GIAC
CERTIFICATIONS

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at http://www.giac.org/registration/gsec

Security Elements of IIS 6.0

© SANS Institute 2003, Author retains full rights.

Anthony DeVoto

GIAC Security Essentials Certification (GSEC)
Practical Assignment Version 1.4b, **Option 1**

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Summary

Microsoft's latest release of its web server product (IIS 6.0) has been dramatically improved upon over earlier versions. It was completely re-architected and developed on the Windows Server 2003 platform. IIS 6.0 runs on the new Windows Server 2003 platform exclusively and has a Web Edition optimized for serving web content and applications. Improvements made in the product were designed for increased performance, reliability, scalability, and security. This discussion will focus on the security elements of IIS 6.0 as well as the security improvements made to those elements in this release.

Although significant steps were taken by Microsoft to tighten up security in its products, IIS 6.0 must still be configured correctly by a trained administrator and should be closely monitored afterward for any sign of unauthorized activity. Another prudent step in system security is to review and apply any necessary security patches as they are released. We'll take a look at the parts of IIS 6.0 that contain security related features that you should be familiar with before planning to deploy and administer this product. Combined efforts between Microsoft and system administrators to secure IIS are an ongoing process that will provide a more secure environment in which to operate. Microsoft has done its part and made significant improvements to its web server security both in its code and its default deployment configuration, as we will see. This release of IIS 6.0 should convince customers of Microsoft's commitment to enhance security in its products.

Trustworthy Computing

Before I talk about the specific security elements of IIS 6.0, it is important to note that Microsoft has undertaken a *get serious* approach to securing its products. Microsoft's *Trustworthy Computing* initiative, as it is called, has brought about significant change in the way Microsoft develops its products. IIS 6.0 is no exception. *Trustworthy Computing* consists of what is considered *The Four Pillars of Trustworthy Computing* (Security Enhancements 2-3). They are as follows:

- *Secure by Design* – This is where Microsoft concentrates its efforts on security during the development process. IIS 6.0 gets a new architecture.
- *Secure by Default* – Microsoft has done an about-face here, and is now turning nearly all features and services off by default, leaving it up to the administrator to turn on only what is needed. IIS 6.0 is installed in locked down mode.
- *Secure in Deployment* – System administrators must be able to effectively manage and monitor Microsoft products with security in mind.
- *Communication* - Microsoft will provide accurate and timely security information to its customers and users. They will also provide training in order to keep systems secure.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The first two pillars pertain to the code itself and the configuration of IIS 6.0 as shipped. The last two steps take place once the product is up and running. A lot has changed in IIS 6.0, and now we will take a closer look at how these concepts apply to this release.

IIS 6.0 Architecture

Microsoft decided to rebuild IIS 6.0 from the ground up in order to accommodate new and enhanced security features while also improving performance and scalability. Windows Server 2003 provides the underlying framework that IIS 6.0 was built upon. New to Windows Server 2003 is the Application Server Role. This is a technology that provides the infrastructure and services to applications such as IIS 6.0. It is from within this role that the IIS 6.0 components can be installed among others.

IIS 6.0's new architecture includes a kernel-mode HTTP protocol stack (HTTP.sys), the WWW Service Administration and Monitoring Component (w3svc service), Worker Processes (w3core), and Inetinfo.exe, the latter three running in user-mode as figure 1 points out. The operating system runs applications in user-mode, and user-mode processes do not have any direct communication with hardware. Kernel-mode processes, however, can communicate directly with hardware, as is the case with HTTP.sys directly interfacing with the network interface card (NIC). Two main components (HTTP.sys and w3svc service) allow for separation of the actual processing of web site code from the operation of the web server. This design makes it possible to achieve greater performance and a higher level of security. See figure 1. (Hill "Features" 8).

