Global Information Assurance Certification Paper

Securing Internet Information Server 6.0

Rick Siple
September 29, 2004
Version 1.4c

Abstract

Over the last few years Microsoft has dedicated increasing time and effort to improving the security of its products. Microsoft Internet Information Server is no exception. It has, in fact, been the focus of many improvements and rightly so. Internet Information Server is the gateway via which the public accesses the services and information that a company provides and as such is the focal point of most attacks. This paper will briefly review the improvements Microsoft has made and then make further recommendations for securing both Internet Information Server and the underlying operating system.

Secure By Design[1]

"Secure by design means that the processes and tools used to design and build Microsoft's products must be capable of producing more secure software."[2] In 2002 Microsoft stopped all development so that the staff could be given security training. The design for IIIS 6 was then reviewed and modified to make the underlying product more secure. Briefly, Microsoft introduced the kernel mode driver called HTTP.SYS that handles all HTTP communication. With all the communication code concentrated in one place it was easier to secure. The new kernel mode architecture increased performance as well. One of the main benefits of the new HTTP.SYS driver was to improve the application isolation model. Applications can now be compartmentalized into their own process without significant performance loss. HTTP.SYS also improves logging and application management capabilities with the ability to queue requests, restart failed applications, restart applications on a periodic basis and more.

Secure By Default[3]

"Secure by default means that every Windows component should default to the most secure possible settings."[4] Under Windows 2000 Internet Information Server was installed by default with almost every option enabled. This led to many problems due simply to the fact that the administrator did not know IIS was installed. Microsoft has corrected this error. IIS is not installed by default and when initially installed will serve only static HTML pages. Any ups or extras

---

[1] Microsoft, "Security Enhancements," p. 2.

[2] Microsoft, "Security Enhancements," p. 2.

[3] Microsoft, "Security Enhancements," p. 2.

[4] Microsoft, "Security Enhancements," p. 2.

(ASP, ASP.NET, server side includes, etc.) must be specifically installed and/or enabled by the administrator.

Securing Windows

Before IIS can be secured the underlying installation of the operating system must be secured. There are several steps that can be taken to "harden" the operating system against attack.

Unnecessary Services

A web server, especially one connected directly to the Internet, should ideally be a stand-alone, single purpose server. This reduces the "surface area" available for attack. A service that is disabled or uninstalled can neither be attacked nor present a back-door or stepping stone to elevated privileges or access if the web server is compromised. "In most cases the following default Windows services are not needed on a Web server: Alerter, Browser, Messenger, Netlogon … Simple TCP/IP Services and Spooler."[5]

Disable SMB/NETBIOS

NetBIOS and the Server Message Block protocols are the foundations for file-sharing in Windows. In addition NetBIOS can be used to query the server for configuration information that could be used to architect an attack. Disable SMB and NetBIOS on the Internet facing network interface or disable them completely. Use the Device Manager utility to disable NetBIOS over TCP/IP and the Network control panel to unbind SMB from, at least, the internet facing network interface. See also the section on Port Filtering.

Disable Null Sessions/Anonymous Logons

Null Sessions or Anonymous Logons are unauthenticated connections used by Windows to enumerate, among other things, user and computer names and other resources on a computer or domain. This information is very useful in planning and executing an attack on a network. All web servers should have the registry value HLKM\System\CurrentControlSet\Control\LSA\RestrictAnonymous set to 1 or 2. The possible values are:

0 None. Rely on default values

1 Do not allow enumeration of SAM accounts or names

2 No access without explicit anonymous permission[6]

Note, though, that setting this "could cause undesired behavior because many Windows 2000 services, as well as third-party programs, rely on anonymous

[5] Microsoft, "Securing You Web Server," p. 15.

[6] Microsoft, Knowledge base article 246261, p. 2

access capabilities to perform legitimate tasks."[7] For a stand-alone web server this should not be a problem.

TCP/IP Port Filtering

Even if other services are disabled it may be wise to filter access to the web server. In addition to the firewall, which should only allow access to ports 80 (HTTP) and 443 (HTTPS) IPSec Security Policy can be used to restrict access on the host itself. For those unfamiliar, "Internet Protocol security (IPSec) is a protocol that provides encryption, integrity and authentication services for IP-based network traffic…IPSec is completely transparent to applications because…services are implemented at the transport level." While we are not interested in authentication or other services, IPSec can be configured to block all traffic, or vice versa, allow only traffic destined for desired ports. IPSec policies are configured using the Local Security Policy Microsoft Management Console snap-in.

