Global Information Assurance Certification Paper

SANS Security Essentials
GSEC Practical Assignment
Version 1.2f

Title : Understanding IIS Vulnerabilities – Fix them!

Table of Contents Page

1.0 Introducing the Internet and Intranet Environment

The explosive growth of the Internet has had some unexpected consequences. One of the major consequences is a realization that the Internet paradigm and particularly the World Wide Web (WWW) paradigm provide a methodology of providing improved access to data. This paradigm works not only on the Internet but also for intranets. Now it describes the employment of Internet technology for enterprise-wide networks and the use of World Wide Web servers and browsers to collect and deliver data to enterprise functions next door and around the world. Intranets are being integrated with the Internet in many cases.

At the base of all of this development is the server. The server is the delivery vehicle for all of the information to be published on the Internet. Microsoft Internet Information Server/Service (MS IIS) is Microsoft's foundation product for the Internet. It demonstrates Microsoft's dedication to the principle of making software straightforward and usable.

Internet Information Server/Service is quickly becoming a *de facto* standard in the burgeoning Internet server market. It provides an easy way to create an Internet or intranet site. It installs and runs all services on an existing Windows NT/2000 Server in just minutes.

2.0 Understanding IIS Security

Microsoft Internet Information Server/Service (IIS) is integrated with the Microsoft Windows NT/2000 Server operating system to provide a Web server for organizations.

Integrated Security? The security architecture of Windows NT/2000 Server is used across all system components, with authentication tied to controlled access to all system resources. IIS integrates into the Windows NT/2000 security model and operating system services such as the file system and directory. Because IIS uses the Windows NT/2000 Server user database, administrators do not need to create separate user accounts on every Web server, and intranet users need only to log on to their network once. IIS automatically uses the same file and group permissions as the existing file, print, and application servers.

Some Web servers install their own security implementations on top of the operating system, creating additional overhead and potential

security exposure due to lack of integration and synchronization. Windows NT/2000 Server is secure by design. Files and system objects can only be accessed with the proper permissions. User and group accounts are managed by a globally unique identification. When accounts are deleted, all access permissions and group memberships are deleted. So even if a new account is created using a previous user name, none of the permissions are inherited.

Manageability? Permissions to control access files and directories can be set graphically, because IIS uses the same Windows NT Server Access Control Lists (ACLs) as all other Windows services, such as file sharing or Microsoft SQL Server™ permissions. Permissions for the Web server are not separate from other file services, so the same files can be securely accessed over other protocols, such as FTP, CIFS/SMB, or NFS without duplicating administration.

Briefly, IIS provides frontline for your Web site, including Authentication and Web permissions.

There are several ways to start or enhance the IIS security. Let's start with the "checklist" where this can help you more efficiently in getting proper security.

As recommended by SANs Institute in it's Windows NT Security Step-by-Step guidelines, version 3.03 February 2001, if you use Internet Information Server (IIS), block known vulnerabilities as follows :

No	Actions
1	Do not install IIS on a domain controller
2	Place the Web Server in the DMZ and use the external router to control the Internet traffic
3	Do not install a printer on the IIS machine
4	Install the web folders on a drive other than the system drive
5	Remove IIS sample pages
6	Remove the virtual directory \IISAMPWD
7	Move, rename, or delete any command-line utilities
8	Apply the very latest Service Packs and hot fixes
9	Disable unnecessary services and features
10	Disable .htr mapping if it is not needed

11	Remove the MS Data Access Components functionality unless specifically needed
12	Secure the anonymous IIS account (*IUSR_computername*)
13	Ensure that the *IUSR_computername* account does not have write access to any files on the system
14	Disable parent Paths
15	Take advantage of IP address restrictions
16	Use either Challenge/Response authentication or Basic authentication with SSL encryption
17	Do not assign both the Write and Script/Execute permissions to the same folder
18	Use the Script permission for Active Server Pages and CGI Scripts
19	Use NTFS on all IIS hard drives
20	Enable W3C Extended logging to keep detailed records of client-server interaction
21	Disable directory browsing, especially on folders containing scripts or executable
22	Unless absolutely required, uninstall the HTML version of the Internet Service Manager utility
23	When using FTP, only allow anonymous access
24	Avoid allowing FTP upload or write privileges
25	Set a relatively short connection time-out period, and a limited number of simultenous sessions on FTP servers
26	Consider using Virtual Private Networking technologies along with FTP when FTP is necessary
27	If at all possible, do not install the MS Frontpage Server Extensions and do not allow users to manage their personal web sites with Frontpage
28	Do not install MS Index Server if it will not be used

29	If you wrote your own web applications, it is crucial to always perform proper bounds checking and validation of input data

Review these checklist web sites to determine if aspects of your security could be improved.

http://www.microsoft.com/technet/itsolutions/security/tools/iischk.asp

http://www.ciac.org/ciacNT/iis/CheckList.htm

http://www.microsoft.com/technet/itsolutions/security/tools/iis5chk.asp

http://www.sans.org/infosecFAQ/audit/audit_list.htm

http://www.microsoft.com/windows2000/en/server/iis/htm/core/iisckl.htm

Each day more people are "called" due to security issues on internet therefore more reports or writings are published in the internet free of charge.

