GIAC Certified Intrusion Analyst (GCIA)

Practical Assignment
Version 4.0

Ken Foster
SANSFIRE, Monterey, CA July, 2004
Submitted: Dec 20, 2004

Abstract

This paper is a report from Curious Minds consulting to Digital University, analyzing a log of network alerts, 2002.6.14. The report is divided into 3 sections:

1. Executive Summary
2. In-depth Analysis
3. Analysis Process

A variety of statistics were selected to demonstrate overall state of the network, and 3 important detects were analyzed in detail:

1. DNS NAMED version attempt
2. Webroot Directory Traversal
3. IIS ISAPI Overflow

Part 1 - Executive Summary

Curious Minds Consulting has completed its review of the IDS log you sent us from July 13-14, 20002. The log contained a moderate amount of malicious traffic into your network, including DNS NAMED version queries, Webserver Directory Traversals, CodeRed attacks, as well as a variety of noisy, less-serious looking scans and probes; however, we found no evidence of compromised machines. The overall security of the network appears to be moderate to good.

We must stress that looking at intrusion detects only provides a limited view of the overall security of Digital University's network. We would prefer to have a picture of all the traffic on the network from either a firewall or network sniffer like tcpdump that will log every packet. These tools would provide us with much needed context around the detects we found. For example, we are very interested in whether the computers that received NAMED version queries were in fact DNS servers and whether they responded to the queries. Therefore, Curious Minds would like to recommend a follow up consult, where the appropriate amount of monitoring and context could be added, including a 2nd Snort sensor inside the firewall so that we could see what traffic was in fact getting through to the protected network. We could provide this for the

Detailed Analysis

Scenario

The log file http://www.incidents.org/logs/Raw/2002.6.14 from Digital University was used as the basis this analysis. Despite the file name the Snort intrusion detection system reports the timeframe of the packets inside the file as ranging from 8:09 PM July 13, 2002 to 7:56 PM July 14, 2002.

Relationship analysis

A quick look at the windump text output of the university log file shows that its home network to be 46.5.0.0/16, since every datagram is either coming from or going to that address range. Several recent practicals (Esler, Perdue, Stodola) have started their analysis by reviewing the MAC (Media Access Control) addresses seen by the University snort detection system that generated the log. Sure enough, all packets in this log leaving the university's network have a source MAC address of 00:00:0c:04:b3:33 (identified by ethereal as Cisco_04:b3:33) and a destination Mac address of 00:03:e3:d9:26:C0 (Cisco_ d9:26:C0). A search of IEEE OUI (Organizationally Unique Identifier) database (IEEE, 2004) verified that both MAC addresses were registered as Cisco Systems, Inc. This suggests that the snort sensor sits between two Cisco devices, which may be border routers, Firewalls, or possibly one of each.

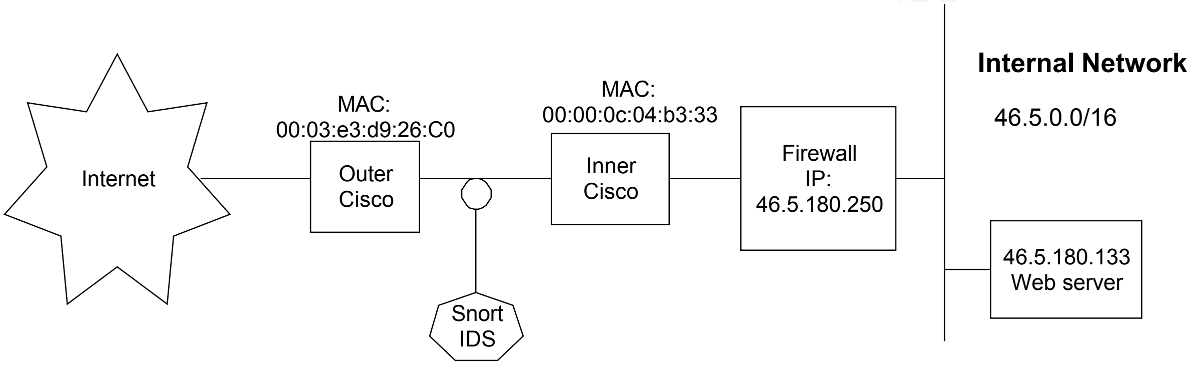
There were 48 outbound packets from the university network: 46 from 46.5.180.250 and 2 from 46.5.180.133. Looking more closely at those leaving 46.5.180.250: 44 went to one address (64.154.80.51), the other two packets went to separate addresses (64.12.184.141 and 64.94.89.210). Pushing a little further into the 44 packets, we see big divergence in the TTL values in these packets that ate supposedly all going between 2 addresses. 19 of these packets had a TTL of 124 while 25 had a TTL of 240. This suggests the address 46.5.180.250 may indeed be a Firewall "hiding" at least two different computers with different Operating Systems talking behind it.

Turning to the other IP address that is sending outbound from the university network, windump (-X) reveals that this is an Apache Web server running on RedHat Linux:

```
13:10:24.674488 46.5.180.133.80 > 195.29.69.205.1317: P 1804976994:1804977530(536) ack
7714629 win 32696 (DF)
0x0000  4500 0240 d145 4000 3f06 8303 2e05 b485 E..@.E@.?.......
0x0010  c31d 45cd 0050 0525 6b95 c362 0075 b745 ..E..P.%k..b.u.E
0x0020  5018 7fb8 5ec8 0000 4854 5450 2f31 2e31 P...^...HTTP/1.1
0x0030  2034 3033 2046 6f72 6269 6464 656e 0d0a .403.Forbidden..
0x0040  4461 7465 3a20 5375 6e2c 2031 3420 4a75 Date:.Sun,.14.Ju
0x0050  6c20 3230 3032 2031 373a 3034 3a35 3120 l.2002.17:04:51.
0x0060  474d 540d 0a53 6572 7665 723a 2041 7061 GMT..Server:.Apa
0x0070  6368 652f 312e 332e 3132 2028 556e 6978 che/1.3.12.(Unix
0x0080  2920 2028 5265 6420 4861 742f 4c69 6e75 )..(Red.Hat/Linu
0x0090  7829 206d 6f64 5f6a 6b20 6d6f 645f 7373 x).mod_jk.mod_ss
```

