NIST
National Institute of
Standards and Technology
Technology Administration
U.S. Department of Commerce

Special Publication 800-44

Guidelines on Securing Public Web Servers

Recommendations of the National Institute of Standards and Technology

Miles Tracy, Wayne Jansen, and Mark McLarnon

NIST Special Publication 800-44

Guidelines on Securing Public Web Servers

Recommendations of the National Institute of Standards and Technology

Miles Tracy, Wayne Jansen, and Mark McLarnon

C O M P U T E R   S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

February 2002

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-44
Natl. Inst. Stand. Technol. Spec. Publ. 800-44, xx pages (Mon. 2002)
CODEN: XXXXX

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2002

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

Acknowledgements

The authors, Wayne Jansen from NIST and Miles Tracy and Mark McLarnon from Booz Allen wish to express their thanks to colleagues at both organizations who reviewed drafts of this document. In particular, their appreciation goes to John Wack, Murugiah Souppaya, and Tim Grance from NIST, and Steve Allison, Scott Bisker, Alexis Feringa, Kevin Kuhlkin, and Jonathan Holleran at Booz Allen, for their research, technical support, and written contributions to this document.

Table of Contents

List of Figures

Executive Summary

The World Wide Web (WWW) is a system for exchanging information over the Internet. At the most basic level, the Web can be divided into two principal components: Web servers, which are applications that make information available over the Internet (in essence publish information) and Web browsers (clients), which are used to access and display the information stored on the Web servers. This document will primarily address the security issues of Web servers.[1]

Unfortunately, in most organizations, the Web server is the most targeted and attacked host on their network. These circumstances result in the need to secure Web servers and the network infrastructure that supports them. The specific security threats to Web servers generally fall into one of the following categories:

- Malicious entities may exploit software bugs in the Web server, underlying operating system, or active content to gain unauthorized access to the Web server. Examples of this unauthorized access are gaining access to files or folders that were not meant be publicly accessible or being able to execute commands and/or install software on the Web server.
- Denial of service (DoS) attacks may be directed to the Web server denying valid users an ability to use the Web server for the duration of the attack.
- Confidential information on the Web server may be distributed to unauthorized individuals.
- Confidential information transmitted between Web server and browser may be intercepted if not encrypted.
- Information on the Web server may be changed for malicious purposes. A common often reported example is Web site defacement.
- Malicious entities may gain unauthorized access to resources elsewhere in the organization's computer network via a successful attack on the Web server.
- Malicious entities may attack external organizations from a successful attack on a Web server host, thus concealing the intruders' identities, and perhaps making the organization liable for damages.
- Use the server as a distribution point for illegally copied software, hacker tools, or pornography, perhaps making the organization liable for damages.

[1] For more information on securing Web browsers, see NIST Draft Special Publication 800-46, *Security for Telecommuting and Broadband Communications* (http://csrc.nist.gov/publications/nistpubs/index.html).

This document has been developed to assist Federal departments and agencies, state agencies and commercial organizations in installing, configuring, and maintaining a secure public Web server. More specifically, this document describes in detail the following practices to apply:

- Securing, installing, and configuring the underlying operating system
- Securing, installing, and configuring Web server software
- Deploying appropriate network protection mechanisms:
  - Firewalls
  - Routers
  - Switches
  - Intrusion detection systems
- Maintaining the secure configuration through application of appropriate patches and upgrades, security testing, monitoring of logs and backups of data and operating system
- Using, publicizing, and protecting information and data in a careful and systemic manner.
- Administrating the Web server in a secure manner:
  - Backups
  - Security testing
  - Updating and patching
  - Log reviews.

1. Introduction

1.1 Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Computer Security Act of 1987 and the Information Technology Management Reform Act of 1996, specifically 15 United States Code (U.S.C.) 278 g-3 (a)(5). This document is not a guideline within the meaning of 15 U.S.C 278 g-3 (a)(3).

These guidelines are for use by federal organizations that process sensitive information. They are consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Appendix III.

The guidelines herein are neither mandatory nor binding standards. This document may be used by nongovernmental organizations on a voluntary basis. It is not subject to copyright.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding upon federal agencies by the Secretary of Commerce under his statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, the Director of the OMB, or any other federal official.

1.2 Purpose and Scope

The purpose of Guidelines on Securing Public Web servers is to present security guidance for the design, implementation, and operation of publicly accessible Web servers. This document is published by the NIST as recommended guidance for federal departments and agencies. It may be used in the private sector on a voluntary basis.