For more securing IIS information go to :

http://www.sans.org/infosecFAQ/win2000/sec_IIS.htm

http://www.sans.org/infosecFAQ/win2000/win2000_sec.htm

http://www.sans.org/infosecFAQ/encryption/SSL_IIS.htm

3.0 Understanding Web Server

3.1 Understanding HyperText Transport Protocol (HTTP)

The HyperText Transport Protocol (HTTP) is an application-level protocol used by Web clients and Web servers to communicate with each other. HTTP has been in use since 1990.

HTTP is a generic and stateless protocol. It is lightweight and fast. Messages are in a format similar to that used by Internet Mail and the Multipurpose Internet Mail Extensions (MIME).

HTTP is a request/response protocol. A Web client establishes a connection with a Web server and sends a resource request. The

request contains a request method, protocol version, followed by a MIME-like message. The message contains request modifiers, client information, and possible body content.

The Web server responds with a status line, including the message's protocol version and a success or error code. It is followed by a MIME-like message containing server information, entity meta-information, and possible body content.

You can find out details of HTTP in the following Request for Comments (RFC).

HTTP 1.0 specifications are described in RFC 1945:
(http://www.cis.ohio-state.edu/htbin/rfc/rfc1945.html)

MIME specifications are described in RFC 1521:
(http://www.cis.ohio-state.edu/htbin/rfc/rfc1521.html)

3.2 Understanding HyperText Markup Language (HTML)

The HyperText Markup Language is a document-layout, hyperlink-specification, and markup language. Web clients use it to generate resource requests for Web servers, and to process output returned by the Web server for presentation. A markup language describes what text means and what it is supposed to look like.

A fundamental property of HTML is that the text it describes can be rendered on most devices. A single HTML Web page on a Web server can be displayed on a PC, Mac, UNIX, and so on.

HTML 3.2 specifications are available online at:
(http://www.w3c.org/)

3.3 Understanding the Universal Resource Locator (URL)

A Uniform Resource Locator (URL) is an abstract identification that locates a resource on a Web server.

A URL contains the following information:

- **Protocol**
  Specifies the Internet protocol to access a resource. The

abstract encompasses FTP, Gopher, and HTTP Internet protocols.

- **Network Endpoint**
  Internet address of Internet Information Server and protocol port number

- **Resource Location**
  Path information to locate a resource on Internet Information Server

URL syntax is

{service}::///{host}[:port]/[path/.../] [file name]

Required parameters are surrounded by {}. Optional parameters are surrounded by []. Other characters are mandatory separators.

- Service is a required field. Web servers support FTP, Gopher, and HTTP services.
- Host is a required field. This field is the host name or IP address of the Internet Information Server.
- Port is an optional field. This field is an abstraction used by the network and transport layers to select a service on the server. This field is not frequently used. It may be specified if the service is available on a nonstandard protocol port number.
- Path is an optional field. It specifies URL resource location. A path without a file name following must always end with a / character.

The combination of host and port is a network endpoint.

An example of a URL is

http://www.infomax.com:8080/welcome.htm

The http: component is the service. The http://www.infomax.com:8080 component is the network endpoint. The /welcome.htm component is the resource location.

Uniform Resource Locators are described in RFC 1738 and RFC 1808.
(http://www.cis.ohio-state.edu/htbin/rfc/rfc1738.html)
(http://www.cis.ohio-state.edu/htbin/rfc/rfc1808.html)

3.4 Understanding Web Server Services

Web servers offer the following services:

- **File Transfer Protocol (FTP) Service**

  The Web server File Transfer Protocol (FTP) service can transfer any type of file between the Web server and an FTP client.

  The Web server FTP service handles concurrent access by multiple FTP clients. Each FTP client establishes a socket connection to the Web FTP service, and logs onto it. Web browsers hide the login process from the user.

  FTP clients use a limited set of commands, and have restricted file access. The socket connection to the Internet Information Server FTP Service lasts until the FTP client disconnects.

  FTP is one of the earliest Internet TCP/IP protocols. Web browsers and other graphical interface applications have replaced early FTP client applications.

  Most FTP services do not provide descriptions of files. Browsing through directories is a slow process.

- **Gopher Service**

  The Internet Gopher is a tool for browsing through files and directories over the Internet. A Gopher client establishes a socket connection to a Web server Gopher service. Login is usually not required for a Gopher client.

  A Gopher client displays a hierarchy of items and directories much like a file system, in a menu of text-labeled choices. It may be a list of files, subdirectories, or a combination of both. A Gopher client copies a selected file over the network and

displays it.

The Gopher menu can point to files and directories on other Gopher servers on the Internet. It was the first Internet service to offer such a feature.

The Internet Gopher has limited graphical presentation abilities. It cannot present graphics and text together.