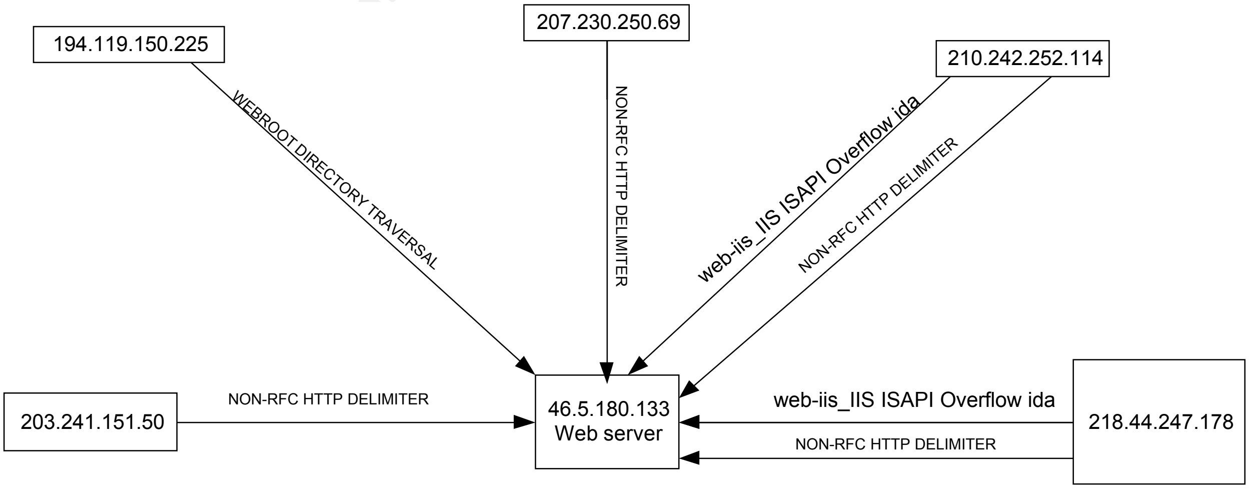
View of net topology

The wide variety of addresses and port/services flowing through the device with the MAC of Cisco_ d9:26:C0, labeled Outer Cisco in the drawing below, suggests that it is probably a border router to the Internet. The 2nd Cisco device, labeled Inner Cisco, may also be a router, or it may be a firewall. We are being conservative in the drawing below by showing the the Cisco as a router with the firewall sitting behind it. The firewall is providing Network Address Translation (NAT) for most of the machines behind it. The Webserver at 46.5.180.133 is an exception; since we have seen inbound and outbound packets with this address it must be set up to pass through without NAT.



Link graph

The Webserver at 46.5.180.133 is attacked on 7 different vectors by 4 sources. Both IIS ISAPI Overflow attacks include non-RFC delimiters. We can also see 2 other instances where the non-RFC Delimiter anomaly is present without the Overflow attack.



Overview of detects

Alerts were produced by Snort version 2.20 with this command line:

```
snort -de -r 2002.6.14 -c "C:\snort\etc\snort.conf" -l log -k none
```

Command	**Description**
-d	dump the Application Layer
-e	display the 2nd layer header info, which contains the MAC address
-r 2002.6.14	read from the file 2002.6.14
-c "C:\snort\etc\snort.conf"	use this rules file
-l log	Log to the log subdirectory
-k none	Ignore checksums (scrubbing data produced bad checksum for each packet)

```
Snort processed 291 packets.
===============================================================================
Breakdown by protocol:
   TCP: 275   (94.502%)
   UDP: 13     (4.467%)
DISCARD: 3    (1.031%)
===============================================================================
Action Stats:
ALERTS: 113 LOGGED: 113
PASSED: 0
===============================================================================
Fragmentation Stats:
Fragmented IP Packets: 37 (12.715%)
   Fragment Trackers: 37
```

Snort ID	Description of Detects	Number
[1:524:8]	BAD-TRAFFIC tcp port 0 traffic	64
[1:1616:6]	DNS named version attempt	13
[119:4:1]	(http_inspect) BARE BYTE UNICODE ENCODING	10
[1:523:5]	BAD-TRAFFIC ip reserved bit set	9
[119:18:1]	(http_inspect) WEBROOT DIRECTORY TRAVERSAL	8
[119:13:1]	(http_inspect) NON-RFC HTTP DELIMITER	4
[116:46:1]	(snort_decoder) WARNING: TCP Data Offset is less than 5!	3
[1:0:0]	IDS552/web-iis_IIS ISAPI Overflow ida	2
	Total Detects	**113**

Detect 1: DNS named version attempt

Description of Detect

This detect shows reconnaissance for the version of BIND that may be running on a DNS server. Some older versions of BIND respond to these queries. The version could be potentially valuable information to an attacker, who could attempt to attack vulnerabilities associated with a given version. For example, the Common Vulnerabilities and Exposures CVE-1999-0009 (MITRE.org, 2004a) discusses Inverse query buffer overflow in BIND 4.9 and BIND 8 Releases.

Reason this detect was selected

These scans, if successful, often lead to direct attacks against core DNS servers that are fundamental to the productivity of the network infrastructure.

Detect was generated by

Snort IDS version 2.2.0 for Windows was used with the default rule-set. The UDP query fired off Snort rule 1:1616, “DNS Named version attempt”:

```
alert udp $EXTERNAL_NET any -> $HOME_NET 53 (msg:"DNS named
version attempt"; content:"|07|version"; offset:12; nocase;
content:"|04|bind"; offset:12; nocase; reference:arachnids,278;
reference:nessus,10028; classtype:attempted-recon; sid:1616;
rev:6;)
```

The above rule states that Snort will send an alert message ("DNS named version attempt") whenever it sees a query from any external address to an internal address using the DNS port 53 that also has the words version and bind in the payload of the packet. Specifically, the rule says skip the first 12 bytes of the UPD packet then look for the words version and bind in either case.