This document should be used by organizations interested in enhancing security on Web server systems, in an effort to reduce the number and frequency of Web-related security incidents. This document presents generic principles that apply to all systems. In addition, specific examples are presented that address two of the more popular Web server applications: Apache and Microsoft Internet Information Server (IIS).

This guideline does NOT cover the following aspects relating to securing a Web site:

- Securing other types of network servers
- Firewalls and routers used to protect Web servers beyond a basic discussion in Section 7
- Security considerations related to Web client (browser) software[2]
- Commercial or sensitive governmental transactions via the Web

[2] For more information on securing Web browsers see draft NIST Special Publication 800-46, *Security for Telecommuting and Broadband Communications* (http://csrc.nist.gov/publications/nistpubs/index.html).

- Special considerations for high traffic Web sites with multiple hosts.
- Securing backend servers that may support Web servers (e.g., database servers, file servers)
- Services other than Hypertext Transfer Protocol (HTTP) and Secure Hypertext Transfer Protocol (HTTPS)
- Protection of intellectual property.

1.3 Audience and Assumptions

The intended audience for this document is varied. It covers details specific to the various components of Web content, Web applications, and Web servers. The document is technical in nature; however, it provides the necessary background information to fully understand the topics that are discussed.

The following list highlights how people with differing backgrounds might use this document:

- System engineers and architects when designing and implementing Web servers
- System administrators when administering, patching, securing, or upgrading Web servers
- Security consultants when performing security audits to determine information system (IS) security postures
- Program managers and information systems security officers (ISSO) to ensure that adequate security measures have been considered for all phases of the system life cycle.

This document assumes that readers will have some minimal operating system and Web server application expertise. Because of the constantly changing nature of the Web server threats and vulnerabilities, readers are expected to take advantage of other resources (including those listed in this document) for more current and detailed information.

1.4 Document Structure

The document is divided into eight sections followed by six appendixes. This subsection is a roadmap describing the structure.

- Section 1 (this section) provides an authority, purpose and scope, audience and assumptions, and document structure.
- Section 2 discusses Web server security problems and presents and overview.
- Section 3 provides general information on choosing and securing the underlying operating system of a Web server.
- Section 4 discusses securely installing and configuring the Web server.
- Section 5 examines the security of Web content.

- Section 6 examines the popular Web authentication and encryption technologies.
- Section 7 discusses implementing a secure network for a Web server.
- Section 8 provides the best practices to securely administering and maintaining a Web server.
- Appendix A provides the details of securing the Apache Web server.
- Appendix B provides the details of securing Microsoft's IIS Web server.
- Appendix C provides a variety of online Web security resources.
- Appendix D defines terms most frequently used in this document.
- Appendix E provides a list of commonly used Web server security tools and applications.
- Appendix F lists the references used in this document.

2. Web Server Security Problems and Overview

The World Wide Web (WWW) is one of the most important ways for an organization to publish information, interact with Internet users, and establish an e-commerce business presence. However, if an organization is not rigorous in configuring and operating a public Web site, the organization may be vulnerable to a variety of security threats. Although the threats in cyberspace remain largely the same as in the physical world (e.g., fraud, theft, vandalism, and terrorism), they are far more dangerous as a result of three important developments: increased profitability, action at a distance, and rapid technique propagation [Sch00].

- **Increased Profitability**. Automation makes attacks, even those with minimal return, extremely profitable. For example, in the physical world an attack that would succeed one in 10,000 attempts would be ineffectual because of the time and effort required for a single success. The time invested in getting a single success would be outweighed by the time invested in the 9,999 failures. On the Internet, automation enables the same attack to be a stunning success. Computing power and bandwidth are getting less expensive daily, and the number of hosts that can be targeted is growing exponentially. This “synergy” means that almost any attack, no matter how low its success rate, will likely be exploited.

- **Action at a Distance**. The Internet allows action at a distance. The Internet has no borders, and every point on the Internet is adjacent to every other point. This means that an attacker in one country can now attack a remote web site in another country without ever leaving home.

- **Rapid Technique Propagation**. The Internet allows for easier and more rapid technique propagation. Before the Internet, techniques for attack were developed that would take years, if ever, to propagate, allowing time to develop effective countermeasures. Today, a new technique can be propagated within hours or days. It is now more difficult to develop effective countermeasures in a timely manner.