The Internet Gopher and HTTP are similar network protocols. They became available at about the same time. Most new Internet sites do not offer Gopher services.

Many older Internet sites have stopped offering it. They have converted Gopher documents to HTML documents because HTML can present graphics and text together. HTML documents rely on HTTP protocol.

- **World Wide Web (WWW) Service**

  Web browsers processing HTML documents use Internet HTTP protocol to transact with World Wide Web (WWW) Service.

  Web Server WWW Service knows how to respond to an HTML request by analyzing URL fields.

  - **Static HTML Page**
    WWW Service looks for a file name in the path field. An example is:
    http://www.infomax.com/welcome.htm
  - **CGI Application**
    WWW Service looks for a file name in the path field with a file extension that has been associated with an application. An example is:
    http://www.infomax.com/cgiapps/gcidoit.pl

The HyperText Transmission Protocol is a stateless protocol designed to process a single transaction during a connection to a server. It is layered on the TCP and IP protocols.

There are four steps during a single HTTP transaction:

- **Connection**
  The HTTP client establishes a socket connection to the Web WWW service.

- **Request**
  The HTTP client sends a request to the Web WWW service. The request contains the type of HTTP service request and other information.

- **Response**
  The Web WWW service sends a response back to the HTTP client. The response contains the state of the transaction and the data requested.

- **Disconnection**
  The Web WWW service signals the end of the transaction by closing the socket connection.

A Web browser makes a connection to a Web server WWW service for each file that is a part of the HTML document. It makes a connection for the HTML text file first. The WWW server disconnects after sending it.

The Web browser parses the returned text file looking for graphical image file names. It then makes a connection to the Web server and requests a single graphical image file. The Web server disconnects after sending it. This process repeats for each graphical file.

HTTP is the most used Internet protocol. It accounts for about 25 percent of Internet packets. FTP Internet protocol is second. It accounts for about 15 percent of Internet packets.

3.5 Understanding Dynamic Web Pages

The popularity of the static, billboard style, Web page is declining. Web server content developers are creating dynamic Web pages with data from databases and other data sources, such as real-time stock market data feeds. Web technology is being used to create new client/server applications because of the ability to dynamically create Web pages on the fly.

Many Web servers do not have database or other data access mechanisms built-in. They rely on the Common Gateway Interface (CGI).

- **Understanding CGI**

  The Common Gateway Interface (CGI) is a standard way of interfacing external applications with Web servers.

  A CGI external application executes in real-time and dynamically produces output information. It processes HTTP requests from Web clients and returns an HTML document. GCI external applications usually access information not in HTML form. They act as a gateway between the Web client and the information.

  Further information about CGI can be found at: (http://hoohoo.ncsa.uiuc.edu/cgi/overview.html)

4.0 Understanding Attack Techniques

The different attack techniques used to break into a Web server can be categorized into three groups: Web server attacks, Web application attacks, and Indirect Attacks.

4.1 Web Server Attacks

These techniques send HTTP requests to the Web server. The firewall captures this traffic and, typically, concentrates on analyzing the communication parameters of the traffic. It checks the destination port, the source and destination IP addresses, and similar other attributes. However, a firewall's weakness lies in its inability to verify the data portion (e.g., requests) of the communication consistently. This allows the request to appear legitimate to the firewall. When it arrives at the Web server, it is serviced normally. However, the request may be malicious and exploit a server vulnerability, producing undesired results.

Between 1998 and 2000, about 50 new attacks that exploit Microsoft's widely utilized Internet Information Server (IIS) were created and published. Of those attacks, 55% allowed an intruder to read sensitive information such as Active Server Pages (ASP) source files, configuration information, and files on

the same drive but outside of the file tree dedicated to the Web server (virtual tree).

Approximately 20% of the attacks target the ASP component in IIS. ASP is a server-side scripting technology that can be used to create dynamic and interactive Web applications. The ASP source files often include valuable information such as database file names, schema description and passwords that are not supposed to be exposed. A well-known example for an ASP related vulnerability is the "MS Index Server '%20' ASP Source Disclosure Vulnerability" (Bugtraq #1084). It is exploited by the browser, sending the following URL:
http://target/null.htw?CiWebHitsFile=/default.asp%20&CiRestriction=none&CiHiliteType=Full

As a result, the source of the file specified in the 'CiWebHitsFile' field is sent back to the browser.

Another well-known vulnerability is the '+.HTR' vulnerability of the IIS Web server. Requesting a filename with an appendage of "+" and ".HTR" will force IIS to call ISM.DLL to open the target file. If the target file is not an .HTR file, part of the target file's source code will be revealed. Again, the exploit is very simple: send the following URL using your browser and view the source code of the returned page:
http://www.victim.com/global.asa+.htr

The "global.asa" file is a primary target for hackers, since it is used to specify event scripts and declare objects that have session or application scope. It is not a content file displayed to the users; instead, it stores event information and objects used globally by the application. This file has to be named "global.asa" and has to be stored in the root directory of the application. As a result, the hackers can easily locate it and use any one of the above exploits to obtain its content. The file typically contains several functions including "Application_OnStart" which is activated when a new session starts. In many cases, the code connects to the database and makes the necessary initialization. In the following excerpt from a real world "global.asa" file, the connection string provides the database name (DB), the user name (DBADMIN) and the password (supersecretpswrd).