The packet below matched the Snort rule shown above. It was produced by windump version 3.6.2 with winpcap version 2.3 in Hex dump mode (-X), and it shows the text strings version and bind.

```
22:49:41.734488 210.195.43.71.2076 > 46.5.147.58.53:  4660 [b2&3=0x80] TXT CHAOS)?
version.bind. (30)
0x0000 [4500 003a 4997 0000 2f11 87da d2c3 2b47    E..:I.../.....+G
0x0010 2e05 933a] <081c 0035 0026 08b6 1234 0080   ...:...5.&...4..
0x0020 0001 0000 0000 0000 0776 6572 7369 6f6e     .........version
0x0030 0462 696e 6400 0010 0003>                   .bind.....
```

Probability the source address was spoofed

Since the prober needs to see the response of the query, the source address is most likely not spoofed.

Attack mechanism

A DNS NAMED version attempt is not an attack, but rather reconnaissance that often leads to an attack. The Berkeley Internet Name Daemon (BIND), NAMED, is an open source DNS server from the Internet Systems Consortium (isc.org) that is the de facto standard for DNS servers on the Internet. Many proprietary DNS servers are also based on BIND. BIND-based NAMED servers (prior to version 9) will respond to queries of the CHAOS TXT record with their version and type.

There were 13 detects by snort broken down into 2 sets, 7 from one address and 6 from another. The first scan, from address 210.195.43.71, was slow-paced, taking over 5 ½ hours (22:49 -> 4:19).

```
1. 22:49:41.734488 210.195.43.71.2076 > 46.5.147.58.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30)
2.   23:33:11.024488 210.195.43.71.3497 > 46.5.239.181.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
3.   00:33:41.724488 210.195.43.71.2651 > 46.5.148.180.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
4.   00:41:24.594488 210.195.43.71.2709 > 46.5.218.212.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
5.   02:33:28.934488 210.195.43.71.3623 > 46.5.49.65.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
6.   02:38:45.714488 210.195.43.71.1182 > 46.5.183.173.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
7.   04:19:43.284488 210.195.43.71.2214 > 46.5.21.54.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
     (30)
```

The second scan from address 203.197.102.21 was much quicker taking only taking 51 minutes (9:40 -> 10:31).

```
1. 09:40:18.434488 203.197.102.21.1633 > 46.5.171.227.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
   (30)
2. 09:55:27.794488 203.197.102.21.1842 > 46.5.185.1.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30)
3. 10:03:44.254488 203.197.102.21.2700 > 46.5.0.83.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind. (30)
4. 10:04:28.144488 203.197.102.21.3473 > 46.5.180.251.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
   (30)
5. 10:17:49.624488 203.197.102.21.1768 > 46.5.92.200.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
   (30)
6. 10:31:35.304488 203.197.102.21.4468 > 46.5.143.173.53:  4660 [b2&3=0x80] TXT CHAOS)? version.bind.
   (30)
```

Although getting this information could permit an attacker to take dead aim at a core server, the log examined showed no responses to any of these UDP probes, so there is no evidence of compromise; in fact, since there was no other traffic from these computers, so we cannot confirm whether any of them are even DNS servers.

Correlations

Common Vulnerabilities and Exposures CVE-1999-0009 (MITRE.org, 2004a) documents an inverse query buffer overflow for Bind 4.9 and BIND 8 Releases. Arachnids (IDS278) states that the buffer overflow attacks, discovered in 1998, that would allow arbitrary commands to be run on the affected server. Several SANS students, like Marks (2004) and Shakeel (2003), analyze similar NAMED version attempts.

Evidence of active targeting

It's hard to say if this is active targeting of DNS servers or a scan looking for them because we don't have other traffic to verify what address the DNS servers actually have. There are no traces of traffic responding from port 53, the DNS port, and in fact all the traffic in the log that I inspected going to port 53 consisted of these recon probes.

Severity

Criticality	4 - It's recon, not an attack, but since it's looking for DNS servers.
Lethality	3 - This is an older, well-known vulnerability
System Countermeasures	3 - DNS BIND servers should patched or blocked by now.
Network countermeasures	3 - don't have any background information about the systems involved, so I would assume a mixed bag of patching and versions
Severity	1

Detect 2: Webroot Directory Traversal

Description of attack

This detect shows an attempt to gain inappropriate access to files outside the bounds of the website. It targets unpatched servers running either IIS versions 4 or 5. CVE-2000-0884 refers to a flaw in the way URL strings are decoded, whereby a remote user can read files and possibly execute commands.

Reason this detect was selected

These scans, if successful, may allow a remote user to alter the contents of the web site or worse, depending on how the server is configured. The remote user can read or execute and files on the same logical drive as the Web site, and in the case where the site resides on the same drive as the operating system, the user would have access to all the system commands accessible to a locally logged on user. In addition, the user could use these tools to try to elevate privileges or attempt to take advantage of other known vulnerabilities of the system.

McAfee (sv_ent01) reports that from Feb. 12-16, 2001, attackers used this vulnerability to deface web sites of several large companies, including Compaq, AltaVista, HP, CompUSA, Lycos, Intel, and the New York Times.

Detect was generated by

This detect was picked up by pre-processor component of Snort IDS version 2.2.0 for Windows, which examines http payloads for malicious and mal-formed requests.

```
 [**] [119:18:1] (http_inspect) WEBROOT DIRECTORY TRAVERSAL [**]
07/13-23:57:35.474488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x71
194.119.150.225:3963 -> 46.5.180.133:80 TCP TTL:111 TOS:0x0 ID:49921
IpLen:20 DgmLen:99 DF ***AP*** Seq: 0x8AE38823  Ack: 0xBA2E0F35  Win:
0x2238  TcpLen: 20
```

Probability the source address was spoofed

The source address is most likely not spoofed since the attack requires an established TCP session and the attacker to be successful needs to receive the output of the command he or she is trying to run on the remote web server.