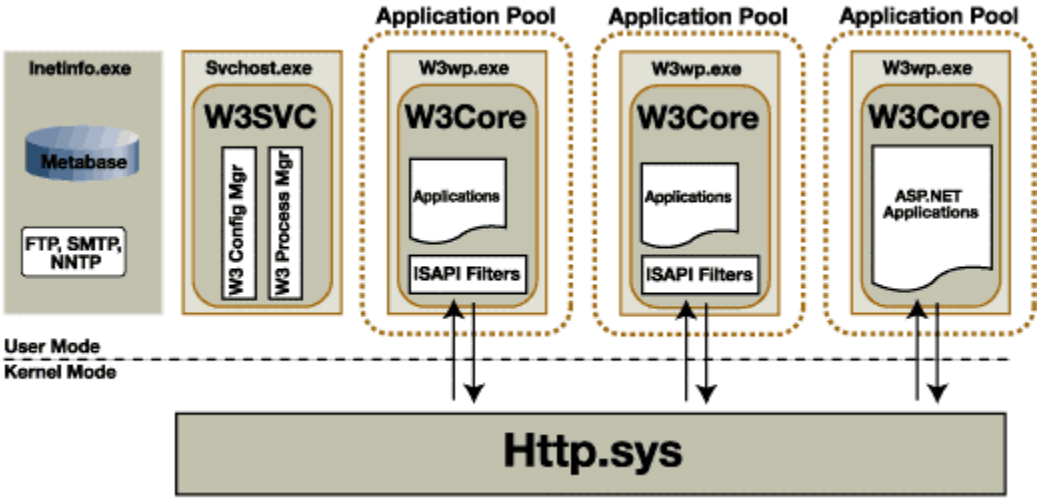

Figure 1.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

HTTP.sys HTTP.sys is a device driver that is part of TCP/IP and is used to route web requests directly to the user mode process that runs the web site where the code is executed. Prior versions had to take several steps to determine where the request should be routed and used Winsock as the mechanism to accept the requests. Now HTTP.sys does so by creating a request queue that directly corresponds to an application pool. HTTP.sys itself however, runs entirely in kernel-mode eliminating the performance hit of transitioning between user mode and kernel mode operations. Since no user code runs here, it can't be taken down by bad code or exploits. And since it runs in kernel-mode it is highly efficient and can continue to accept requests and queue them up until a failed worker process is restarted and starts servicing the application pool again. *(Worker Processes and Application Pools are described below).* It can also cache content and directly process requests in certain circumstances. HTTP.sys also provides URL checking, and logging functions. URL's are checked for proper length and formation. In its logging duties it writes to a log before requests are processed so that failed processes can still be traced (Malhotra np).

WWW Service Administration and Monitoring Component (svchost.exe or W3SVC) The WWW Service Administration and Monitoring component is the service that enables administration, configuration, and process management of your IIS 6.0 web server. In its administrative role it interacts with the metabase, thereby initializing HTTP.sys with the proper routing information needed to communicate with application pools. When the service is utilizing process management capabilities, it is monitoring and controlling worker processes. "WWW Service Administration and Monitoring runs in user mode in a non-shared svchost.exe, and runs as LocalSystem" (Microsoft TechNet "IIS Core" 2). LocalSystem is a special account that I will cover in the *IIS and Built-in Accounts* sub-section of *Access Control*. Since it is separated out from the processes that run web sites and applications, the w3svc will continue to be available even when applications fail.

Worker Process (w3wp.exe or W3Core) New to IIS 6.0 architecture is the worker process. A worker process is what services both the requests for web sites and applications in application pools. They operate independently of one another so that a failure in one process does not affect another whether they are from the same application pool or a separate one. Worker process is controlled by WWW Service Administration and Monitoring (Technical 3-5). It runs in user mode and typically processes requests to return static pages, invokes Internet Server API (ISAPI) extension or filters such as ASP, or runs a Common Gateway Interface (CGI) handler (Microsoft TechNet "IIS Core" 2). Worker processes run as NetworkService, a new built-in account. This account is a low-privilege account. It is one of the most significant security improvements in IIS 6.0 (Malhotra np). This account doesn't have write access to Inetpub or execute access to most of system32. This default set-up would successfully stop a part of the CodeRed attack in where it tries to execute an exe (Hill "Overhauled" 5). Although you can change the account in which a worker process runs, it is best

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

to keep this account to the minimum rights necessary. It is also a good idea to create new accounts or *identities* for each application pool so that if one account is somehow compromised, the others will not be. Remember to include it in the IIS_WPG group so that it has the proper permissions. Worker processes are monitored and a new one is started once a failure is detected. HTTP.sys request queue will then send the requests it has queued up for that particular application pool without the user ever knowing about the failure. Here we see a clear benefit in the new architecture. The user is connected to HTTP.sys not the application, thus providing a more reliable service.

Application Pools Application pools are created for partitioning web sites and applications into distinct areas of management. An application pool directly corresponds to a request queue within HTTP.sys and the worker processes that process the request. After completing a new install of IIS 6.0 you are automatically running in what is called Worker Process Isolation Mode. All applications in a single pool share one or more worker processes. Different pools have different worker processes assigned and one cannot affect the other. In previous versions if a web application failed it could cause the failure of other web sites and applications on the same server (Internet 1).

Application pools provide a feature called recycling. When a worker process is hung or it has consumed a predefined limit of memory or CPU, the process can be restarted. Using a technique called *overlap recycle* restarts the worker process (Hill “Overhauled” 6). By default, recycling happens every 29 hours. This procedure starts a new worker process before destroying the old, meanwhile continuing to service requests. Another option for trouble-shooting purposes is to leave the failed worker process running by orphaning it from the application pool (Internet 3).

