains malicious code.
   f) RPC services include: NFS, the Network File System, and NIS, the Network Information System, RPCBind and others. Obviously gaining trusted access to these services would be a goldmine for any hacker, both for pilfering data and causing mayhem.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #9 continues:

6) Correlations:

   a) Several attacks on port 111 where reported to SANS on 060900. To see them go here: http://www.sans.org/y2k/060900.htm
   b) For information on Portmapper and Rpcbind (port 111) see the SANS Information Security Paper: "Rpcbind and Portmapper" by David P. Reece located at: http://www.sans.org/newlook/resources/IDFAQ/blocking.htm
   c) For information on RPC and the portmapper daemon see: http://www.uwsg.iu.edu/usail/network/services/portmapper.html

7) Evidence of active targeting:

   a) Yes. This was targeted against our subnet.
   b) When a web server the web servers at a.b.c.18 and a.b.c.19 where found the attack was concentrated against these systems. The DNS system at a.b.c.17 was skipped.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity
   b)

Category:	Rating:	Comments:
Crit:	5	Our DNS and web servers are critical!
Leth:	5	Could gain trusted access to file systems, etc.
Sys:	5	All current and patched.
Net:	3	The up-stream router should be configured to block this and other unused services.
Severity:	2	This is an issue!

9) Defensive recommendation:

   a) For the target systems: none! As this attack was handled correctly and blocked.
   b) For the up-stream router: audit for blocking unused port access or replace with a statefull firewall system.

10) Multiple choice test question:

This trace is typical of:
   a) An RPC attack.
   b) Normal RPC communications.
   c) A malformed packet sequence attack.
   d) A “hijacked” RPC session.

Answer: a

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #10

Trigger Packets (Reference format #4):

09:16:57.0060 192.115.216.142 209.79.222.18 NBNS 137 > 137 Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)

(Reference format #1):

09:16:57.0062 209.79.222.18 192.115.216.142 ICMP Destination unreachable

Sample #1 (Reference Format #1):

09:05:41.5495 192.115.216.142 a.b.c.18 ICMP Echo (ping) request
09:05:41.5498 a.b.c.18 192.115.216.142 ICMP Echo (ping) reply
09:05:41.5504 192.115.216.142 a.b.c.17 ICMP Echo (ping) request
09:05:41.5512 a.b.c.17 192.115.216.142 ICMP Echo (ping) reply
09:05:41.5517 192.115.216.142 a.b.c.19 ICMP Echo (ping) request
09:05:41.5518 a.b.c.19 192.115.216.142 ICMP Echo (ping) reply
.

A 10 minute break…

.

Sample #2 (Reference Format #2):

09:16:54.6941 192.115.216.142 a.b.c.17 TCP 1017 > 12346 [SYN] 723558:0 Win=8192 Len=0
09:16:54.6947 a.b.c.17 192.115.216.142 TCP 12346 > 1017 [RST, ACK] 0:723559 Win=0 Len=0
09:16:54.6946 192.115.216.142 a.b.c.18 TCP 1018 > 12346 [SYN] 723573:0 Win=8192 Len=0
09:16:54.6947 a.b.c.17 192.115.216.142 TCP 12346 > 1017 [RST, ACK] 0:723559 Win=0 Len=0
09:16:54.6948 a.b.c.18 192.115.216.142 TCP 12346 > 1018 [SYN, ACK] 3925449965:723574 Win=32120 Len=0
09:16:54.7267 192.115.216.142 a.b.c.19 TCP 1019 > 12346 [SYN] 723588:0 Win=8192 Len=0
09:16:54.7268 a.b.c.19 192.115.216.142 TCP 12346 > 1019 [SYN, ACK] 3930116322:723589 Win=32120 Len=0
09:16:55.2540 192.115.216.142 a.b.c.18 TCP 1018 > 12346 [ACK] 723574:3925449966 Win=2920 Len=0
09:16:55.2894 192.115.216.142 a.b.c.19 TCP 1019 > 12346 [ACK] 723589:3930116323 Win=2920 Len=0
09:16:55.3581 a.b.c.18 192.115.216.142 TCP 12346 > 1018 [FIN, ACK] 3925449966:723574 Win=32120 Len=0
09:16:55.5576 192.115.216.142 a.b.c.18 NBNS Name query unknown Illegal NetBIOS name (character not between A and Z in first-level encoding)
09:16:55.5579 a.b.c.18 192.115.216.142 ICMP Destination unreachable
09:16:55.8619 192.115.216.142 a.b.c.17 TCP 1017 > 12346 [SYN] 723558:0 Win=8192 Len=0
09:16:55.8625 a.b.c.17 192.115.216.142 TCP 12346 > 1017 [RST, ACK] 0:723559 Win=0 Len=0
09:16:56.0936 192.115.216.142 a.b.c.18 TCP 1018 > 12346 [ACK] 723574:3925449967 Win=8760 Len=0
.