Attack mechanism

The fundamental mechanism in this attack occurs with the way IIS handles input strings. According to Rain Forest Puppy, an online security researcher who first reported the vulnerability to Microsoft (Zetter, 2001), the problem occurs because "IIS seems to decode UNICODE at the wrong instance (after path checking, rather than before)" (arachNIDS IDS432). In other words, an attacker can use Unicode characters to disguise the fact that a URL string is really pointing to an inappropriate location outside the bounds of the web site. Default, unpatched installations of IIS version 4 and 5 are vulnerable to this attack.

In this case the attacker is trying to run the Windows "directory" command that will list the contents of the important windows' system directory, "system32." Looking at the hex output below of this packet provided by the windump –X command, we can see the text rendering of the payload on the right. In bold is the attacking command: **GET./scripts/..%5c%5c../winnt/system32/cmd.exe?/c+dir.**

```
23:57:35.474488 194.119.150.225.3963 > 46.5.180.133.80: P
2330167331:2330167390(59) ack 3123580725 win 8760 (DF)
0x0000    4500 0063 c301 4000 6f06 12b6 c277 96e1 E..c..@.o....w..
0x0010    2e05 b485 0f7b 0050 8ae3 8823 ba2e 0f35 .....{.P...#...5
0x0020    5018 2238 0e6e 0000 4745 5420 2f73 6372 P."8.n..GET./scr
0x0030    6970 7473 2f2e 2e25 3563 2535 632e 2e2f ipts/..%5c%5c../
0x0040    7769 6e6e 742f 7379 7374 656d 3332 2f63 winnt/system32/c
0x0050    6d64 2e65 7865 3f2f 632b 6469 720d 0a69 md.exe?/c+dir..i
0x0060    720d 0a                                  r..
```

%5c%5c – a Unicode representation for 2 backslashes.
.. – 2 dots tell the operating system to move up one level the parent of the present directory

By using the combination of characters above the attacker is able to direct the IIS command processor to move out the web folders, basically anywhere on the logical drive.

Assuming an unpatched IIS server that has its website installed on the same logical drive as the operation systems (say, c:\), this command would actually execute the “dir” command inside the windows command shell (cmd.exe), which would display the contents of the important system32 directory and more importantly tell the attacker that she or he has a vulnerable server.

According to Microsoft’s security bulletin MS00-078 (Microsoft, 2000) that includes the patch for this vulnerability, the attacker is limited to the logical drive where IIS is installed. Best practices state that IIS should not be installed on the same drive as the operating system, and in that case this specific command could not work. However, all files on the web site would still be vulnerable, and as seen with the defacements mentioned in the previous section, quite a bit of harm could still be done.

Since we only have access to the one file, we don’t know if these attacks were successful. However, if we had access to either the IIS event logs or a firewall log with this traffic in it, we could quickly check if the attacker received a 200 response indicating success.

Correlations

Rain Forest Puppy, discovered that IIS servers were vulnerable to directory traversal attacks by embedded Unicode characters and reported it to Microsoft in December, 2000 (Zetter, 2001). CVE-2000-0884 was created in January 22, 2001 to recognize a variety of vulnerabilities for IIS servers to Unicode-embedded URLs. SANS Malware FAQ: Windows NT UNICODE Vulnerability Analysis (Marin) contains an excellent discussion of the vulnerability as well as the role and history of Unicode.

Evidence of active targeting

Based on the limited evidence from just one day's log of detects, it definitely looks like these machines probably were actively targeted The attacker was able to establish TCP sessions and submit the same request containing the Directory Traversal to eight different machines in the span 11 seconds (23:57:35 – 23:57:46) as seen in the windump output below.

```
1.  23:57:35.474488 194.119.150.225.3963 > 46.5.180.133.80: P
2.  23:57:35.474488 194.119.150.225.3965 > 46.5.180.135.80: P
3.  23:57:35.474488 194.119.150.225.3964 > 46.5.180.134.80: P
4.  23:57:35.494488 194.119.150.225.3975 > 46.5.180.145.80: P
5.  23:57:39.014488 194.119.150.225.4204 > 46.5.180.151.80: P
6.  23:57:39.024488 194.119.150.225.4206 > 46.5.180.153.80: P
7.  23:57:39.034488 194.119.150.225.4211 > 46.5.180.158.80: P
8.  23:57:46.234488 194.119.150.225.4728 > 46.5.180.250.80: P
```

Severity

Criticality	4 - Intenet-facing webservers usually are important to the business and public image of a University.
Lethality	3 - This attack has been around for a while.
System Countermeasures	2 - Don't have any information about the systems but will assume that they at least installed IIS on a different drive than the Operating System (per best practices).
Network countermeasures	1 - The network has to allow http traffic to the webservers. Best defenses for this attack are at the system level.
Severity	3

Detect 3: ***IIS ISAPI Overflow ida***

Description of Detect

This detect is an attack against vulnerable IIS servers. CVE-2001-0500 discusses a vulnerability for all unpatched Microsoft IIS Web servers version 4 and 5 installed in the default manner, in which a buffer overflow in the IIS Index Server 2.0 (also known as the Indexing Service on Windows 2000) can result in a remote user gaining full control of the server.