Compromised Web sites have served as an entry point for intrusions into many organizations’ internal networks. Organizations can face monetary losses or legal action if an intruder successfully violates the confidentiality of their data. Denial of service (DoS) attacks can make it difficult, if not impossible, for users to access an organization’s Web site. These attacks may cost the organization significant amounts of time and money. An organization can also find itself in an embarrassing situation resulting from malicious intruders changing the content of the organization’s Web pages.

Three main security issues are related to the operation of a publicly accessible Web site [CERT01]:

- Misconfiguration or other improper operation of the Web server, which may result, for example, in the disclosure or alteration of confidential or sensitive information. This information can include items such as the following:

  - Assets of the organization

  - Configuration of the server or network that could be exploited for subsequent attacks

  - Information regarding the users or administrator(s) of the Web server, including their passwords.

- Vulnerabilities within the Web server that might allow, for example, attackers to compromise the security of the server and other hosts on the organization's network by taking actions such as the following:

  - Deface the Web site or otherwise affect information integrity

  - Execute unauthorized commands or programs on the server host machine, including ones that the intruder has installed

  - Gain unauthorized access to resources elsewhere in the organization's computer network

  - Launch attacks on external sites from the Web server host, thus concealing the intruders' identities, and perhaps making the organization liable for damages

  - Use the server as a distribution point for illegally copied software, hacker tools, or pornography, perhaps making the organization liable for damages.

- Inadequate or unavailable defense mechanisms for the Web server to prevent certain classes of attacks, such as DoS attacks, which disrupts the availability of the Web server and prevents authorized users from accessing the Web site when required.

Several steps are required to ensure the security of any public Web server. As a prerequisite for taking any step, however, it is essential that the organization have a security policy in place. Otherwise, it is not possible to determine whether the specific measures taken are effective and appropriate for the needs of the organization. Taking the following steps within the context of the organization's security policy should prove effective:

- Step 1. Securing, installing, and configuring the underlying operating system

- Step 2. Securing, installing, and configuring Web server software

- Step 3. Employing appropriate network protection mechanisms (e.g., firewall, packet filtering router, and proxy)

- Step 4. Maintaining the secure configuration through application of appropriate patches and upgrades, security testing, monitoring of logs and backups of data and operating system

- Step 5. Using, publicizing, and protecting information and data in a careful and systemic manner.

- Step 6. Employing secure administration (including regular testing, updating and log reviews).

The practices recommended in this document are designed to help mitigate the risks associated with these and several other known security problems. They build on and assume the implementation of practices described in the following NIST guidelines as appropriate:

- NIST Special Publication 800-3, *Establishing a Computer Security Incident Response Capability*
- NIST Special Publication 800-18, *Guide to Developing Security Plans for Information Technology Systems*
- NIST Special Publication 800-26, *Security Self-Assessment Guide for Information Technology Systems*
- NIST Special Publication 800-27, *Engineering Principles for Information Technology Security*
- NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*
- NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*
- NIST Special Publication 800-41, *Guide to Firewall Selection and Policy Recommendation*s
- NIST Draft Special Publication 800-42, *Guideline on Network Security Testing*
- NIST Draft Special Publication 800-46, *Security for Telecommuting and Broadband Communications*
- NIST Draft Special Publication 800-40, *Applying Security Patches. (not yet posted)*
- NIST Draft Special Publication 800-43, *Guide to Securing Windows 2000 Professional*

All these guidelines and others can be found at the NIST Computer Security Resource Web site at http://csrc.nist.gov/publications/nistpubs/index.html.

3. Securing the Operating System

The first step in securing a Web server is securing the underlying operating system. All commonly available Web servers operate on a general-purpose operating system. Many security issues can be avoided if the operating systems underlying Web servers are configured appropriately. Default hardware and software configurations are typically set by vendors to emphasize features, functions, and ease of use at the expense of security. Because vendors are not aware of each organization's security needs, each Web administrator must configure new servers to reflect their organization's security requirements and reconfigure them as those requirements change. The practices recommended here are designed to help a Web administrator configure and deploy Web servers that satisfy his or her organization's security requirements. Web administrators with existing Web servers should confirm that their current configurations address the issues discussed here.

The techniques for hardening different operating systems vary greatly; therefore, this section will include the generic procedures common in securing most operating systems. References for securing specific operating systems are provided in Section 3.4. In addition, many organizations maintain their own guidelines specific to their requirements. Some automated tools also exist for hardening the operating system and we recommend considering the use of such tools and others with similar functionality (see Appendix E).