Harden the TCP Stack

There are several modes of attack that are directed at the transport layer itself rather than at the application layer, the web server. Possibilities include SYN flooding, and using ICMP and or SNMP to probe the network topology or manipulate host routing tables.

The most visible and damaging attack is the SYN flood. Basically, a SYN flood attack is a Denial-of-Service attack that disables a web server by filling the TCP half open connection queue. Once the half-open connection queue is full, the server must wait until a connection is fully opened or until a half open connection times out before accepting any new connections. Setting the value HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Paramters\SynAttackProtect to 1 causes the TCP stack to become more aggressive timing out half-open TCP connections when certain resource availability thresholds are exceeded.

There are at least two other obscure settings worth mentioning. The EnableICMPRedirect setting, when set to 0, will disable ICMP host routing. An attacker can use the ICMP redirect feature to instruct a workstation or server to route traffic through another host, one to which the attacker may have access. There is also the EnableDeadGWDetect setting. If this option is set to 1 an attacker may be able to use the IP dead gateway detection feature to force the server to switch to another gateway.

While any Internet connected computer is vulnerable to a SYN flood attack, the other attacks are probably best dealt with by blocking all other protocols and ports at the firewall and host using IPSec.

---

[7] Microsoft, Knowledge base article 246261, p. 2

Disable the Guest Account

The guest account allows unauthenticated users to log into a server or workstation. While the guest account cannot be deleted, it can be disabled. Use the Computer Management tool to disable the Guest account.

Rename the Administrator Account

The Administrator account is a well-known high-value account against which an attacker can launch an attack. The attacker can be denied this avenue of attack by renaming the Administrator account. The account can be renamed using the Computer Management tool. The account will maintain its elevated privileges due to the fact that it is the GUID for the account which uniquely identifies it, not the name. This is the reason for the many warnings that deleting an account cannot be undone. Creating another account by the same name will result in a different GUID.

Also be sure the Administrator account, by whatever name, is assigned a strong password.

NTFS Permissions

While Internet Information Server will enforce its own permissions on files and other data it is a good idea to properly setup NTFS permissions as well:

> First Grant FULL CONTROL to the Administrator account to the root (\), the remove access rights for the Everyone group from the following directories:
>
> - Root (\)
> - System directory (\WINNT\System32)
> - Framework tools directory (\WINNT\Microsoft.NET\Framework\Tools\{version})
> - Web site root directory and all content directories (the default is \Inetpub*)[8]

Securing Internet Information Server

Microsoft has greatly improved the installation for Internet Information Server 6. Previous version of IIS installed almost every available feature for maximum functionality. This led to configurations with large attack "surface area" with unused services that, never the less, may have contained exploitable security holes. Microsoft has rethought this strategy and for IIS 6, by default, only the minimum necessary services are selected for installation. When installed in the default configuration only the HTTP service is installed and it will only serve static HTML pages. No active content is enabled.

[8] Microsoft, "Securing Your Web Server," p. 19.

IISLockdown

"You can largely automate the process of securing your Web server by running the IISLockdown tool. It enables you to pick a specific type of server role and then improve security for that server with customized templates that either disable or secure various features."[9]

IISLockdown will perform tasks such as removing sample code, disabling (or enabling based on the profile) ISAPI extensions, mapping active content extensions to ISAPI modules and setting NTFS permissions.

Many of the things IISLockdown can or will do will be discussed in some detail later in the paper.

Banners

Microsoft IIS will return some information about the server with every page request. One piece of information is the "Content-Location" header. The content location header will, by default, return the internal IP address of the server. While not useful in and of itself, this information could be used to assist in the deduction of the IP structure used on the internal network. IIS should be configured to return the Fully Qualified Domain Name instead.

Unfortunately, according to Microsoft Knowledge base article 218180, "[c]urrently, there is no supported method of masking the IP address [for IIS6]. Microsoft knows about this issue and is investigating it." Two steps forward, a small step back.

URLScan/URL Parsing

Microsoft created an ISAPI filter called URLScan for IIS 5.0. The purpose of the filter was to scan requested URLs for suspicious or restricted characters or strings and sanitize the URL before further processing. Microsoft has included most of the URLScan functionality in IIS 6, though URLScan still works with IIS 6.