Snort detected 2 attempts (shown below) directed at this vulnerability in the University log named 2002.6.14 that is being examined for this report. It should be noted that despite the name of the log that snort reports the traffic in the log covering 7/13 21:09 to 7/14 20:56. Both detects were to the same server (46.5.180.133) from different source addresses.

```
[**] [1:0:0] IDS552/web-iis_IIS ISAPI Overflow ida [**]
[Classification: Web Application Attack] [Priority: 1]
07/14-14:14:50.804488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5EE
210.242.252.114:4013 -> 46.5.180.133:80 TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1504
***AP*** Seq: 0x4277DF51  Ack: 0x7B4480EB  Win: 0x7D78  TcpLen: 20
[Xref =>  arachnids 552]


[**] [1:0:0] IDS552/web-iis_IIS ISAPI Overflow ida [**]
[Classification: Web Application Attack] [Priority: 1]
07/14-15:20:35.534488 0:3:E3:D9:26:C0 -> 0:0:C:4:B2:33 type:0x800 len:0x5C0
218.44.247.178:8324 -> 46.5.180.133:80 TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:1458
***AP*** Seq: 0x3A858958  Ack: 0xAD449B1  Win: 0x7F0A  TcpLen: 20
[Xref =>  arachnids 552]
```

Looking at some of the responses from this server in the log reveals that this server is running Apache website (see bold red text below) and therefore could not be compromised because this an attack against II servers.

```
13:10:24.674488 46.5.180.133.80 > 195.29.69.205.1317: P 1804976994:1804977530(536) ack
7714629 win 32696 (DF)
0x0000  4500 0240 d145 4000 3f06 8303 2e05 b485 E..@.E@.?.......
0x0010  c31d 45cd 0050 0525 6b95 c362 0075 b745 ..E..P.%k..b.u.E
0x0020  5018 7fb8 5ec8 0000 4854 5450 2f31 2e31 P...^...HTTP/1.1
0x0030  2034 3033 2046 6f72 6269 6464 656e 0d0a .403.Forbidden..
0x0040  4461 7465 3a20 5375 6e2c 2031 3420 4a75 Date:.Sun,.14.Ju
0x0050  6c20 3230 3032 2031 373a 3034 3a35 3120 l.2002.17:04:51.
0x0060  474d 540d 0a53 6572 7665 723a 2041 7061 GMT..Server:.Apa
0x0070  6368 652f 312e 332e 3132 2028 556e 6978 che/1.3.12.(Unix
0x0080  2920 2028 5265 6420 4861 742f 4c69 6e75 )..(Red.Hat/Linu
0x0090  7829 206d 6f64 5f6a 6b20 6d6f 645f 7373 x).mod_jk.mod_ss
```

Reason this detect was selected

This detect was chosed because even though it has been around since the summer of 2001, default installations of IIS that aren’t patched are still vulnerable to take over or defacement.

Detect was generated by

Snort IDS version 2.2.0 for Windows generated this detect using the following rule:

```
2. alert TCP $EXTERNAL_NET any -> $HOME_NET 80 (msg: "IDS552/web-iis_IIS ISAPI
   Overflow ida"; dsize: >239; flags: A+; uricontent: ".ida?"; classtype: web-
   application-attack; reference: arachnids,552;)
3.
```

Originally, however, this detect was only picked up by a visual review of the hex packets. That prompted question of why a packet with the signature so obviously like Code Red wasn’t detected by Snort. Here is the “WEB-IIS ISAPI .ida attempt” rule from the web-iis.rules file that I would have expected to catch these packets:

```
4. 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; reference:bugtraq,1065; reference:cve,2000-0071;
   classtype:web-application-attack; sid:1243; rev:11;)
```

I think the explanation lies with the flow option part of the rule: ***flow:to_server,established***. When I removed the flow option and reran the log through snort, it alerted on the two packets mentioned in the description. The snort user manual describes this flow option as “verifying this is traffic going to the server on an established session” (Snort, 2003). I think the problem is that the log being examined doesn’t contain all the related packets including the 3-way TCP handshake that led to this attempted attack; therefore, the rule doesn’t work as intended and these packets become false negatives. For the purpose of this analysis, I added the

One other note, there is a 3rd packet in this log with the Code Red signature. Snort never alerted on it, even when the flow requirement was removed. The packet was the first fragment received (shown below). No other fragments or packets were exchanged between these computers in this log. If this is all the packets for this day then the packet time-out without doing any damage. I would definitely check this computer at 46.5.23.118 to make sure that if is IIS it is patched.

```
5. 21:30:30.064488 80.6.66.193.2437 > 46.5.23.118.80: P 760737404:760738832(1428)
   ack 2140171777 win 17520 (frag 25611:1448@0+)
```

Probability the source address was spoofed

The source address is not likely to be spoofed since the attacker needs to complete the tcp handshake to establish a web session with the target machine.

Attack mechanism

IIS ISAPI (Internet Server API) web attacks target a vulnerability in the IIS Index Server discovered on June 18, 2001 by eEye Digital Security (Hassel, 2001). IDQ.DLL (Internet Data Query), the indexing engine, does not correctly validate user inputs, which leaves it susceptible to buffer overflows. Since IDQ.DLL runs under the SYSTEM context, a successful remote attacker will have complete control of the victim server.

Ironically, the Indexing service does not have to be running to be vulnerable. According to Microsoft Security Bulletin MS01-033 (Microsoft, 2003), “as long as the script mapping for .idq or .ida files were present, and the attacker was able to establish a web session, he could exploit the vulnerability.” .IDA files (Internet Data Administration) refer to administrator scripts that can run as part of the Index Server.

The most well-known attacks against this vulnerability were the several variants of the CodeRed worm that did considerable destruction during the summer of 2001. Through the buffer overflow the worm injects a virus into the memory of the victim, and randomly targets other hosts by trying to establish TCP sessions on port 80. If successful, the virus sends an HTTP GET request that will exploit the overflow on a vulnerable web site.