Four basic steps are necessary to maintain basic operating system security:

- Planning, installing, and deploying the Web server operating system
- Configuring the Web server operating system to adequately address security
- Patching and updating the Web server operating system as required
- Testing the Web server operating system to ensure that the previous three steps are adequately addressing all security issues.

3.1 Planning, Installing, and Deploying the Web Server Operating System

As it is much more difficult to address security once deployment and implementation have occurred, security should be considered from the initial planning stage. Organizations are more likely to make decisions about configuring computers appropriately and consistently when they develop and use a detailed, well-designed deployment plan. Developing such a plan will support Web administrators in making the inevitable tradeoff decisions between usability, performance, and risk. Consistency is a critical component of a successful security posture because it leads to predictable behavior. This will make it easier for an organization to maintain secure configurations and will assist in identifying security problems (which often manifest themselves as deviations from predictable, expected behavior).

In the planning stages of a Web server, the following items should be considered [CERT00a]:

- Identify the purpose(s) of the Web server.
  - What information categories will be stored on the Web server?

- What information categories will be processed on or transmitted through the Web server?
- What are the security requirements for this information?
- Is any information retrieved from or stored on another host (e.g., backend database, mail server)?
- What are the security requirements for any other hosts involved (e.g., backend database, mail server, proxy servers)?
- What other service(s) are provided by the Web server (a Web server should run only on a dedicated host)?
- What are the security requirements for these additional services?

■ Identify the network services that will be provided on the Web server, such as those supplied through the following protocols:

- HTTP
- HTTPS[3]
- Secure Hypertext Transfer Protocol (S-HTTP)[4]
- File Transfer Protocol (FTP)
- Remote Authentication Dial-In User Service (RADIUS) Protocol
- Open Database Connectivity (ODBC) Protocol
- Network File System (NFS) Protocol[5]
- Common Internet Files System (CIFS)
- Internet Caching Protocol (ICP).

■ Identify any network service software, both client and server, to be installed on the Web server and any other support servers.

■ Identify the users or categories of users of the Web server and any support hosts.

■ Determine the privileges that each category of user will have on the Web server and support hosts.

[3] HTTP transactions protected via the Secure Socket Layer (SSL) protocol (see Section 6.5).

[4] A seldom-used alternative to HTTPS.

[5] Not generally recommended unless used exclusively within a DMZ for data replication to multiple Web servers.

- Decide if and how users will be authenticated and how authentication data will be protected.
- Determine how appropriate access to information resources will be enforced.

The choice of Web server application may determine the choice of operating system. However, to the degree possible, Web administrators should choose an operating system that provides the following [CERT00]:

- Minimal exposure to vulnerabilities (They all have some!)
- Ability to restrict administrative or root level activities to authorized users only
- Ability to deny access to information on the server other than that intended to be available
- Ability to disable unnecessary network services that may be built into the operating system or server software
- Ability to control access to various forms of executable programs, such as Common Gateway Interface (CGI) scripts and server plug-ins in the case of Web servers
- Ability to log appropriate server activities to detect intrusions and attempted intrusions
- Provide a host-based firewall capability
- Acceptable costs for insurance and liability (some insurers charge more for certain operating systems).

In addition, organizations should consider the availability of experienced trained staff to administer the server and server products, which is a important consideration in server selection.

3.2 Securely Installing and Configuring an Operating System

3.2.1 Patch and Upgrade Operating System

Once an operating system is installed, apply any patches or upgrades to correct for known vulnerabilities. All operating systems released today have some known vulnerabilities that should be corrected before using the operating system to host a Web server. To adequately detect and correct for these vulnerabilities, Web administrators should be aware of the following:

- Announcements of security-related problems[6]
- Steps to take to reduce exposure of the vulnerability
- Permanent fixes (often called patches, hotfixes, service packs, or updates).

[6] To check for operating system or Web server application vulnerabilities, see the NIST ICAT Metabase at http://icat.nist.gov.

For more information on vulnerabilities and patching, see NIST Draft Special Publication 800-40, *Applying Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

3.2.2 Remove or Disable Unnecessary Services and Applications

Ideally, a Web server should be on a dedicated, single-purpose host. Many operating systems are configured by default to provide a wider range of services and applications than required by a Web server; therefore, a Web administrator should configure the operating system to remove or disable them. Some common examples that should usually be disabled would include:

- Windows Network Basic Input/Output System (NetBIOS), if not required
- NFS, if not required
- Telnet
- Network Information System (NIS)
- Simple Mail Transfer Protocol (SMTP)
- Compilers
- Software development tools

Removing unnecessary services and applications is preferable to simply disabling them through configuration settings, because attacks that attempt to alter settings and activate a disabled service cannot succeed when the functional components are completely removed.