Inetinfo.exe Inetinfo is where other service components of IIS 6.0 run such as FTP, SMTP, NNTP, and the metabase. “In IIS 6.0, the services that run in Inetinfo.exe run as dynamic link libraries [DLLs] with LocalSystem identity (the account the services run as). Because this account can be used to gain access to virtually every resource on the local computer, the LocalSystem account should be used with caution, especially on computers that provide services on the Internet.” (Microsoft TechNet “IIS Core” 2). Inetinfo is also utilized when IIS 6.0 is configured to run in IIS 5.0 isolation mode. Certain Internet Server API (ISAPI) filters may cause trouble running in the default out-of-process context of IIS 6.0. Running code out-of-process means the application code is executed in a separate executable (exe) and memory space from its parent process. An example of this would be Inetinfo spawning off dllhost.exe to run a web application within that context. In-process running code is where the application code runs within the original service exe and memory space. In cases where IIS 6.0 worker process isolation mode does not work for your application, you can use IIS 5.0 isolation mode and still benefit from other architectural improvements such as HTTP.sys.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Metabase Similar in functionality to the registry, the metabase stores the configuration information for IIS. The metabase in IIS 6.0 is now stored in Extensible Markup Language (XML) format as opposed to its binary implementation in previous versions. Only administrators can write to the metabase and sensitive information is encrypted within it. In past versions anonymous users had some level of access to the metabase but that is no longer the case (Malhotra np). While I don't see any glaring security improvements to note, I thought it would be beneficial to at least note the change in format since the metabase is a key component of the overall architecture. The metabase's main benefits come from a manageability and performance standpoint, which is outside the scope of this discussion.

For a comprehensive diagram of IIS 6.0's architecture, click the link below.

http://www.sqlservercentral.com/columnists/ckempster/IIS6_Map.pdf (Kempster np)

Elements of Security and Enhanced Features

With the introduction of IIS 6.0 come many changes and enhancements to its security features. Over the years, previous versions of IIS have proven to be major targets for hackers because of the product's lack of strong security. Microsoft, using its Trustworthy Computing initiative, incorporates security features that will help administrators overcome many of the burdens they were accustomed to in earlier versions.

To begin, IIS 6.0 is no longer installed by default (Internet 6), as was the case with Windows 2000. One exception to this rule is the Web Edition of Windows Server 2003. Many unknowing or poorly trained administrators were running web servers even if they didn't need to once they installed Windows 2000 server. And in Windows 2000/IIS 5.0, most features on the web server were turned on by default. This made the amount of attack points for hacking a huge problem since there were a number of vulnerabilities present in IIS 5.0. During an upgrade from IIS 5.0 to 6.0 the www service is disabled unless you have already run the IIS lockdown utility on your Windows 2000 server before starting the upgrade. Microsoft strongly encourages you to do this as it will disable or remove unnecessary features that would otherwise be carried over to IIS 6.0 (Microsoft TechNet "What's" 1).

Once IIS 6.0 is installed it is running in a locked down state. This means that it only has the capability to serve static web pages and no more. Even command line tools are restricted to run only by administrators helping protect against denial-of-service attacks (Internet 6-7). An administrator must enable the required functions in order to operate the web site as it was intended. This strategy reduces the attack surface of the web server, exposing only what is

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

absolutely necessary for your web site and applications. The feature used for enabling things such as Active Server Pages (ASP), FrontPage Server extensions, and ASP.NET is the Web Service Extensions feature (Henrickson 185).

Besides having IIS installed in a locked down state, administrators now have the ability to disable IIS from installing on particular machines via group policy. This helps eliminate the rouge IIS installations from appearing as they did in the past. Usually those installations were prime targets since administrator were unaware of there existence and the person who installed it would generally not have been security conscious, leaving the web server wide open and unpatched.

Some other security enhancements to note are the strict limits put in place on the way the web server behaves by default. Aggressive time-outs, upload data limitations, buffer overflow protection, write protection on the web content from anonymous users, and responding only to recognized file extensions are all part of the new highly secure settings turned on by default. A couple of tables summarizing these and other security enhancements are available from Microsoft at http://www.microsoft.com/windowsserver2003/docs/iisoverview.doc (Technical 9-10) and http://www.microsoft.com/windowsserver2003/docs/iisenhance.doc (Security 15-16).

Let's now take a closer look at the main configuration options and security elements associated with securing a web server. These include the following topics: Authentication, Access Control, Encryption, Certificates, and Auditing and Logging.

Authentication

The basis of authentication is to either grant or deny access to content based on the supplied credentials of the client. IIS 6.0 provides several different methods for authenticating which can be set at the web site, directory, or file level. Authenticating is only the first step in determining access. Once successfully authenticated, a client or user must have the proper access to resources set via permissions which are discussed in the A*ccess Control* section below.

Anonymous Authentication The most flexible and commonly used method for public Internet access is the anonymous authentication method. This type allows anyone with a web browser to access information on your web server without providing a user id or password. It is the least secure of all the methods but is necessary to allow the Internet community access to your site. Windows uses a built in account IUSR_*computername*, where *computername* is the name of the Windows machine IIS is running on for automatically supplying the credentials. This account belongs to the Guests group, which has restrictions on it that work in conjunction with NTFS permissions. NTFS permissions will allow or deny

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

access based on the Guest accounts access rights. Anonymous authentication is the first authentication method IIS will try. If in fact the permissions for anonymous do not allow access, another authentication method will be tried providing there is another method configured (Microsoft TechNet "Anonymous" 1).*The order of which method is tried is listed in the diagram following the next section after all relevant topics are covered.* A major security enhancement in IIS 6.0 is that anonymous authentication no longer requires the *Allow Log on Locally* user right. *Allow log on locally* is part of the *Interactive logon type*. The default logon type is a property stored in the metabase called *LogonMethod*. The new log on type is *Network_Cleartext*, which is considerably more secure (Henrickson 191).