Sub Application_OnStart

```
'==Visual InterDev Generated - startspan==
'--Project Data Connection
Application("FmLib_ConnectionString") =
"DSN=DB;UID=DBADMIN;PWD= supersecretpswrd"
Application("FmLib_ConnectionTimeout") = 15
Application("FmLib_CommandTimeout") = 30
Application("FmLib_CursorLocation") = 3
Application("FmLib_RuntimeUserName") = "sa"
Application("FmLib_RuntimePassword") = ""
'-- Project Data Environment
Set DE = Server.CreateObject("DERuntime.DERuntime")
Application("DE") = DE.Load(Server.MapPath("Global.ASA"),
"_private/DataEnvironment/DataEnvironment.asa")
'==Visual InterDev Generated - endspan==
ReadApplicationSettings
End Sub
```

Once the hackers obtain this information, they will look for other vulnerabilities such as MDAC RDS (described later) that will allow them to log into the database and obtain confidential information.

One of the major goals of hackers is to run their own code on the server. If hackers are able to run their code with privileged access rights, they can, for example, add a new user with Administrator rights and actually control the machine. Approximately 15% of the attacks allow an intruder to execute code on the server. For example, "IIS Hack" is a buffer overflow vulnerability exposed by the way IIS handles requests with .HTR extensions. A hacker sends a long URL that ends with ".HTR". IIS interprets it as a file type of HTR and invokes the ISM.DLL to handle the request. Since ISM.DLL is vulnerable to a buffer overflow, a carefully crafted string can be executed in the security context of IIS, which is privileged. For example, it is relatively simple to include in the exploit code a sequence of commands that will open a TCP/IP connection, download an executable and then execute it. This way, any malicious code can be executed.

A growing number of attacks target the databases behind the Web server. By exploiting vulnerabilities in the IIS server, it is possible to run SQL commands gaining access to the database,

or even obtaining administrative privileges. An example in this category is the MDAC RDS vulnerability. MDAC is a package used to integrate Web and database services. It includes the RDS component that provides remote access to database objects through IIS. By exploiting vulnerabilities in RDS (provided that several conditions in the target Web site are met), attackers can send arbitrary SQL commands that manipulate the database or retrieve any desired information. In this specific case, the attacker can even gain administrative rights by embedding the shell () VBA command into the SQL command and execute any highly privileged system commands.

4.2 Web Application Attacks

Web applications have become ubiquitous and are used by most Web sites to generate dynamic Web pages based on inputs and databases. Most Web servers provide an interface used to spawn and communicate with the Web application. The interface links between an HTTP request and an application. It specifies which application should be invoked, the parameters/data passed to the application and the mechanism used to provide the Web server with the dynamically generated page. One such interface, the Common Gateway Interface (CGI), is widely supported.

In many cases, CGI programs are distributed as part of the Web server distribution disks and installed by default. According to a bulletin entitled "How To Eliminate the Ten Most Critical Internet Security Threats" published by the SANS Institute, many CGI programmers fail to consider ways in which their programs may be misused or subverted to execute malicious commands. The report illustrates how vulnerable CGI programs present a particularly attractive target to intruders because they are relatively easy to locate, and they operate with the privileges and power of the Web server software itself.

One of many recent examples is the vulnerability found in CGI Script Center's Account Manager PRO script. According to the SecurityFocus Web site (www.securityfocus.com), any remote user can modify the administrative password of the Account Manager program. The hacker simply sends an appropriate POST command and, as a result, is granted full administrative

privileges. This will allow the hacker to access secured areas of the Web site.

Another source that creates vulnerabilities for Web applications are the designers of homegrown and 3rd party Web applications. Typically, these applications are subject to short development cycles, poor testing, and minimal quality assurance procedures. Additionally, they usually lack sufficient security knowledge.

A common problem with Web applications is input validation. An example is given in the following:

> An HTML form has an input field named "e-mail address" where the user is supposed to fill in his email address. A hacker could enter the following string "jsmith.home.com; mail hacker@hackeremail-address </etc/passwd". If the Web application implementing this form does not check the input but rather spawns a shell that executes the input string, the /etc/passwd file – the password file on Unix systems – is sent to the hacker by email.

4.3 Indirect Attacks

There are many alternative routes other than port 80 (HTTP) for breaking into the Web server machine. An intruder will definitely begin his hacking attempts by scanning the TCP/IP ports looking for Internet servers listening on open ports.

For example, the IIS Web server package includes an FTP server that listens on port 21. Some IIS 4.0 FTP servers that have installed a specific post-SP5 FTP hotfix are vulnerable to an exploit whereby FTP clients may download and/or delete files (on the FTP server). Downloading files from the machine is definitely problematic. The hacker might download confidential data or gain additional information that can further allow him to break into the machine and gain administrative privileges.