Many packets later…

.

09:41:08.3157 192.115.216.142 a.b.c.19 TCP 4183 > 12345 [ACK] 727215:3949439253 Win=8760 Len=0
09:42:02.5903 a.b.c.19 192.115.216.142 TCP 12346 > 1019 [FIN, ACK] 3930116323:723590 Win=32120 Len=0
09:42:03.0251 192.115.216.142 a.b.c.19 TCP 1019 > 12346 [ACK] 723590:3930116324 Win=8760 Len=0

Portsenty log as related to port #12346 probe (Reference format #3):

```
06/09/2000 09:17:58 Host: 192.115.216.142 Port: 12346 TCP Blocked
```

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #10 continued:

Data extraction #1: typical of all icmp echo requests and replies:

```
   0  00a0 c9cf e55b 00a0 c9b0 2623 0800 4500   .....[....&#..E.
  10  004e 452d 0000 7201 bb1d c073 d88e d14f   .NE-..r....s...O
  20  de12 0800 3da5 0200 9721 5768 6174 7355   ....=....!WhatsU
  30  7020 2d20 4120 4e65 7477 6f72 6b20 4d6f   p - A Network Mo
  40  6e69 746f                                 nito
```

Data extraction #2: typical of all NBNS packets sent to port 137:

```
   0  00a0 c9cf e55b 00a0 c9b0 2623 0800 4500   .....[....&#..E.
  10  004e fd7d 0000 7211 02bd c073 d88e d14f   .N.}..r....s...O
  20  de12 0089 0089 003a dde3 99d6 0010 0001   .......:........
  30  0000 0000 0000 2043 4b41 4141 4141 4141   ...... CKAAAAAAA
  40  4141 4141                                AAAA
```

Data extraction #3: typical of final packets sent to both a.b.c.18 and a.b.c.19:

```
   0  00a0 c9cf e55b 00a0 c9b0 2623 0800 4510   .....[....&#..E.
  10  0028 1bba 4000 7206 a4a0 c073 d88e d14f   .(..@.r....s...O
  20  de13 03fb 303a 000b 0a86 ea40 d0e4 5010   ....0:.....@..P.
  30  2238 4b4b 0000 0762 616e 6e65             "8KK...banne
```

1) Source of trace:

   a) My network.

2) Detect was generated by:

   b) Format #1 from Ethereal Network Analyzer extraction:
   [time stamp] [source IP] [dest. IP] [protocol] [description]
   c) Format #2 from Ethereal Network Analyzer extraction:
   [time stamp] [source IP] [dest. IP] [protocol] [source port > dest. port] [flags] [beg. seq.:end seq.] [window size] [len.size]
   d) Format #3 from "portsentry.history" log extraction:
   [date & timestamp] [Host: source IP] [Port: dest. port] [protocol] [action]
   e) Format #4 from Ethereal Network Analyzer:
   [time stamp] [source IP] [dest. IP] [source port > dest. port] [protocol] [description]

3) Probability the source address was spoofed:

   a) Low. The attack would need to see the response to make use of this attack.
   b) Israeli Network Information Center (NETBLK-ISRAELC-BLOCK) ISRAEL Internet Society of Israel (ISOC-IL).
   c) Tracking down this particular IP may be problematic due to geographical separation

© SANS Institute 2000 - 2002, Author retains full rights.

Detect #10 continued:

4) Description of attack:

   a) A reconnaissance probe using WhatsUp (Reference the 1st data extraction) is launched against all hosts on our subnet.
   b) Several minutes later a series of packets are sent to all hosts on our subnet to the NetBus trojan ports at #12345 and #12346, and occasionally port #137.
   c) A successful attack would give the attacker control of a trusted system through trojan horse remote control.