Basic Authentication Another method available is the Basic Authentication method. This type of authentication requires a user name and password to be entered. The credentials entered are sent over the wire in clear text (no encryption), so this method is a weak security practice. Basic Authentication should only be implemented in conjunction with Secure Sockets Layer (SSL) or on intranets when security is not a major concern (Tulloch 293). SSL will be explained further under *Certificate Authentication*. When using basic authentication NTFS permissions should be carefully planned and applied to the accounts that will gain access. Again we have an improvement in IIS 6.0 in the way that the default logon type is applied to basic authentication. The LogonMethod is set to Network_Cleartext as it is for anonymous authentication. It also used *Interactive* for its logon type in past versions.

Digest Authentication Digest Authentication works much the same way as basic authentication except credentials are passed over the wire as a Message Digest 5 (MD5) algorithm. An MD5 hash is a one-way hash function (a number generated from a string of text). This means that it takes a message and converts it into a fixed string of digits, also called a message digest. A message digest is the representation of text in the form of a single string of digits, created using the hash (Microsoft TechNet "Digest" 1). "Digest authentication requires that the domain controller keep a plain text copy of each password, so it can check that password against the hash sent by the client. Therein lies the security risk." (Henrickson 193).

There are some requirements to be met before you can successfully implement digest authentication as shown in Digest Authentication from Microsoft.com (Microsoft TechNet "Digest" 1). They are:

- IE 5 or later must be used.
- The user and server must be members of the same domain or of a trusted domain.
- Users must have valid Windows accounts stored in Active Directory (Win2k or later). Win2k requires sub-authentication (explained below).

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- The computer running IIS must be running Win2k or later. For the purposes of this discussion we are assuming IIS 6.0, which can only run on Windows Server 2003.
- When operating in worker process isolation mode, LocalSystem account must be used as the identity.

Sub-Authentication is not installed by default in IIS 6.0, as was the case with prior versions. It is used for what was automatic password synchronization for anonymous access in earlier versions. Three requirements to enable sub-authentication are: (Henrickson 191)

- Register the sub-authentication component, iissuba.dll.
- Set the metabase property to enable for AnonymousPasswordSync.
- Set the identity of the application pool to LocalSystem.

Advanced Digest Authentication Advanced Digest Authentication is new to IIS 6.0. It requires that both the domain controller and the IIS server be running Windows Server 2003, a key point since Windows 2000 domain controllers will not allow for this mode of operation.

The difference between digest authentication and advanced digest authentication is that the user credentials are stored in Active Directory as an MD5 hash, not clear text. This caveat makes discovering user id and passwords even more difficult. Both digest authentication and advanced digest authentication are available to Web Distributed Authoring and Versioning or WebDAV (Microsoft TechNet “Advance” 1).

Integrated Windows Authentication Integrated Windows Authentication is a highly secure method for connecting a client to an IIS server. It uses a hash algorithm for sending credentials before sending them across the network. This method was previously known as NTLM or Challenge/Response. When the client is running a supported browser and your domain controllers are running at least Windows 2000, the default authentication mechanism is Kerberos v5, otherwise NTLM is used (Microsoft TechNet “Integrated” 1). When utilizing Kerberos, it requires that the client and server both have a trusted connection to a Key Distribution Center (KDC) and be Active Directory Services compatible. Kerberos is a widely used standard utilizing a ticket based authentication system. It assigns a ticket from the KDC to each user who logs on and then embeds it in the message to identify the sender of the message (Henrickson 197).

Negotiate is an authentication method that provides a wrapper for both Kerberos and NTLM. It helps get around the weakness of Kerberos and NTLM by themselves. Kerberos has difficulty getting past most firewalls yet can pass through proxies easily. NTLM has just the opposite problem where it can get past firewalls, but is usually stopped at proxies (Tulloch 289).

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Certificate Authentication Certificate Authentication makes use of SSL encryption. Certificates also enable the encryption process explained in the encryption section. SSL has two types of authentication available, server certificates and client certificates. “SSL authenticates by checking the content of an encrypted digital identification submitted by the user’s web browser during the logon process. (Users obtain client certificates from a mutually trusted third-party organization.)” (Microsoft TechNet “Certificate” 1). Client certificates can be directly mapped to Windows user accounts on the web server. This way there is no need for additional authentication methods where a user account must be associated with the client accessing the web server.

UNC Authentication UNC (Universal Naming Convention) Authentication determines the credentials to be used for access to remote computers hosting web content. Starting with IIS 6.0, if no credentials are supplied (stored in the metabase), then the client requesting access to remote shares has those supplied credentials automatically passed-through to the remote machine for access (Microsoft TechNet “UNC” 1).