The signature of a CodeRed worm is quite distinctive. The payload begins with Get /default.ida? followed by a long string of Ns (also A or X have been used), which will overflow the idq.dll buffer, as shown in bold red below.

```
21:30:30.064488 80.6.66.193.2437 > 46.5.23.118.80: P
760737404:760738832(1428) ack 2140171777 win 17520 (frag 25611:1448@0+)
0x0000       4500 05bc 640b 6000 6c06 b3f5 5006 42c1 E...d.`.l...P.B.
0x0010       2e05 1776 0985 0050 2d57 ee7c 7f90 6e01 ...v...P-W.|..n.
0x0020       5018 4470 e686 0000 4745 5420 2f64 6566 P.Dp....GET./def
0x0030       6175 6c74 2e69 6461 3f4e 4e4e 4e4e 4e4e ault.ida?NNNNNNN
0x0040       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0050       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0060       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0070       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0080       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0090       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x00a0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x00b0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x00c0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x00d0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x00e0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
```

```
0x00f0       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0100       4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e 4e4e NNNNNNNNNNNNNNNN
0x0110       4e4e 4e4e 4e4e 4e4e 4e25 7539 3039 3025 NNNNNNNNN%u9090%
0x0120       7536 3835 3825 7563 6264 3325 7537 3830 u6858%ucbd3%u780
0x0130       3125 7539 3039 3025 7536 3835 3825 7563 1%u9090%u6858%uc
0x0140       6264 3325 7537 3830 3125 7539 3039 3025 bd3%u7801%u9090%
0x0150       7536 3835 3825 7563 6264 3325 7537 3830 u6858%ucbd3%u780
0x0160       3125 7539 3039 3025 7539 3039 3025 7538 1%u9090%u9090%u8
0x0170       3139 3025 7530 3063 3325 7530 3030 3325 190%u00c3%u0003%
0x0180       7538 6230 3025 7535 3331 6225 7535 3366 u8b00%u531b%u53f
0x0190       6625 7530 3037 3825 7530 3030 3025 7530 f%u0078%u0000%u0
0x01a0       303d 6120 2048 5454 502f 312e 300d 0a43 0=a..HTTP/1.0..C
```

Correlations

CVE-2001-0500 (CVE Version: 20040901) documents this overflow vulnerability as does Arachnids: 552. Thram (2001), McBee (2003), Yackley (2003), and Zehner (GCIH) all presented excellent history and analysis of IIS vulnerability and the Code Red exploits including the history of the different variants.

Evidence of active targeting

At first look, there appears to be active targeting at work here, since there were three CodeRed-type packets in the log examined, and all of them found a machine listening on port 80. However, this log doesn't seem to represent all the traffic for this day, since the packets establishing the TCP handshakes for these attacks are not here. That leaves open the possibility that there were other attempts that went to machines not listening on port 80 that aren't in this log. Therefore, knowing that CodeRed spreads by randomized targeting and the fact that we don't have all the traffic, I would lean against active targeting, but we would need more evidence to settle the issue.

Severity

Criticality	4 - Intenet-facing webservers usually are important to the business and public image of a University.
Lethality	1 – Web servers attacked are Apache on Linux. No problem.
System Countermeasures	4 – Linux doesn't have a problem.
Network countermeasures	2 - The network has to allow http traffic to the web servers, however, site is running snort they should be alerted to the danger and should review any IIS servers they have for patches.
Severity	-1

Network Statistics

Top Five Talkers

This list is based on the 5 addresses sending the most packets.

Top Talkers	No.	%
46.5.180.250	46	14.2%
255.255.255.255	44	13.5%
194.52.177.9	18	5.5%
202.29.28.1	17	5.2%

Top Five Targeted Ports

Dst Port		No.	%
80	http	175	53.8%
0	No service	64	19.7%
515	Printer?	44	13.5%
53	DNS	13	4.0%
21	FTP	9	2.8%

Suspicious External Source Addresses

211.47.255.20-23

These 4 addresses from Korea sent out 64 Syn packets to Port 0 of 4 different addresses on the University network. These addresses are managed under the Korean National Registry (KRNIC). Currently, none of them are allocated, so whomever was using them 2 years ago has when this log was generated has moved on. I would still be wary of any addresses in the 211.46.0.0 - 211.49.255.255 assigned to KRNIC unless there is a business reason for it.

According to the snort write up of this rule, traffic to port 0 is often used in reconnaissance to determine if a host is alive, but in any case this traffic is never valid.

172.20.10.199

This address is part of a block is reserved by the Internet Assigned Number Authority (IANA) for private internets as proscribed by RFC 1918 (1996); therefore these addresses should not be seen on the internet. This address sent 3 packets to port 80 to 3 different university addresses. All 3 packets had the reset flag set, so they may be a response to stimulus using a spoofed address.

```
172.20.10.199     4940      >     46.5.105.47               80     R       2350747827:2350747827(0) win 5840
<mss 1460 nop nop sackOK nop wscale 0> (DF)
172.20.10.199     2234      >     46.5.237.245              80     R       1526740252:1526741850(1598) ack
2768227195 win 34752 [tos 0x10]
172.20.10.199     1800      >     46.5.51.186               80     R       0:3(3) ack 0 win 0
```

192.1.1.188

This address triggered 9 alerts for Snort rule 1:523, “BAD-TRAFFIC ip reserved bit set.” According to ARIN (American Registry for Internet Numbers), this address belongs to Bolt Beranek and Newman Inc., the company awarded the original contract to build ARAPNET, the forerunner of today’s internet. Most likely the address is spoofed, unless BBN has a network device misconfigured as described below or has a rogue computer on its net. All 9 packets were fragments going to 9 different university addresses with no source or destination ports.

Snort rule 523 provides this background:

> Under normal circumstances IP packets do not use the reserved bit. This may be an indicator of the use of the reserved bit by a malicious user to instigate covert channel communications. [It can be] an indicator of unauthorized network use, reconnaissance activity or system compromise. These rules may also generate an event due to improperly configured network devices.

Other Correlations

Correlations from practicals are referenced throughout the paper. It’s difficult to correlate 2 year-old detects with current activity. The Top 20 targeted ports downloaded from http://isc.sans.org/port_report.php, December 20, 2004, shows that port 53 (4th) and port 80 (13th) are still being actively targeted.