5) Attack mechanism:

   a) The packets shown above labeled as *Trigger Packets* where detected from a tcpdump filter that keys on NetBIOS port #137 activity. A dissection of probe is discussed below on item e). The subsequent packets were extracted by keying on the attacker IP address for that days' collected data.
   b) Unlike the NetBIOS probe discussed in detect #2 this attack differs in 3 basic ways:
      i) The preliminary scan by WhatsUp. Reference the 1st sample.
      ii) The variety of ports probed for NetBIOS trojans. Reference the 2nd sample where, for clarity, only attacks to port #12346 have been extracted).
      iii) The system responses to the probes show correct blocking.
   c) This is effectively a set of 3 concurrent attacks. To simplify this situation I will divide this into 2 sets of attacks as follows:
      i) a.b.c.17 as DNS server, and
      ii) a.b.c.18 and a.b.c.19 which are both web servers.
      To further simplify this analysis: As a.b.c.17 is handling the situation by returning a RST we need only concentrate on analyzing the a.b.c.18 web server as it's representative of both web servers in this attack. Reference the RED packets to follow the thread of the attack.
   d) Notice in *Sample #1* that when an attempt is made to communicate on port 137 (NBNS: NetBIOS Name Services) an ICMP response is stimulated. Inspection of the ICMP packet showed that the Type = 0x03 (Destination Unreachable) and Code = 0x03 (Port Unreachable) which is what I'd expect to see as a response to a port with no services available. Further dissection of the ICMP packet showed nothing unusual and this is an expected response to a IP packet sent to a closed port, unfortunately, only 8 bytes of the embedded originating IP header where available for inspection and nothing untold was seen here.
   e) A closer inspection of the triggering packet *(Trigger Packets)* sent to port 137 shows that the attacker had set the Broadcast flag (reference *data extraction #2*, shown in RED) in the hope that if the packet gets through that the attack will be propagated to all other NetBIOS systems on that subnet.

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #10 continued:

f) One interpretation of the packets sent to port 137 is to exploit a weakness in WINS and cause a DoS. But since this activity was limited to the occasional packet I think this was just part of the reconnaissance aspect of this attack.
g) The packets sent to both the popular NetBus trojan ports at #12345 and #12346 are producing a response, but the attacker was not able to establish communications as there are no services on these ports. Why, if there's no services running on these ports, is there a complete 3-way handshake? Because we are running portsentry on these systems and, if you'll notice, the 3-way handshake is always terminated by our targeted system with a FIN/ACK. The portsentry log (reference *Portsentry log sample*) shows that this has been blocked.
h) Lastly, the response time to the stimulus of the attacker is slowing down tremendously (I noticed this when inspecting the packets with the Ethereal Network Analyzer in "Time from previous packet" mode). This is caused by our systems internal defenses.
i) Reference data extraction #3. Notice the "8KK ..banne"? I have no clue if this means anything, just thought it was interesting!

6) Correlations:

d) Reference CVE Denial of services on WINS at: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0288
e) A example of both ports #12345 & #12346 was sent to SANS by Laurie (in edu land). Search for the port numbers to find on this page: http://www.sans.org/y2k/020700-2000.htm
f) A less extensive example was found on 06/09/2000 by Daniel B. Holzman whom sent a block notice concerning an attack to port #12345 (Reference "Holzman" item #2): http://www.sans.org/y2k/060900.htm
g) For information on WhatsUp the manual is available from ipswitch: http://www.ipswitch.com/Support/manuals.html
h) Reference CVE "NetBus" (under review): http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-1999-0660
i) Information on NetBus, its functionality, and detection and removal can be found at: http://www.nttoolbox.com/Netbus.htm
j) A table of associated protocols, ports, and where applicable, trojans:

Number:	Protocol:	Name:
137	TCP/UDP	NetBIOS-Name Service (Not a trojan)
12345	TCP	NetBus, GabanBus
12346	TCP	NetBus, GabanBus

k) While I'm not saying it doesn't exist I've not been able to find an example of the WhatsUp reconnaissance probe followed by the attack on the NetBIOS services at and applicable trojan port probes. Something new?

© SANS Institute 2000 - 2002 As part of GIAC practical repository. Author retains full rights.

Detect #10 Continued:

7) Evidence of active targeting:

   a) Yes! This was directed against all hosts on our subnet.

8) Severity:

   a) Formula: [Criticality + Lethality]
      -[System Countermeasures + Network Countermeasures]
      = Severity

   b)

Category:	Rating:	Comments:
Crit:	5	Web servers are critical to our operation.
Leth:	5	NetBIOS DoS or Trojan compromise.
Sys:	5	Caught and blocked by portsentry!
Net:	3	Upstream router allow is allowing this traffic to this port.
Severity:	3	This is an issue!

9) Defensive recommendation:

   a) Will recommend that the up-stream routed block access to unused ports and especially port 137.

10) Multiple choice test question:

This trace is typical of:
   a) A standard service query followed by load balancing.
   b) Reconnaissance probe followed by an attempt to exploit NetBIOS services.
   c) Established NetBus trojan communications.
   d) An example of ICMP hidden payload communications.

Answer: b

© SANS Institute 2000 - 2002, Author retains full rights.