URLScan is installed by IISLockdown and configured via the URLScan.ini in %SystemRoot%\inetsrv\urlscan. Some of the important options are:

UseAllowVerbs	"1" – the AllowVerbs section of the ini is used to determine what verbs the server will accept. "0" – the "DenyVerbs" section will determines what verbs the server will reject. Recommended: "1"
UseAllowExtensions	Similar to UseAllowVerbs but applies to extensions.

[9] Microsoft, "Using IISLockdown," p. 1

	Recommended: “1”
NormalizeUrlBeforeScan	Canonicalizes the URL before processing. This removes any encoded characters that may be used to bypass or confuse URL parsing.
VerifyNormalization	Canonicalizes the result of the first canonicalization to be sure they match.
RemoveServerHeader	“0” – Send server information (type and version) to clients. “1” – Suppress server information. Recommended: “1”
DenyUrlSequences	A list of characters to be rejected in the URL.

There are more options for logging and responding to rejected URL requests. See Bhalla, Belani and Microsoft

Listed below are a set of registry values, found under HKLM\System\CurrentControlSet\Services\HTTP\Parameters) that control the URL parsing behavior built into IIS6: [10]

AllowRestrictedChars	This key accepts a Boolean value, which if non-zero allows HTTP.sys to accept hex-encoded characters in the request URL. The default value for this key is 0. This is also the recommended value as it facilitates the task of input validation at the server-level. If set to 1, potentially malicious characters may be hex-encoded by the attacker in an attempt to bypass input validation routines.
MaxFieldLength	This key allows the administrator to set an upper limit (in bytes) for each header. Its default value is 16KB.
MaxRequestBytes	This key establishes the upper limit on the total size of the request line and the headers. Its default value is also 16KB.
UrlSegmentMaxCount	This key determines the maximum number of URL path segments accepted by the server. It effectively limits the number of slashes that can be included by the user in a request URL.It is recommended that one set fairly stringent limits on this value based on the

[10] Belani, Muckin, p. 3.

	depth of the web document root tree to protect the server from a file system traversal attack. The default value for this key is 255.
UrlSegmentMaxLength	This key sets an upper bound on the maximum number of characters in any URL path segment. This value can also be customized in accordance with the normal operation of the hosted applications to prevent the acceptance of unusually long segments that may cause the application to behave in an anomalistic manner. The default value for this key is 260.
EnableNonUTF8	The value of this key controls the character set that is permitted by HTTP.sys. The default value of 1 permits HTTP.sys to accept ANSI- and DBCS-encoded URLs in addition to those encoded in the UTF8 format.

Location of the WebRoot

The web site files should be stored on a non-system drive or partition. In addition to keeping the system partition clean it can prevent URL canonicalization issues. Canonicalization is when a directory path is simplified or reduced to its lowest complexity. A path may legitimately include one or more ".." (parent) directory references, for instance to obtain a common image such as the web site logo. The web server will manipulate the path string to remove the parent references and leave the direct path to the requested resource. The web server may perform this canonicalization incorrectly causing it return an otherwise inaccessible resource.

Closely related to this is the directory traversal attack wherein the attacker uses parent path references to reach a resource outside the web directory.

If the website is on a partition different from the system canonicalization failures or directory traversal attacks cannot be used to reach the system utilities or data. The system cannot traverse across drives.

Location of the Log Files

In addition to putting the web site data on a non-system partition the web server log files should also be on a non-system partition different from the web site data itself. In addition to potentially spreading the data accesses across multiple physical devices (if the partitions are not on the same physical drive) this prevents an attacker from being able to reach the log files using a directory traversal attack.

Further secure the log files by settings NTFS permission so that only Administrator and System have full control and Backup Operators have Read access.

IUSR/IWAM_ComputerName Accounts

The IUSR_ComputerName and IWAM_ComputerName accounts are anonymous accounts created by IIS. Whenever an unauthenticated user requests a resource the default account is the IUSR_ComputerName account. IWAM_ComputerName is the default account under which to start web applications. Both of the these accounts are well known and should be renamed and IIS reconfigured to the use the renamed accounts.