Another typically open port is the DNS port. The DNS server is used for Internet name resolution, providing domain name to IP address translation that facilitates the routing on the Internet. At a minimum, a hacker can break into the DNS server, manipulate the routing table so e-mail sent to a specific interesting domain

will be diverted to his machine, allowing him to read all the incoming mail.

When the hacker only wants to crash or slow down the server, he can apply several low-level network attacks that target the OS networking software. For example, a recently published attack effective for Windows and some Cisco routers forces CPU utilization of 100% on the target, slowing down the machine considerably. This is done by sending identical fragmented IP packets to the target at the rate of approximately 150 packets per second.

5.0 Understanding IIS Vulnerabilities

5.1 Web Server Survey

From the Netcraft Web Server survey of Web Server software usage on Internet connected computers, Microsoft has been a second player in the Totals for Top Active Servers Across All Domains (Figure 1)

Microsoft continues its recent gains, with a further half a percent rise, due in part to the remainder of a large domain hosting system at Network Solutions completing a migration to Windows 2000, and in part because it has far less exposure to the mass hosting companies than Apache. Our data was collected at the start of the month, and we will have a clearer picture of whether Code Red has caused any significant movement away from Microsoft-IIS in September.

Developer	July 2001	Percent	August 2001	Percent	Change
Apache	7314577	60.53	7156849	60.33	-0.20
Microsoft	**3372341**	**27.91**	**3356363**	**28.29**	**0.38**
iPlanet	282517	2.34	275619	2.32	-0.02
Zeus	184895	1.53	181098	1.53	0.00

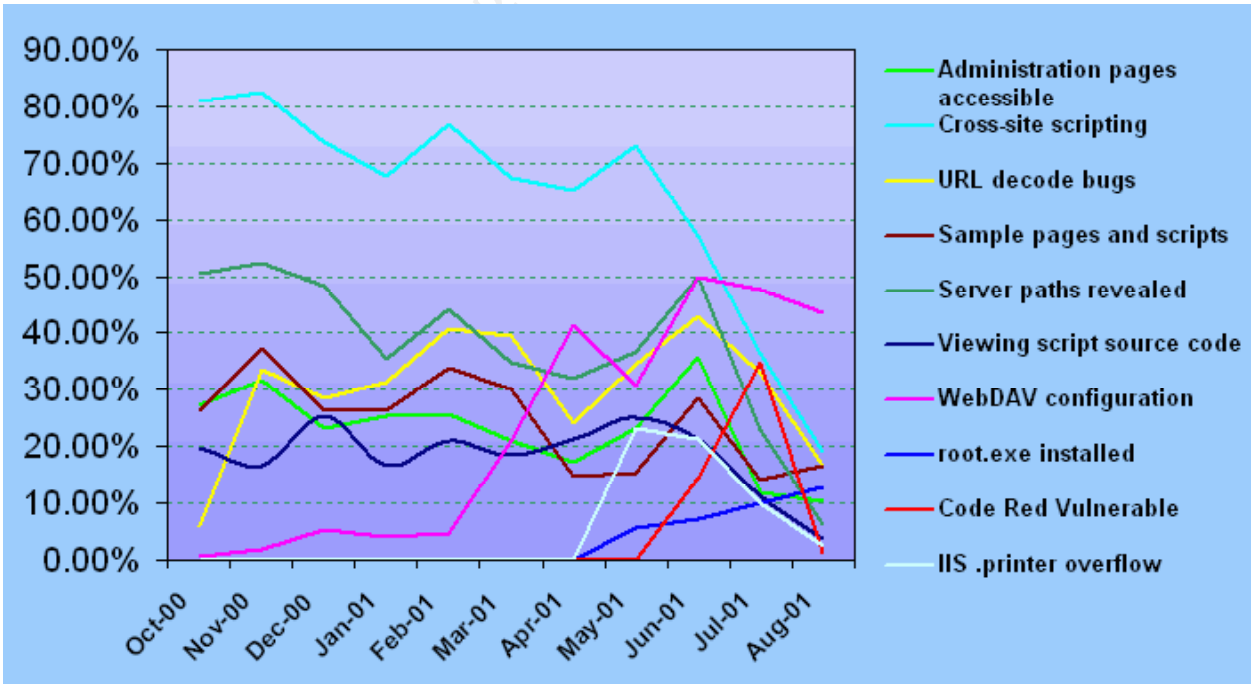
Figure 1 : Top Active Servers Across All domain

However, the combination of the Code Red worm and the first cumulative patch for Microsoft-IIS has significantly improved the

security of Microsoft-IIS systems on the internet. Figures are shown below are for the vulnerability of Microsoft-IIS sites tested for the first time by Netcraft security services over the last year. This is typically in the range of a few hundred systems in each month.