Target Port	Reports	Sources	Targets
445	553011	36503	95021
135	209924	7463	109868
139	31650	4325	10501
53	20491	3599	558
1026	13679	3506	7998
1025	24074	3311	6244

1434	128666	2992	85772
1027	10671	2844	7414
113	8467	2305	247
137	64243	1599	46091
25	11215	1437	335
3127	31574	1275	17529
80	8378	1045	2120

Insights into internal machines

I saw no evidence of compromised machines.

Defensive Recommendations

All servers running BIND should be checked for version and exposure to the NAMED version vulnerability. Nessus (ID 100028) has a script for its scanner to run against BIND servers that will report if they are vulnerable, and recommends using the 'version' directive in the 'options' section will block the 'version.bind' query. Servers could also be upgraded to a newer version of BIND without this vulnerability.

Defense in depth is the key to protecting business assets. A stateful firewall that restricts traffic inbound and outbound to only what is necessary for the business is a must. Network and host-based intrusion detection systems should be used to monitor to how well your protections are working, as well as to provide indicators of previously unknown anomalous behavior and forensic capability in case of attack.

Hosts should follow best-practice installation and configuration, disabling all services that aren't needed, like FTP, Telnet, and web servers for servers that aren't using it (Windows 2000 installs IIS by default) and installing IIS on a different drive than the operating system is installed on. All servers, particularly those that are internet-facing, must get on timely patch management program to keep with the swift current of vulnerabilities and exposures that is accelerating all the time. IIS servers should have URLscan and IISLockdown run on them.

Be sure to have an up-to-date inventory of all IIS servers. Verify that they have all the critical IIS and operating system patches installed. Microsoft Security Bulletin MS01-033 (Microsoft, 2003) provides a patch for the ISAPI buffer overflow problem in particular. In addition, URLscan and IISLockdown should be run on all IIS servers.

Analysis Process

Sorting the Packets

I wanted to import windump output into Excel so that it could be easily sorted by time, source, source port, destination, or destination port. Tcpdump tends to separate data fields by spaces, but not the source or destination ports, which are just appended to the ip address like a 5th division. Also, rest of the packet after the flags field has so many spaces between either pieces or strings of data that if each space was treated a delimiter, you could have 20-30 fields tacked on, making it very hard to see and interpret that data.

Therefore, I decided I could normalize the packet data with a fairly straightforward PERL script. The source and destination ports were decoded and separated from the rest of the ip address by a tab so that each would now be in its own field. Then I substituted tabs for the spaces between the first 7 fields only. Other spaces were left in so that the rest of the packet would wind up in the 8th field without any distortion. One final adjustment I decided on was to replace the colons in the time field with periods to eliminate the whole issue with Excel misinterpreting the time.

Importing the adjusted file into Excel using tabs as the delimiter, created an easily sortable table like this:

Time	Source	Src Port	>	Destination	Dest Port	Flags	Rest of WinDump packet
04.04.37.984488	194.230.125.224	3617	>	46.5.69.147	3128	S	942067828:942067828(0) win 5840 <mss 1460 nop nop sackOK nop wscale 0> (DF)
13.41.14.434488	46.5.180.133	80	>	213.191.149.3	3187	P	0:3(3) ack 0 win 0
04.03.23.234488	194.230.125.224	3552	>	46.5.69.147	8080	S	873808934:873808934(0) win 5840 <mss 1460 nop nop sackOK nop wscale 0> (DF)
04.03.26.374488	194.230.125.224	3552	>	46.5.69.147	8080	S	873808934:873808934(0) win 5840 <mss 1460 nop nop sackOK nop wscale 0> (DF)

Processing the Alerts

I wrote a PERL script to read through the snort alert.ids file, putting each type of alert in hash and keeping a running account as it went.

```
    if(/\[\*\*/) {        #process only lines with **; these are the titles
            chomp();    # remove new line or carriage return
            s/\[\*\*\]//g; # remove braces and asterisks
            s/^\s//;       # remove beginning space
            $record_count++;

            # put in hash if unique
            $rule_count = undef;
            $rule_count = $snortrules{$_};   # get the current count for this type
            #print "rule count $rule_count after fetch\n";

            # check if type exists; if not, add it; else increment existing count
            unless($rule_count) {
                    $rule_count = 1;
                    $snortrules{$_} = $rule_count;
            } else {
                    $rule_count = $rule_count + 1;
                    $snortrules{$_} = $rule_count;
            }
    }
```

The script generated this table of summary data:

64	[1:524:8]	BAD-TRAFFIC tcp port 0 traffic
44	[1:184:6]	BACKDOOR Q access
13	[1:1616:6]	DNS named version attempt
10	[119:4:1]	(http_inspect) BARE BYTE UNICODE ENCODING
9	[1:523:5]	BAD-TRAFFIC ip reserved bit set
8	[119:18:1]	(http_inspect) WEBROOT DIRECTORY TRAVERSAL
4	[119:13:1]	(http_inspect) NON-RFC HTTP DELIMITER
3	[116:46:1]	(snort_decoder) WARNING: TCP Data Offset is less than 5!
2	[1:1243:11]	WEB-IIS ISAPI .ida attempt
157	total detects	

Reference List

Akhter, Shakeel (2003, March 11). GCIA Version 3.3 Practical Detects. Retrieved December 1, 2004 from http://www.dshield.org/pipermail/intrusions/2003-March/007097.php

Arachnids (IDS278). *NAMED-PROBE-VERSION*. Retrieved November 22, 2004 from http://www.whitehats.com/info/IDS278

Arachnids (IDS552). *IIS ISAPI OVERFLOW IDA*. Retrieved November 22, 2004 from http://www.whitehats.com/info/IDS552

Bailey, Brian (2004, October 17). SANS GIAC GCIA Practical. Retrieved December 15, 2004 from http://www.giac.org/practical/GCIA/Brian_Bailey_GCIA.pdf.