Use the Computer Management tool to rename the accounts. Edit c:\windows\system32\inetsrv\metabase.xml and change the AnonymousUserName settings to the renamed IUSR account. Change the WAMUserName to the renamed IWAM account. See also later the section on application pools where each application can be assigned a different user.

NTFS Permissions

The Metabase.xml file is the configuration file for Internet Information Server 6. It was a binary file called Metabase.bin under IIS 5. The NTFS permission on this file should be set so that the only access granted is to Administrators and Local System which have Full Control.

IISLockdown creates to groups called Web Anonymous Users and Web Application. The default IUSR account is placed in Web Anonymous Users and IWAM is placed in Web Applications. IISLockdown places a deny write access control entry in the access control list on the web content directories and system utilities. This makes it impossible for any user in the two groups to run system utilities or alter or deface the website. Even it they manage to elevate the processes privileges the deny ACE takes precedence.

Place any custom web application users in the groups so the deny ACE will apply to them as well.

Configuring Active Content

By default Internet Information Server 6 will only serve static pages. To do anything interesting some form of active content will have to be enabled. An administrator will have to manually enable ISAPI filters, ISAPI extensions and their associated file extensions.

Worker Process Isolation

"One significant improvement in IIS 6.0 is a new fault-tolerant request processing architecture that greatly boosts the reliability of Web sites and applications."[11] IIS 6 uses application pools to host site and applications. These application pools are independent from the core web server, HTTP.SYS, and run as an unprivileged user by default. While it was possible to completely isolate web

[11] Microsoft, "Security Enhancements," p. 5

processes under IIS 5 there was a performance penalty. IIS 6 allows this to be done with no performance penalty.

An application pool can be added using the Internet Information Services Manager. Open the IIS Manager, expand the server and right-click “Application Pools”->”New”->”Application Pool”.

A website for application can be assigned to the pool by going to the “Directory” tab on the property pages for the site or application. Select the pool in the “Application Pool” list box.

Web Gardens

If an application is not processor bound, meaning it is frequently waiting on a resource other than the processor, an application pool can be serviced by multiple processes. If one process slows down due to high resource demand another process will continue to accept requests for the web application. Note that if the web application is processor bound adding more processes will not help matters. While not a security feature per se, if an attacker managed to crash one process another in the garden will continue operating. Use the Performance property page of the application pool to add more processes.

Health Monitoring

Internet Information Server 6 will monitor the application pool to ensure that it is responsive to requests. If a pool becomes unresponsive IIS will terminate it and restart the application pool. Use the Health property page of the application pool to configure the ping interval. Further, IIS can be configured to fail an application pool if it crashes a set number of times in a set period of time. This is called “rapid-fail” protection. The server will begin returning error 503 – Service Unavailable.

Also, IIS can be configured to periodically restart worker processes based on time or number of requests processed. This can help keep the server running by dealing with error such as memory leaks that can degrade application performance over time. Also, if a process is compromised by an attacker the compromised processes will be killed after a set period of time.

Alternately, IIS can be configured to orphan a misbehaving process instead of killing it so that diagnostics may be run on the active process.

.NET Framework

The .NET Framework is a development environment created by Microsoft. Along with the new Framework came a web application development model called ASP.NET. The ASP.NET development model has a great many advantages over other models such as Java and classic ASP. It also has many features that, while helpful in development and debugging, would also be helpful to an attacker.

HttpForbiddenHandler

The settings for ASP.NET are stored in an XML file called machine.config in c:\Windows\Microsoft.NET\Framework\(version number)\CONFIG. ASP.NET maps some file extensions to its own handlers "below" the IIS ISAPI handler aspnet_isapi.dll. These handlers know how to compile and execute specific types of code modules such as web pages (.aspx) and remoting interfaces (.rem and .soap).

The types of modules that are not needed should be mapped to the "System.Web.HttpForbiddenHandler". This handler will simply return an error page indicating that the requested type of page is not served. For instance, files with extensions such as .vb, .vbproj and .config contain code and configuration information and should never be served to remote clients. They are all mapped to the HttpForbiddenHandler by default so that they can never be served if any are erroneously copied to the web server.

One note, though. The HttpForbiddenHandler returns a message stating that specific resource may not be served. This indicates that the specific resource exists, however, perhaps giving an attacker a clue about how to proceed with an attack. This is called information leakage. An arguably more secure configuration is to map the offending extensions to the 404.dll handler in Internet Information Server itself. The 404 error message indicates that the specified resource was not found.