	Percentage %										
	Oct-00	Nov-00	Dec-00	Jan-01	Feb-01	Mar-01	Apr-01	May-01	Jun-01	Jul-01	Aug-01
Administration pages accessible	27.38	31.61	23.16	25.49	25.58	20.93	17.33	23.08	35.71	11.76	10.26
Cross-site scripting	80.95	82.58	73.68	67.65	76.74	67.44	65.33	73.08	57.14	36.47	19.23
URL decode bugs	5.95	33.55	28.42	31.37	40.70	39.53	24.00	34.62	42.86	32.94	16.67
Sample pages and scripts	26.19	37.42	26.32	26.47	33.72	30.23	14.67	15.38	28.57	14.12	16.67
Server paths revealed	50.60	52.26	48.42	35.29	44.19	34.88	32.00	36.54	50.00	22.94	6.41
Viewing script source code	19.64	16.77	25.26	16.67	20.93	18.60	21.33	25.00	21.43	11.18	3.85
WebDAV configuration	0.60	1.94	5.26	3.92	4.65	20.93	41.33	30.77	50.00	47.65	43.59
IIS .printer overflow	0.00	0.00	0.00	0.00	0.00	0.00	0.00	23.08	21.43	10.00	2.56
Code Red Vulnerable	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.29	34.71	2.00
Root.exe installed	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.77	7.14	10.00	12.82

Figure 2 : % of Vulnerable Microsoft-IIS SSL Sites



The table (in Figure 2) demonstrates in part the deep set complacency regarding security amongst ecommerce sites, and in part the difficulties in maintaining a reasonable level of security without the benefit of regular external testing. The high visibility of Code Red induced many ecommerce sites running Microsoft-IIS to patch their systems for the first time, and the availability of a cumulative patch has eliminated a lot of earlier vulnerabilities from many sites.

Note that the patch does not necessarily remove the root.exe facility installed by both sadmind/IIS and Code Red II. root.exe allows anyone on the internet to have commands on the machine executed with web server privileges, and can typically be used to set up logging of credit card information and other sensitive data on SSL servers. This has created a new class of ecommerce site which has been correctly patched for known server vulnerabilities, but have a live backdoor facility enabling attackers to continue to remain in control of the machine. Currently around 12% of SSL sites running Microsoft-IIS tested for the first time are in this state.

5.2 Fix the IIS Vulnerabilities

The Internet is now the world's most popular network and it is full of potential vulnerabilities. Let’s explore the vulnerabilities of the Internet and what you can do to mitigate them.

The following table is a summary of the supported web sites that can assist you to fix the IIS Vulnerabilities.

IIS Vulnerability	Supported Web Sites

URL Redirection DoS	http://www.microsoft.com/technet/security/bulletin/MS01-044.asp http://www.microsoft.com/technet/security/bulletin/MS01-031.asp http://www.securityfocus.com/vdb/ http://www.cert.org/incident-notes/IN-2001-10.html http://www.microsoft.com/technet/treeview/default.asp?url=/technet/itsolutions/security/topics/codealrt.asp
SSI Buffer Overrun Privelege Elevation	http://www.microsoft.com/technet/security/bulletin/MS01-044.asp http://www.securityfocus.com/templates/archive.pike?list=1&msg=200108170112.f7H1CfZ01880@www.nsfocus.com http://www.securityfocus.com/vdb/ http://www.securiteam.com/windowsntfocus/5JP0B2055G.html
MIME Header Denial of Service Vulnerability	http://www.microsoft.com/technet/security/bulletin/MS01-044.asp http://www.microsoft.com/technet/treeview/default.asp? url=/TechNet/prodtechnol/iis/tips/iis5chk.asp http://www.securityfocus.com/vdb/
MS Index Server and Indexing Service ISAPI Extension Buffer Overflow	http://www.microsoft.com/technet/security/bulletin/MS01-033.asp http://www.securityfocus.com/vdb/ http://www.eeye.com/html/Research/Advisories/AD20010618.html http://www.cert.org/advisories/CA-2001-23.html http://www.securiteam.com/windowsntfocus/5WP0L004US.html http://xforce.iss.net/static/6705.php
WebDAV Invalid Request Denial of Service	http://www.microsoft.com/technet/security/bulletin/MS01-044.asp http://www.microsoft.com/technet/support/kb.asp?ID=241520 http://www.securityfocus.com/vdb/