.NET Passport Authentication Another new addition to the feature set of IIS 6.0 is .Net Passport Authentication. It provides a single sign-in service for clients accessing your web site. .NET Passport is a Microsoft developed technology that is a component of the .NET framework. This feature would most likely be implemented in a transaction oriented web site where people are coming to buy goods and services. .NET passport requires IE 4.0 or later, and Netscape 4.0 or later browsers.

With .NET Passport users create a single sign-on name and password to access all .NET Passport enabled web sites. The .NET Passport central server issues the credentials and does the authenticating of clients. It is then up to an administrator to configure the web site permissions for .NET enabled accounts.

.NET Passport servers also provide the ability to store user information in encrypted profiles. This profile contains personal information that can be shared with the web site to speed up the registration process. The single sign-on service is similar to the forms based authentication model that is commonly used today. .NET Passport extends the forms based model in the following ways: (Microsoft TechNet “About .NET” 1)

- A centrally hosted server, rather than individual authentication per site.
- Strong encryption for sign-on and profiles. Each site receives an encryption key.
- The central servers return encrypted sign-on and profile information to the hosting site, giving the ability to write local cookies and avoid redirects back to the central server.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- Web sites never receive a client's password. The authentication cookie is a pair of encrypted time stamps. When signing out a central page deletes the cookies from all the sites that the client visited.

Active Directory can be used to store .NET Passport account information and user credentials. When this is the case, a user authentication token is returned to the corresponding user for logging on to the local computer. In cases where Active directory is not being utilized, the IUSR_*computername* will be used as the authenticating account. One difference with this account when using .NET Passport authentication versus anonymous authentication, is that .NET Passport authentication creates an IIS authentication token for the anonymous account (Microsoft TechNet "Configuring" 1). A disadvantage to using passport is that you no longer have control of the user database and if there is a problem with the link or service itself there is no way to authenticate (Rampling 156).

There is an order to which the above authentication options are processed by IIS along with the options of the following section to determine the access level of a client connecting to an IIS 6.0 web site or application. A chart will follow the next section to summarize this concept.

Access Control

Access control works in conjunction with authentication. Once a client is authenticated, that client must have access to resources via NTFS permissions and web permissions to view your content. There will be cases where you will want to restrict access to resources depending on who is accessing your site. The account accessing your site has specific user rights and permissions assigned to it. User rights are either granted or denied the ability to perform specific actions on a computer or the network. "Permissions are rules associated with an object, such as a file or folder, to regulate which accounts can gain access to that object." (Microsoft TechNet "About Access" 1). The type of granular control of who can access what content is performed with the following access control methods.

NTFS Permissions NTFS permissions allow administrators to control access to files and directories based on users and/or groups. Remember the IUSR_*computername* account needs to have at least read permissions assigned to it so that the public Internet users can access your site. NTFS permissions can get quite granular and seemingly complicated when you have many users or groups to assign different sets of permissions (Microsoft TechNet "NTFS" 1).

Web Site Permissions Web permissions are set for and always include everyone accessing your site. Web permissions work in conjunction with NTFS permissions, although there is not the type of granular control as with NTFS permissions. Web permissions apply to virtual directories and when a conflict in

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

access between NTFS and Web permissions happens, the most restrictive setting takes precedence.

"In conclusion, NTFS permissions are a characteristic of the Windows file system and are enforced by the operating system, while web permissions are a characteristic of HTTP and are enforced by IIS." (Tulloch 280).

IIS and Built-in Accounts While we are talking about permissions it is important to note the accounts that IIS makes use of. Microsoft.com *IIS and Built in Accounts* gives a detailed look at this and how these accounts interact with access control permissions and rights, explained below (Microsoft TechNet "IIS and Built" 1-2*)*.

- **LocalSystem** In previous versions of IIS, most web applications were configured to run under this built-in account. Unfortunately, this was not the greatest way of implementing a web application since this account has a high level of access rights. Many exploits were targeted against IIS and having this account configured with all its privileges for an application was a serious threat to the security of the system. The Inetinfo process runs within this context.
- **NetworkService** A built-in account that IIS 6.0 now uses as a default account for worker processes instead of the LocalSystem account within Inetinfo. It has fewer access rights but just enough to do its job. This account has access to interact across the network.
- **LocalService** Another built in account new to IIS 6.0. It has fewer access rights than the NetworkService account by limiting its authority to only have access to the local computer resources and not any resources across the network.
- **IIS_WPG** New again to IIS 6.0, this group has minimal permissions and user rights under the context of the NetworkService account. It is used for starting the worker processes. All accounts that start worker processes must be a member of this group.
- **IUSR_*computername*** As we've stated earlier this account is a guest account used for anonymous access to your web site. Special note for IIS 6.0 is that this account now has the deny write permission assigned to it for all web content.
- **IWAM_*computername*** In IIS 6.0 this account is used for backwards compatibility with IIS 5.0. It is used when an application is set to run in out-of-process IIS 5.0 isolation mode. It launches worker processes and is a member of the IIS_WPG group.
- **ASPNET** A built-in account used for running ASP.NET worker process in IIS 5.0 isolation mode.