Eliminating or disabling unnecessary services enhances the security of a Web server in several ways [CERT00]:

- Other services cannot be compromised and used to attack the host or impair the Web server services. Each service added to a host increases the risk of compromise for that host because each service is another possible avenue of access for an attacker. Less is truly more in this case.
- Different individuals may administer different services. Isolating services so each host has a single administrator will minimize the possibility of conflicts between the administrators. Also, one administrator per host provides better accountability.
- The host can be configured to better suit the requirements of the particular service. Different services might require different hardware and software configurations, which could lead to unnecessary vulnerabilities or service restrictions.
- By reducing services, the number of logs and log entries is reduced; therefore detecting unexpected behavior becomes easier.

When configuring the operating system, apply the principle "disable everything except that which is expressly permitted"—that is, disable or remove (preferable) all services and applications and then selectively enable those required by the Web server. If possible, install the minimal operating system configuration that is required for the Web server application. If the operating system installation system provides a “minimal installation” option, choose that because it will minimize the effort required to remove unnecessary services. Many uninstall scripts or programs do not completely remove all components of service; therefore, it is always better to avoid installing unnecessary services if possible.

The services enabled on a Web server will depend on the functions the organization wants the server to provide. Those services might include database protocols to access a database, file transfer protocols, and remote administration services. Each of these services, even though they may be required, does come with an increased risk to the server. Whether the risks outweigh the benefits is a decision each organization must make for itself.

3.2.3 Configuring Operating System User Authentication

For Web servers, authorized users who can configure the system and initiate Web services are typically a small number of designated Web administrators and Webmasters. However, the users who can access the public Web server may range from unrestricted to restricted subsets of the Internet community. To enforce policy restrictions, if required, the Web administrator must configure the system to authenticate a prospective user by requiring proof that he or she is authorized for such access. Even though a Web server may allow unauthenticated access to most Web services, administrative and other types of specialized access should be limited to specific individuals and groups.

Configuring the computer for authentication usually involves configuring parts of the operating system, firmware, and applications on the server, such as the software that implements a network service. In special cases, for high-value/high-risk sites, organizations may also use authentication hardware, such as tokens or one-time password devices. Use of authentication mechanisms where authentication information is reusable (e.g., passwords) and transmitted in the clear over a network is strongly discouraged, because the information can be intercepted and used by an attacker to masquerade as an authorized user (see Section 6).

To ensure the appropriate user authentication is in place, take the following steps [CERT00]:

- **Remove or disable unneeded default accounts and groups**. The default configuration of the operating system often includes guest accounts (with and without passwords), administrator or root level accounts, and accounts associated with local and network services. The names and passwords for those accounts are well known.[7] Remove or disable unnecessary accounts to eliminate their use by intruders, including guest accounts on computers containing sensitive information. If there is no requirement to retain a guest account or group, severely restrict its access and change the password in accordance with the organizational password policy.

  For default accounts that need to be retained, change the names (where possible particularly for administrator or root level accounts) and passwords to be consistent with the organizational password policy. Default account names and passwords are commonly known in the hacker community.

[7] For an extensive list of default accounts and passwords, see http://www.securityparadigm.com/dad.htm.

- **Disable noninteractive accounts**. Disable accounts (and the associated passwords) that need to exist but do not require an interactive login. For UNIX systems, disable the login shell, or provide a login shell with NULL functionality (/bin/false).

- **Create the user groups**. Assign users to the appropriate groups. Then assign rights to the groups, as documented in the deployment plan. This approach is preferable to assigning rights to individual users.

- **Create the user accounts**. The deployment plan identifies who will be authorized to use each computer and its services. Create only the necessary accounts. Discourage or prohibit the use of shared accounts.

- **Check the organization's password policy, and set account passwords appropriately**. This policy should address the following:

  - **Length**—a minimum length for passwords. Specify at least a minimum length of eight characters.

  - **Complexity**—the mix of characters required. Require passwords to contain both uppercase and lowercase letters and at least one nonalphabetic character.

  - **Aging**—how long a password may remain unchanged. Require users to change their passwords periodically. Administrator or root level password should be changed every 30 to120 days. User password should also be changed periodically with period of time determined by the enforced length and complexity of the password combined with the sensitivity of the information protected.