Unicode .asp Source Code Disclosure	http://www.securityfocus.com/vdb/ http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/iischk.asp http://www.vigilante.com/inetsecurity/advisories/VIGILANTE-2001001.htm http://xforce.iss.net/static/6742.php
IIS/PWS Escaped Characters Decoding Command Execution	http://www.microsoft.com/technet/security/bulletin/MS01-026.asp http://www.securityfocus.com/templates/advisory.html?id=3309 http://www.securityfocus.com/vdb/ http://xforce.iss.net/static/6534.php
Microsoft Index Server Buffer Overflow	http://www.microsoft.com/technet/security/bulletin/MS01-025.asp http://www.atstake.com http://www.gap.com http://www.securityfocus.com/vdb/
IIS 5.0 .printer ISAPI Extension Buffer Overflow	http://www.microsoft.com/technet/security/bulletin/MS01-023.asp http://www.securityfocus.com/vdb/ http://www.eeye.com/html/Research/Advisories/AD20010501.html http://xforce.iss.net/static/6485.php
WebDAV 'Search' Denial of Service	http://www.microsoft.com/technet/security/bulletin/MS01-016.asp http://www.securityfocus.com/vdb/ http://www.microsoft.com/technet/support/kb.asp?ID=241520
Multiple Invalid URL Request DoS	http://www.microsoft.com/technet/security/bulletin/MS01-014.asp http://www.securityfocus.com/vdb/ http://www.esecurityonline.com/vulnerabilities.asp http://xforce.iss.net/static/6171.php

File Fragment Disclosure	http://www.microsoft.com/technet/security/bulletin/ms01-004.asp http://www.microsoft.com/technet/security/bulletin/fq01-004.asp http://www.securityfocus.com/vdb/ http://xforce.iss.net/static/5903.php
Front Page Server Extension DoS	http://www.microsoft.com/technet/security/bulletin/ms00-100.asp http://www.securityfocus.com/vdb/ http://www.securityfocus.com/templates/advisory.html?id=2993 http://xforce.iss.net/static/5823.php
Web Server File Request Parsing	http://www.microsoft.com/technet/security/bulletin/MS00-086.asp http://www.securityfocus.com/templates/advisory.html?id=2914
Session ID Cookie Marking	http://www.microsoft.com/technet/security/bulletin/MS00-080.asp http://www.securityfocus.com/templates/advisory.html?id=2766
Web Server Folder Traversal	http://www.microsoft.com/technet/security/bulletin/MS00-078.asp http://www.securityfocus.com/templates/advisory.html?id=2777 http://www.f-secure.com/v-descs/codeblue.shtml
Invalid URL	http://www.microsoft.com/technet/security/bulletin/MS00-063.asp http://www.securityfocus.com/templates/advisory.html?id=2587
IIS Cross-Site Scripting	http://www.microsoft.com/technet/security/bulletin/MS00-060.asp http://www.securityfocus.com/templates/advisory.html?id=2077

File Permission Canonicalization	http://www.microsoft.com/technet/security/bulletin/MS00-057.asp http://www.securityfocus.com/vdb/

For more information on IIS vulnerabilities, refer to the SANs Institute - Security Reading Room, on the following web sites :

http://www.sans.org/infosecFAQ/threats/SADMIND.htm
http://www.sans.org/infosecFAQ/threats/web_spoof.htm
http://www.sans.org/infosecFAQ/threats/CGI_basics.htm
http://www.sans.org/infosecFAQ/threats/semantic.htm
http://www.sans.org/infosecFAQ/threats/traversal.htm
http://www.sans.org/infosecFAQ/win/life_cycle.htm
http://www.sans.org/infosecFAQ/win/MDAC.htm
http://www.sans.org/infosecFAQ/win2000/vulnerabilities.htm
http://www.sans.org/infosecFAQ/win/IIS_vulnerabilities.htm

6.0 Introducing IIS Security tool – SecureIIS

Worried about the next Microsoft IIS vulnerability? Want to go to sleep at night and not have to worry about your Web site being defaced?

SecureIIS™ The application firewall - protects Microsoft IIS (Internet Information Services) Web servers from known and unknown attacks. SecureIIS wraps around IIS and works within it, verifying and analyzing incoming and outgoing Web server data for any possible security breaches. SecureIIS combines the best features of Intrusion Detection Systems and conventional Network Firewalls all into one.

Named as one of "Three Great Security Tools" by Windows 2000 Magazine, SecureIIS has created quite a stir in the market as it raises the bar for proactive security tools.

SecureIIS protects against the following **types of attacks**:

Buffer Overflow Attacks	Buffer overflow vulnerabilities stem from problems in string handling. Whenever a computer program tries copying a string or buffer into a buffer that is smaller than itself, an overflow is sometimes caused. If the destination buffer is overflowed sufficiently it will overwrite various crucial system data. In most situations an attacker can leverage this to takeover a specific program's process, thereby acquiring the privileges that process or program has. SecureIIS limits the size of the "strings" being copied. Doing this greatly reduces the chance of a successful buffer overflow.
Parser Evasion Attacks	Insecure string parsing can allow attackers to remotely execute commands on the machine running the Web server. If the CGI script or Web server feature does not check for various characters in a string, an attacker can append commands to a normal value and have the commands executed on the vulnerable server.
Directory Traversal Attacks	In certain situations, various characters and symbols can be used to break out of the Web server's root directory and access files on the rest of the file system. By checking for these characters and only allowing certain directories to be accessed, directory traversal attacks are prevented. In addition, SecureIIS only allows clients to access certain directories on the server. Even if a new hacking technique arises, breaking out of webroot will still be impossible.
General Exploitation	Buffer overflows, format bugs, parser problems, and various other attacks will contain similar data. Exploits that execute a command shell will almost always have the string "cmd.exe" in the exploiting data. By checking for common attacker "payloads" involved with these exploits, we can prevent an attacker from gaining unauthorized access to your Web server and its data.