BUGTRAQ (2001, June 18). *Remote buffer overflow* (SYSTEM Level Access). Retrieved November 8, 2004 from http://www.securityfocus.com/bid/1065

Esler, Joel (2004, September). SANS GIAC GCIA Practical. Retrieved December 13, 2004 from http://www.giac.org/practical/GCIA/Joel_Esler_GCIA.pdf.

Evans, Andrew (2003). SANS GIAC GCIA Practical version 3.3. Retrieved December 13, 2004 from http://www.giac.org/practical/GCIA/Andrew_Evans_GCIA.pdf.

Hassell, Riley and Permeh, Ryan (2001, June 18). eEye Digital Security, *eEye Advisory 20010618*. Retrieved October 16, 2004 from http://www.eeye.com/html/Research/Advisories/AD20010618.html

IEEE Registration Authority (2004, December 16). IEEE OUI and Company_id Assignments. Retrieved November 16, 2004 from http://standards.ieee.org/regauth/oui/oui.txt

Perdue, Robert (2004, September 29). SANS GIAC GCIA Practical. Retrieved December 12, 2004 from http://www.giac.org/practical/GCIA/Rob_Perdue_GCIA.pdf.

Marin, Marvin. SANS Malware FAQ: Windows NT UNICODE Vulnerability Analysis. Retrieved November 27, 2004 from www.sans.org/resources/malwarefaq/wnt-unicode.php

Marks, Ian (2004, July 25). GIAC GCIA Version 3.5 Practical Detect. Retrieved from http://lists.sans.org/pipermail/intrusions/2004-July/008200.html

McAfee (sv_ent01). Unicode Directory Traversal Vulnerability in Microsoft IIS. Retrieved November 27, 2004 from http://www.networkassociates.com/us/security/resources/sv_ent01.htm

McBee, Rob (July 8, 2003).GIAC GCIA Practical (version 3.3). Retrieved November 29, 2004 from http://www.giac.org/practical/GCIA/Rob_McBee_GCIA.pdf.

Microsoft (2003, November 04 -updated) Microsoft Security Bulletin MS01-033, *Unchecked Buffer in Index Server ISAPI Extension Could Enable Web Server Compromise*. Retrieved November 22, 2004 from http://www.microsoft.com/technet/security/bulletin/MS01-033.mspx

Microsoft (2000, October 17). Microsoft Security Bulletin MS00-078, Patch Available for 'Web Server Folder Traversal' Vulnerability. Retrieved November 27, 2004 from http://www.microsoft.com/technet/security/bulletin/MS00-078.mspx

Mitre.org (2004, September 1). Common Vulnerabilities and Exposures, *CVE-2001-0500.* Retrieved November 12, 2004 from http://cve.mitre.org/cgi-bin/cvename.cgi?name=2001-0500

Mitre.org (2004a, September 1). Common Vulnerabilities and Exposures, *CVE-1999-0009.* Retrieved November 23, 2004 from http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0009

Mitre.org (2004b, September 1). Common Vulnerabilities and Exposures, *CVE-2000-0884*. Retrieved November 28, 2004 from http://cve.mitre.org/cgi-bin/cvename.cgi?name=2000-0884

Nessus (ID 100028). Nessus Plug-ins, *Determine which version of BIND name daemon is running*. Retrieved October 17, 2004 from http://cgi.nessus.org/plugins/dump.php3?id=10028

RFC 1918, February 1996. RFC 1918 *Address Allocation for Private Internets*, p. 3. Retrieved December 16 2004 from http://www.isi.edu/in-notes/rfc1918.txt

SANS.org. CodeRed II: Incident Handling Process and Procedures. Retrieved November 28, 2004 from http://www.sans.org/rr/whitepapers/incident/639.php

Snort.org (2003). Snort Users Manual 2.2.0, *section 3.8.3 catch the oddities of the protocol in the rule*. Retrieved September 8, 2004 from http://www.snort.org/docs/snort_manual/

Snort.org (SID 116:46). snort_decoder: TCP Data Offset is less than 5!. Retrieved November 8, 2004 from http://www.snort.org/snort-db/sid.html?sid=116%3A46

Snort.org (SID 119:18). http_inspect: WEBROOT DIRECTORY TRAVERSAL. Retrieved November 28, 2004 from http://www.snort.org/snort-db/sid.html?sid=1243

Snort.org (SID 523). BAD-TRAFFIC tcp port 0 traffic. Retrieved November 5, 2004 from http://www.snort.org/snort-db/sid.html?sid=523

Snort.org (SID 524). BAD-TRAFFIC tcp port 0 traffic. Retrieved November 5, 2004 from http://www.snort.org/snort-db/sid.html?sid=524

Snort.org (SID 1243). *WEB-IIS ISAPI .ida attempt*. Retrieved October 12, 2004 from http://www.snort.org/snort-db/sid.html?sid=1243

Snort.org (SID 1616). *DNS named version attempt*. Retrieved October 12, 2004 from http://www.snort.org/snort-db/sid.html?sid=1616

Stodola, Jan (2004, September 23). SANS GIAC GCIA Practical. Retrieved December 9, 2004 from http://www.giac.org/practical/GCIA/Jan_Stodola_GCIA.pdf.

Thram, Adrian (2001, August 4). What is Code Red Worm? Retrieved November 28, 2004 from http://www.sans.org/rr/whitepapers/malicious/45.php

Yackley, Matt (2003, January 15). GSEC Practical, *Worms don't care if you're not a bank*. Retrieved November 19, 2004 from http://www.giac.org/practical/GSEC/Matt_Yackley_GSEC.pdf.

Zehner, Roger T. (GCIH Practical). Security Breach; Code Red. Retrieved November 30, 2004 from www.giac.org/practical/Roger_Zehner_GCIH.doc

Zetter, Kim (2001, September 28). PCWorld.com, *Three Minutes with Rain Forest Puppy*. Retrieved November 27, 2004 from http://www.pcworld.com/news/article/0,aid,63944,00.asp