  - **Reuse**—whether a password may be reused. Some users try to defeat a password-aging requirement by changing the password to one they have used before. If possible, ensure that the user cannot change the password by simply appending or "prepending" characters to their original password (e.g., original password was "mysecret" and is changed to "1mysecret" or "mysecret1".

  - **Authority**—who is allowed to change or reset passwords and what sort of proof is required before initiating any changes.

- **Configure computers to deny login after a small number of failed attempts**. It is relatively easy for an unauthorized user to try to gain access to a computer by using automated software tools that attempt all passwords. If the operating system provides the capability, configure it to deny login after three failed attempts. Typically, the account is "locked out" for a period of time (such as 30 minutes) or until a user with appropriate authority reactivates it.

  This is another situation that requires the Web administrator to make a decision that balances security and convenience. Implementing this recommendation can help prevent some kinds of attacks, but it can also allow a malicious intruder to make failed login attempts to prevent user access, a DoS condition.

  Failed network login attempts should not prevent an authorized user or administrator from logging in at the console. Note that all failed log in attempts whether via the network or console should be logged. Also, if remote administration is not going to be implemented (see Section 8.5), disable the ability for the administrator or root level accounts to log in from the network.

- **Install and configure other security mechanisms to strengthen authentication**. If the information on the Web server requires it, consider using other authentication mechanisms such as tokens, client/server certificates, or one-time password systems. Although they can be more expensive and difficult to implement, they may be justified in some circumstances. When such authentication mechanisms and devices are used, the organization's policy should be reviewed and reflected in the way in which they are applied.

As mentioned earlier, intruders using network sniffers can easily capture reusable passwords passed across a network in clear text. Consider implementing instead less vulnerable authentication and encryption technologies, such as Secure Shell (SSH) and Secure Socket Layer (SSL) (see Section 6.5).

3.2.4 Configure Resource Controls Appropriately

Many operating systems provide a capability to specify access privileges individually for files, directories, devices, and other computational resources. By carefully setting access controls, the Web administrator can reduce intentional and unintentional security breaches. For example, denying read access to files and directories helps protect confidentiality of information, whereas denying unnecessary write (modify) access can help protect the integrity of information. Limiting the execution privilege of most system-related tools to authorized system administrators can prevent users from making configuration changes that could reduce security. It also can restrict the ability of intruders to use those tools to attack the system or other systems on the network. Because operating system resource controls act in tandem with Web server resource controls, this topic is addressed in greater detail in Section 4.2.

3.3 Security Testing the Operating System

Periodic security testing of the operating system is a vital way to identify vulnerabilities and to ensure that the existing security precautions are effective. There are several methods for testing operating systems, the most popular of which are vulnerability scanning and penetration testing. Vulnerability scanning usually entails using an automated vulnerability scanner to scan a host or groups of hosts on a network for application, network, and operating system vulnerabilities. Penetration testing is a testing process designed to compromise a network using the tools and methodologies of a "hacker." It is an iterative testing process whereby the weakest areas of the network are identified and exploited to expand access to the remainder of the network which eventually results in compromising the overall security of the network. Vulnerability scanning should be conducted on a quarterly basis and penetration testing should be conducted on a one to three year basis. Since both of these testing techniques are applicable to testing the Web server application as well, they are discussed in greater detail in Section 8.4. [8]

[8] For information on other testing techniques, see NIST Special Publication 800-42, *Guideline on Network Security Testing* (http://csrc.nist.gov/publications/nistpubs/index.html).

3.4 Resources for Operating System Specific Security Procedures

The following Web sites provide detailed information about securing specific operating systems:

- **UNIX**—CERT *UNIX Security Checklist Version 2.0* (http://www.cert.org/tech_tips/usc20_full.html)
- **Windows NT**—NSA *Guide to Securing Microsoft Windows NT Networks* (http://nsa1.www.conxion.com/winnt/guides/wnt-1.pdf)
- **Windows 2000**—NIST Draft Special Publication 800-43, *Guide to Securing Windows 2000 Professional* (http://csrc.nist.gov/publications/nistpubs/index.html)
- **Windows 2000**— NSA *Guide to Securing Microsoft Windows 2000* (http://nsa1.www.conxion.com/win2k/index.html).