TCP/IP Address and Domain Name Access Another form of access control is to set restrictions based on IP address or a domain name. For instance, IP address access control can include a single machine, a group of machines, or an

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

entire subnet for granting or denying access. Domains are either granted or denied access based on their domain name system (DNS) name. By default all computers and domains are granted access.

URL Authorization is yet another form of access control for your web site. URL Authorization in IIS 6.0 is implemented with Windows Server 2003 Authorization Manager. It eases the configuration of access by authorizing user access to the URLs instead of files and directories. Users request URLs and IIS URL authorization validates the user's access based on that user's role. Roles are the set of rules assigned to the user account requesting access. A role can be defined in Lightweight Directory Access Protocol (LDAP) queries, custom user roles, and by Authorization Manager scripts (Microsoft TechNet "URL" 1).

TCP/IP Port Filtering The last form of access control to talk about is TCP/IP Port filtering. While the firewall will take care of the filtering needs of traffic that originates on the outside, it is possible that you may want to filter the ports on the web server to provide an additional layer of security on the inside. This will, however, impact response time since the server must now take on the added responsibility of inspecting each packet as it arrives for access control.

The process in which access is either granted or denied encompasses the authentication and access control sections we've just discussed. Figure 2 shows a summary of how that process works (Tulloch blueprints 7).

© SANS Institute 2003, Author retains full rights.

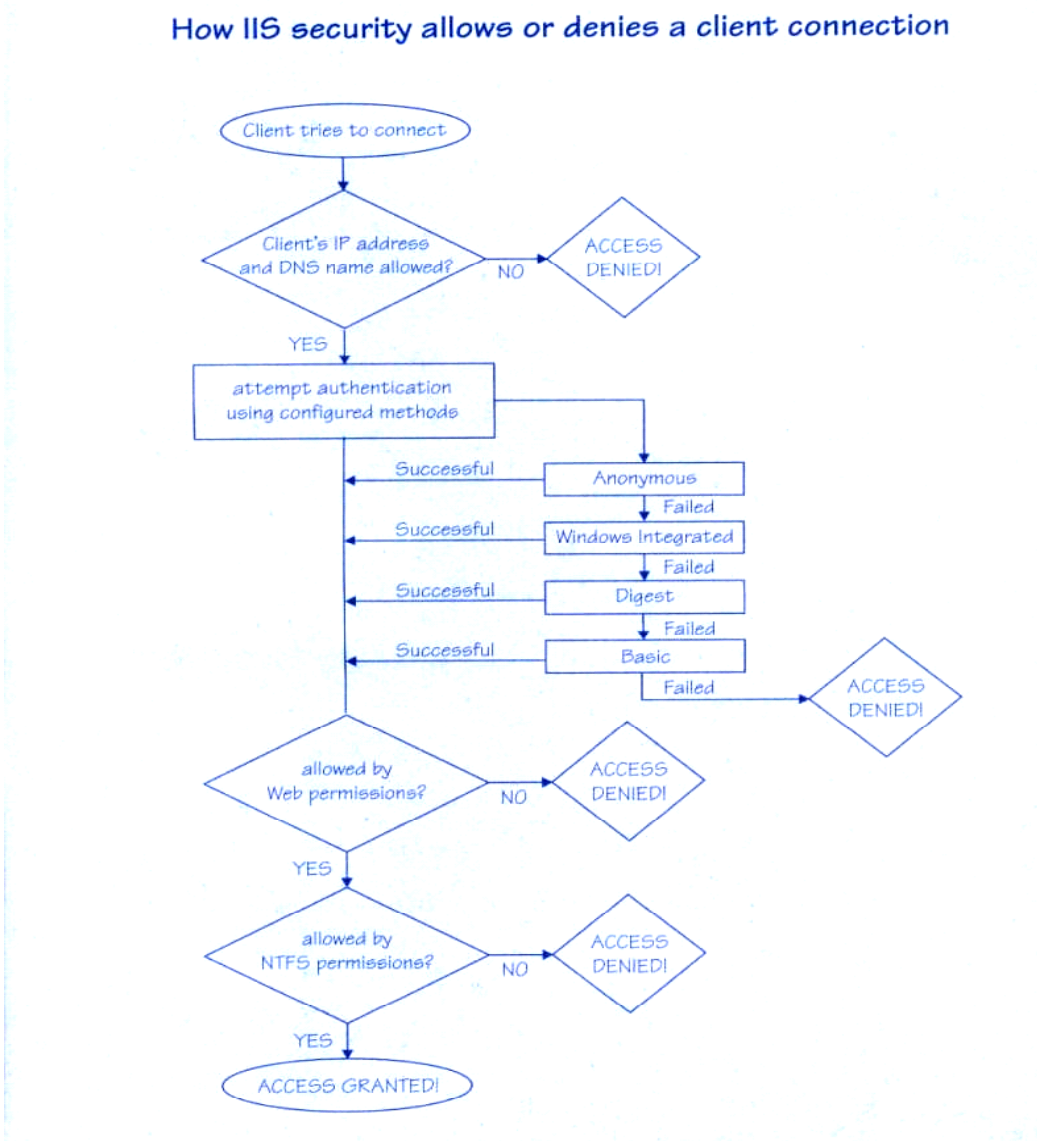

Figure 2.