SecureIIS also has the following **features:**

HTTPS/SSL Protection	SecureIIS resides inside the Web server, thus capturing HTTPS sessions before and after SSL (Secure Socket Layer) encryption. Unlike any Intrusion Detection System or firewall currently on the market, SecureIIS has the ability to stop attacks on both encrypted and unencrypted sessions.
High Bit Shellcode Protection	Shellcode is what is sent to a system to effectively exploit a hole called a "buffer overflow". High Bit Shellcode Protection offers you a high degree of protection against this type of attack because it will drop and log all requests containing characters that contain high bits. All normal Web traffic, in English, should not contain these types of characters and almost all "shellcode" requires them to produce the effective exploit.
Third Party Application Protection	The power of SecureIIS is not limited to IIS specific vulnerabilities. SecureIIS can also protect third party applications and custom scripts from attack. If your company has developed customized components for your Web site, components that might be vulnerable to attack, you can use SecureIIS to protect those components from both known and unknown vulnerabilities. Let SecureIIS work as your own web based "Security Quality Assurance" system.

Logging of Failed Requests	In the installed SecureIIS directory, we post a file called SecureIIS.log. This file contains a log of all attacks and what triggered the event that caused SecureIIS to drop the connection. This is an effective way to monitor why requests are being stopped, and who is requesting things that they shouldn't. Since SecureIIS enforces a strong security policy for how sites are configured, you can use this log to find places where your Web site may not be acting correctly due to an insecure setting. Also, since Internet Information Server has the unfortunate habit of not logging attacks like buffer overflows that are successful, a twofold security benefit is provided here. Such attacks are not only stopped, but also logged so you can take action accordingly.
Additional Checks	Additional checks are in place for attacks that do not follow recognized patterns, such as the common ones listed above. This approach provides extra security and protects against various attacks that involve data conversion problems. Limitations are also placed on the size of Uniform Resource Locators (URL/URI), HTTP variables, Request methods, Request Header Size, and other HTTP related content.

All of these additional protection features make SecureIIS the product of today that protects you from the attacks of tomorrow, making it the ultimate proactive security tool.

Benefits :

SecureIIS protects Microsoft IIS (Internet Information Services) Web servers from known and unknown attacks. SecureIIS looks for classes of attacks such as buffer overflows, format string attacks, file path attacks and does not look for specific attack signatures. Most security products rely on vulnerability databases and signatures to detect attacks. This leaves the server susceptible to new undocumented vulnerabilities. By looking for classes of attack, SecureIIS is able to

provide protection from known as well unknown vulnerabilities. With vulnerabilities being discovered on a daily basis, IT Admins are not in a position to keep their servers continuously patched and updated. This is where SecureIIS becomes a powerful insurance policy against unknown attacks.

The power for SecureIIS to stop known and unknown attacks is provided by its use of CHAM (Common Hacking Attack Methods) technology. An eEye innovation, CHAM gives SecureIIS the capability to understand Web server protocol and also various classes of attacks that Web servers are vulnerable to. SecureIIS protects against various classes of attacks, and has the ability to give your Web server up-to-the-minute security that is unmatched by any other product in the market.

SecureIIS wraps around IIS and works within it, verifying and analyzing incoming and outgoing Web server data for any possible security breaches. By working as a module loaded into IIS, SecureIIS does not degrade the performance of the Web Server and does not add overhead.

Refer to the latest version of SecureIIS product at http://www.eeye.com/html/Products/SecureIIS/index.html

List of References

1. Using Microsoft Internet Information Server – Special Edition, http://docs.rinet.ru:8083/MIIS/
2. Understanding Internet Information Security, http://www.microsoft.com/ntserver/techresources/webserv/iissecure.asp
3. The SANS Institute, "Networking and Internet Security Settings", Windows NT Security Step by Step, Version 3.03, February 2001: page 38
4. The Future of Web Server Security, Author: Yona Hollander, PhD, Entercept Security Technologies, http://www.entercept.com
5. The SANS Institute, Information Security Reading Room,

http://www.sans.org/

6. Netcraft Web Server Survey, http://www.netcraft.com/survey/
7. Microsoft Security Bulletin, http://www.microsoft.com/technet/itsolutions/security/current.asp
8. Securityfocus.com – Vulnerability Database, http://www.securityfocus.com/vdb/
9. SecuriTeam.com – Windows NT focus, http://www.securiteam.com/windowsntfocus/
10. Internet Security Systems, http://xforce.iss.net/
11. "SecureIIS" product, http://www.whitehatinc.com/nttools/secureiis/
12. eEye Digital Security, SecureIIS Product, http://www.eeye.com/html/Products/SecureIIS/index.html