3.5 Securing the Web Server Operating System Checklist

Completed	Action
	Plan the configuration and deployment of Web server
☐	Identify functions of Web server
☐	Identify information categories that will be stored, processed and transmitted through the Web server
☐	Identify security requirements of information
☐	Identify how information is published to the Web server
☐	Identify a dedicated host to run Web server
☐	Identify network services that will be provided and supported by the Web server
☐	Identify users and categories of users of the Web server and determine privilege for each category of user
☐	Identify user authentication methods for Web server
	Choose appropriate operating system for Web server
☐	Minimal exposure to vulnerabilities
☐	Ability to restrict administrative or root level activities to authorized users only
☐	Ability to deny access to information on the server other than that intended to be available
☐	Ability to disable unnecessary network services that may be built into the operating system or server software
☐	Ability to control access to various forms of executable programs, such as Computer Gateway Interface (CGI) scripts and server plug-ins in the case of Web servers
☐	Acceptable costs for insurance and liability (some insurers charge more for certain operating systems)

Completed	Action
☐	Available experienced staff to install, configure, secure, and maintain operating system
	Patch and upgrade operating system
☐	Identify and install all necessary patches and upgrades to the operating system
☐	Identify and install all necessary patches and upgrades to applications and services included with the operating system
	Remove or disable unnecessary services and applications
☐	Disable or remove unnecessary services and applications
	Configure the operating system user authentication
☐	Remove or disable unneeded default accounts and groups
☐	Disable noninteractive accounts
☐	Create the user groups for the particular computer
☐	Create the user accounts for the particular computer
☐	Check the organization's password policy, and set account passwords appropriately (e.g., length, complexity)
☐	Configure computers to deny login after a small number of failed attempts
☐	Install and configure other security mechanisms to strengthen authentication
	Test the security of the operating system
☐	Test operating system after initial install to determine vulnerabilities
☐	Test operating system quarterly to determine new vulnerabilities

4. Securely Installing and Configuring the Web Server

Once the operating system has been installed and secured, it will be necessary to install the chosen Web server software. Before starting this process, read the vendor documentation carefully and understand the various options available during the install process. Also be sure to visit the vendor's Web site or vulnerability database Web site, such as the ICAT metabase (http://icat.nist.gov), to determine if there are known vulnerabilities and related patches available that should be installed or configured as part of the setup process. Only after these preliminary steps are accomplished should the install be started. Note that this section discusses only generic installation and configuration procedures; for specifics on Apache and IIS, see Appendixes A and B, respectively.

4.1 Securely Installing the Web Server

In many respects, the secure install and configuration of the Web server application will mirror the operating system process discussed in the Section 3. The overarching principle, as before, is to install the minimal amount of Web server services required and eliminate any known vulnerabilities through patches or upgrades. If the installation program installs any unnecessary applications, services, or scripts, they should be removed immediately once the installation process completes. During the installation of the Web server, the following steps should be performed:

1. Install the server software on a dedicated host

2. Install the minimal Internet services required

3. Apply any patches or upgrades to correct for known vulnerabilities

4. Create a dedicated physical disk or logical partition (separate from operating system and server application) for Web content

5. Remove or disable all services installed by the Web server application but not required (e.g., gopher, FTP, and remote administration)

6. Remove all sample documents, scripts, and executable code

7. Remove all vendor documentation from server

8. Apply appropriate security template or hardening script to server (see Appendix E)

9. Reconfigure HTTP service banner (and others as required) NOT to report Web server and operating system type and version. (This can be accomplished in IIS using the Microsoft's free IIS Lockdown Tool and in Apache via the "ServerTokens" directive.)

4.2 Configuring Access Controls

Most Web server host operating systems provide a capability to specify access privileges individually for files, devices, and other computational resources on that host. Any information that the Web server can access using these controls can potentially be distributed to all users accessing the public Web site. The Web server software is likely to provide additional file, device, and resource access controls specific to its operation. In cases where resources permission can be set both at the operating system and Web server application, it is important that they are identical or else it is possible that too much or too little access may be granted to users. Web administrators should consider from two perspectives how best to configure these access controls to protect information stored on their public Web server:

- Limit the access of the Web server software to a subset of computational resources

- Limit the access of users through additional access controls enforced by the Web server, where more detailed levels of access control are required.

The proper setting of access controls can help prevent the disclosure of sensitive or restricted information that is not intended for public dissemination. In addition, access controls can be used to limit resource use in the event of a DoS attack against the public Web site.

Typical files to which access should be controlled are as follows:

- Application software and configuration files

- Files related directly to security mechanisms:

  - Password hash files and other files used in authentication

  - Files containing authorization information used in controlling access

  - Cryptographic key material used in confidentiality, integrity, and non-repudiation services.

- Server log and system audit files

- System software and configuration files.