Encryption and Certificates

Encryption is implemented in IIS 6.0 via Secure Sockets Layer (SSL). SSL enables a secure communication channel to be established between the client and the server. “SSL operates by providing a layer between the application protocol, such as HTTP, and the transport protocol, TCP.” (Rampling 205). The server must have a valid server certificate installed (typically from a trusted third party such as Verisign or Thwate). The use of SSL should be limited to sensitive communication only, such as credit card numbers and personal information, since SSL puts an added burden on the server and may reduce transmission rates and server performance.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Details of how encryption works are outside the scope of our discussion but basically, a session key is used by both the client and server to encrypt and decrypt information. Public key encryption expands upon this method by adding two more keys to the sequence, private and public keys. Essentially, the server sends the client its public key once a secure session has been established. The client then sends back its encrypted information produced with the public key. The server, using its private key, decrypts the message, generates a session key, encrypts it with the public key, and sends it back to the client. The session key is now established for the client and server for encrypt/decrypt functions. Note there are different levels of encryption strength such as 40-bit and 128-bit among others. The higher the value the more secure the encryption is (Microsoft TechNet "About Encryption" 1-2).

Selectable Cryptographic Service Provider (CSP) New to IIS 6.0 is the Selectable Cryptographic Service Provider (CSP). Since cryptography (encrypt and decrypt operations) degrades server performance, Microsoft has added the CSP, which allows for the use of an add-on hardware based solution for the cryptographic computations and creation of public and private keys. This offloads the resource burden and leaves the CPU to take care of other tasks (Microsoft TechNet "Security Features" 1).

Certificates While certificates are an authentication mechanism, they also contain the encryption keys used in establishing an SSL session. "Server certificates contain information about the server that allows the client to positively identify the server before sharing sensitive information. Client certificates contain personal information about the clients that are requesting access to your site, enabling you to positively identify them before allowing them access to the site." (Microsoft TechNet "About Certificates" 1)

As stated earlier, you may obtain the server certificate from a trusted third party. The third party will validate your identification making sure you are who you say you are, and issue the certificate. Once obtained it must be installed on the web server. This certificate will contain the public key.

The clients' certificates contain information about them and also contain the public keys. The key pair between the server and client facilitates the encryption and decryption of data between one another (Microsoft TechNet "About Certificates" 2).

Auditing and Logging

Once your site is up and running you should periodically review the security logs of your server. There are a couple of places where you can set the server and IIS to log activity. First is the security event log. It can be found in the event viewer snap-in for event log monitoring. In order to see certain events you may be looking for, you must first enable auditing. This is set either at the local policies of

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

that machine or at the file system level properties dialog. Events such as logon, account management, and file access activity appear in the event viewer security log. Events from COM objects will be logged in the application log of event viewer. Services such as the w3svc will send its events to the system log in even viewer.

The next type of log is the IIS log that is now handled by HTTP.sys. Again you can set what you want to log and this is done on the web site properties web site tab. This is useful for logging activity types such as who visited and what they were viewing along with the time of that activity. Different formats are available for the log but the most common may be the W3C Extended format for its wealth of information that can be tracked. HTTP.sys sometimes has an error that it will be unable to write to this log so a new log has been created just for that purpose. It's the httperr.log found in \system32\logfiles (Hill "Features" 1).

Additional Steps for Administrators

Besides the specific securing actions you can take on the IIS 6.0 features and components, you should also look for points of weakness at the operating system level and its configuration. In order to ensure that your web servers attack surface is minimized, you will want to disable any unnecessary services including IIS 6.0 specific services not being utilized. Leaving them in a manual startup state might not be good enough since a successful attack could start a service and take advantage of it. Microsoft provides a variety of recommended service startup tables in *Securing Websites and Applications,* which can be found at http://download.microsoft.com/download/5/5/c/55c6de09-c9a5-421c-92a5-9987938dc5af/05_CHAPTER_3_Securing_Web_Sites_and_Applications.doc (Steen 41-62).

Other steps to take, which are also outlined in that same chapter (63-66):

- Renaming the built-in Administrator account is always a good idea. Having a complex password to this account will also help out in slowing down potential attackers.
- Make sure that no development environments exist on the production web server. This is important in ensuring that an attacker cannot remotely compile a program on your server.
- Disable NetBIOS over TCP/IP if possible.
- Store web site content on a physically separated disk volume from the operating system. This helps prevent directory traversal attacks. A directory traversal attack is when a request is sent for a file that is located in another directory structure.