Remoting (.rem and .soap) is an especially good candidate for 404.dll or HttpForbiddenHandler. Remoting is a powerful inter-process communication system introduced in the .NET framework. A web server should never serve objects via remoting. That should be done by middle tier application or data servers behind and protected by the firewall and web server.

Debugging and Tracing

ASP.NET includes informative debugging and tracing tools which would provide a plethora of information to an attacker. Ensure that these are disabled.

Disable tracing in the machine.config file by setting the "enabled" value of the "trace" element to "false". Alternatively, if tracing is needed to debug a production problem, set "enabled" to true and set the "localOnly" value to "true" so that tracing information is returned to local clients only.

The .NET framework uses "Just-In-Time" compilation to produce the binary code executed by the processor. This allows the .NET runtime to examine and verify that the code is safe to run before being compiled into binary. Disable debug builds by setting the "debug" value of the "compilation" element to "false." This setting keeps the JIT engine from producing debugging output.

Custom Errors

The ASP.NET framework will return very informative error message including stack dumps. This feature should be disabled on production web servers. In the

machine.config file set the “mode” value of the “customErrors” element to “RemoteOnly” or “On”.

This setting is somewhat counter-intuitive. When the mode setting is “Off” the standard, detailed ASP.NET error message is returned to all clients in the event of an error. When set to “On” a simple message is returned indicating that an unspecified error has occurred. When set to “remoteOnly” the simple message is returned to remote clients and the detailed message is returned to local clients. In addition a default custom page can be set using the “defaultRedirect” value.

Code Access Security

Security in the .NET framework is not based the user who is executing the code, but the properties of the code itself. For instance, is the code cryptographically signed and from where did it originate? This is called “Code Access Security.” A thorough discussion of the Code Access Security (CAS) is well beyond the scope of this document. Suffice to say, the .NET runtime knows the origin of the code it is going to run. The security policy can be configured to disallow the running of code that originated from a network, either the Internet or the local network. Use the “Microsoft .NET Framework Configuration Tool” to change the permission sets associated with the “Internet_Zone” and “LocalIntranet_Zone” to Nothing. This will prohibit any .NET code that did not originate on the local host from running.

Conclusion

Microsoft has made immense progress in securing Internet Information Server 6.0. It is more secure in its design and more secure in its default installation. Features such as logging and process handling have been improved such that running a more secure installation does not degrade performance as under IIS 5. This paper detailed some further steps to take to secure the web server including hardening the underlying operating system in addition to the web server itself. Microsoft publishes a wealth of information about securing IIS which can be found at the Microsoft Security Guidance Center (http://www.microsoft.com/security/guidance/prodtech/IIS.mspx)

List of References

Microsoft Corporation. “Technical Overview of Internet Information Services (IIS) 6.0”. (February 2004). http://download.microsoft.com/download/8/a/7/8a700c68-d1af-4c8d-b11e-5f974636a7dc/IISOverview.doc.

Microsoft Corporation. “Security Enhancements in Internet Information Services 6.0”. (August 2003). http://download.microsoft.com/download/a/4/c/a4c57604-f17c-4214-9d64-53084036922e/IISEnhance.doc.

Rohyt Belani and Michael Muckin. “IIS 6.0 Security”. (March 5, 2004). http://www.securityfocus.com/infocus/1765.

Microsoft Corporation. "Internet Information Server returns IP address in HTTP header (Content-Location)". (June 28, 2004). http://support.microsoft.com/default.aspx?scid=kb;en-us;218180.

Nishchal Bhalla and Rohyt Belani. "IIS Lockdown and Urlscan". (January 5, 2003). http://www.securityfocus.com/infocus/1755.

Microsoft Corporation. "How To Use IPSec". (January 2004). http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod111.asp

Microsoft Corporation. "How To Harden the TCP Stack". (January 2004). http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod109.asp.

Microsoft Corporation. "Securing Internet Information Services 6.0". (2004). http://www.microsoft.com/smallbusiness/gtm/securityguidance/articles/sec_iis_6_0.mspx

Microsoft Corporation. "Securing Your Web Server." (January 2004). http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod89.asp

Microsoft Corporation. "How To Use IISLockdown". (January 2004). http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secmod/html/secmod113.asp