Patch Management Another important part of maintaining the security of your web server is to install software security patches after they have been tested and deemed necessary. IIS 6.0 allows administrators to install most patches without

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

service interruptions. This is accomplished through worker process recycling (Technical 27). As we spoke about before, worker process recycling is the method of firing up a new worker process to service requests while the original one terminates.

Finally implement, review, and revise corporate security policies, processes, and procedures as necessary. Having written documentation for all to follow will ensure consistency in how you run your web sites and help mitigate any risks.

Conclusion

It takes a thorough understanding of the concepts we've just discussed to effectively secure and manage IIS 6.0. Microsoft made significant improvements to its web server security in the IIS 6.0 product, and although significant, you should take a look at some other areas of security that will compliment the efforts of Microsoft. I have pointed out the different elements within IIS 6.0 that pertain to securing the product, but securing your systems goes beyond focusing on the product itself.

A comprehensive security plan would include policies that require physical security, hardware and software infrastructure security, and training for all employees including the administrators among other elements. The physical security element could help reduce the physical exposure of computer systems to unauthorized personnel. Firewalls and anti-virus software are examples of a security infrastructure system. Having well trained employees can help mitigate social engineering types of attacks and increase awareness of potential security breaches. Making sure administrators are well trained will reduce the likelihood that systems will be misconfigured. Keeping secure is an ongoing process that must be dealt with continually as new types of threats are introduced regularly.

As I write parts of this paper, I am sitting at my desk at work after hours and come across a security guard. I don't know him and surely he doesn't know me. He acknowledges that I am here but doesn't ask who I am or what I'm doing here. Shouldn't he have asked for ID and made a note of this unusual occurrence? This is just an example of how deeply you can think in terms of overall security for your network and systems. While this discussion focused on one aspect of security within a product, the concept of overall security goes beyond that of just technology.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Works Cited

Henrickson, Hethe, and Scott Hofmann. IIS 6: The Complete Reference. Emeryville: McGraw Hill/Osborne, 2003

Hill, Brett. "IIS 6.0 Features." May 2003 URL: http://www.winnetmag.com/windowsserver2003/index.cfm?articleid=38496

Hill, Brett. "IIS Overhauled in Version 6.0." April 2003. URL: http://www.winnetmag.com/windowsserver2003/index.cfm?articleid=38285

"Internet Information Services 6.0 Features." Microsoft. 24 April 2003. URL: http://www.microsoft.com/windowsserver2003/iis/evaluation/features/default.mspx

Kempster, Chris. "Microsoft Internet Information Server (IIS) 6.0 Map." July 2003. URL: http://www.sqlservercentral.com/columnists/ckempster/IIS6_Map.pdf

Microsoft TechNet. Accessed 22 August 2003. Various IIS 6.0 Sections URL:

"About .NET Passport Authentication" http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_abtpassport.asp

"About Access Control" http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_acc_aboutacc.asp

"About Certificates" http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_certabout.asp

"About Encryption" http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_encryp_aboutencryp.asp

"Advanced Digest Authentication" http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_advdigestauth.asp

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

"Anonymous Authentication"

"Certificate Authentication"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_certauth.asp

"Configuring Active Directory for .NET Passport Authentication"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_adpassport.asp

"Digest Authentication"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_digestauth.asp

"IIS and Built-In Accounts"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_acc_wpprivileges.asp

"IIS Core Components"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/arc_core.asp

"Integrated Windows Authentication"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_intwinauth.asp

"NTFS Permissions"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_acc_ntfspermovr.asp

"Security Features"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/gs_secfeatures.asp

"UNC Authentication"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_auth_uncauth.asp

"URL Authorization"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/sec_acc_urlauth.asp

"What's Changed"
http://www.microsoft.com/technet/prodtechnol/windowsserver2003/proddocs/standard/gs_whatschanged.asp

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Malhotra, Vik and Richard Ersek. “Isolating and Securing Web Applications in IIS 6.0.” Support Web Cast: Microsoft Internet Information Services. 11 March 2003. URL: http://support.microsoft.com/default.aspx?scid=%2fservicedesks%2fwebcasts%2fwc031103%2fwcblurb031103.asp

Rampling, Blair. Windows Server 2003 Security Bible. Indianapolis: Wiley Publishing, Inc., 2003.

“Security Enhancements in Internet Information Services 6.0.” Microsoft. May 2003. URL: http://www.microsoft.com/windowsserver2003/docs/iisenhance.doc

Steen, Doug et al. “Securing Web Sites and Applications.” Microsoft ® Windows ® Server 2003 Deployment Kit: Deploying Internet Information Services (IIS) 6.0. Microsoft Corporation, 2003

“Technical Overview of Internet Information Services (IIS) 6.0.” Microsoft. April 2003. URL: http://www.microsoft.com/windowsserver2003/docs/iisoverview.doc

Tulloch, Mitch. IIS 6 Administration. Emeryville: McGraw-Hill/Osborne, 2003

© SANS Institute 2003, Author retains full rights.