4.2.1 Configuring the Permissions of the Web Server Application

The first step in configuring access controls is to ensure that the Web server executes only under a unique individual user and group identity with very restrictive access controls. Thus, new user and group identities to be used exclusively by the Web server software need to be established. This new user and new group should be made independent and unique from all other users and groups. This is a prerequisite for implementing the access controls described in the following steps. Although the server may initially have to run as root (UNIX) or system/administrator (Windows NT/2000/XP) to bind to Transmission Control Protocol (TCP) ports 80 and/or 443 (used respectively to provide HTTP and HTTPS services), do not allow the server to continue to run at this level of access.

In addition, use the Web server's operating system to limit files accessed by the Web service processes. These processes should have read-only access to those files necessary to perform the service and should have no access to other files, such as server log files. Use Web server host operating system access controls to enforce the following [CERT01]:

- Web content files can be read but not written by Web service processes.

- Web service processes cannot write the directories where public Web content is stored.

- Only processes authorized for Web server administration can write Web content files.

- The Web application can write Web server log files, but log files cannot be read by the Web server application. Only root/system/administrative level processes can read Web server log files.

- Temporary files created by Web server application, such as those that might be generated in the creation of dynamic Web pages, are restricted to a specified and appropriately protected subdirectory.

- Access to any temporary files created by Web server application is limited to the Web service process(es) that created these files.

It is also necessary to ensure that the public Web server cannot save files outside the specified file structure dedicated to public Web content. This may be a configuration choice in the server software or it may be a choice in how the server process is controlled by the operating system. Ensure that such directories and files (outside the specified directory tree) cannot be served, even if users know the names Uniform Resource Locator (URLs) of those files.

To mitigate the effects of certain types of DoS attacks, configure the Web server to limit the amount of operating system resources it can consume. Some examples would include the following:

- Install Web content on a different hard drive or logical partition from the operating system and Web application.

- If uploads are allowed to the Web server, place a limit on the amount of hard drive space that is dedicated for this purpose.

- Ensure that log files are stored in a location that is sized appropriately.

These actions will protect to some degree against attacks that attempt to fill the file system on the Web server host operating system with extraneous and incorrect information that may cause the system to crash. This will also protect against attacks that attempt to fill primary random access memory (RAM) with unnecessary processes to slow down or crash the system, thus limiting Web service availability. Logging information generated by the Web server host operating system may help in recognizing such attacks (see Section 8.1).

In addition, it is often necessary to configure timeouts and other controls to further reduce the impact of certain DoS attacks. One type of DoS attack, when it is perpetrated, takes advantage of the practical limits on simultaneous network connections by quickly establishing connections up to the maximum permitted, such that no new legitimate users can gain access. By setting network connection timeouts (the time after which an inactive connection is dropped) to a minimum acceptable time limit, established connections will time out as quickly as possible, opening up new connections to legitimate users. This measure only mitigates the effects; it does not defeat the attack.

If the maximum number of open connections (or connections that are half-open—that is, the first part of the TCP handshake was successful) is set to a low number, an attacker can easily consume the available connections with bogus requests (often called a SYN flood). Setting the maximum to a much higher number may mitigate the effect of such an attack, but at the expense of consuming additional resources. Note that this is only an issue for Web servers that are not protected by a firewall that stops SYN flood attacks. Most current enterprise-level firewalls protect a Web server from a SYN flood by intercepting the attack before it reaches the Web server.

4.2.2 Configuring Secure Web Content Directory

Do not use links, aliases, or shortcuts in the public Web content file directory tree that points to directories or files elsewhere on the server host or the network file system. If possible, disable the ability of the Web server software to follow links and aliases. As stated earlier, Web server log files and configuration files should reside outside the specified file directory tree for public Web content.

The following steps are required to restrict access to a specific Web content file directory tree:

- Dedicate a single hard drive or logical partition for Web content and establish related subdirectories exclusively for Web server content files, including graphics but excluding scripts and other programs.

- Define a single directory exclusively for all external scripts or programs executed as part of Web server content (e.g., CGI, Active Server Page [ASP]).

- Disable the execution of scripts that are not exclusively under the control of administrative accounts. This action is accomplished by creating and controlling access to a separate directory intended to contain authorized scripts.

- Disable the use of hard or symbolic links (a.k.a., shortcuts for Windows).

- Define a complete Web content access matrix. Identify which folders and files within the Web server document are restricted and which are accessible (and by whom).

Most Web server software vendors provide directives or commands that allow the Web administrator to restrict user access to public Web server content files. For example, the Apache Web server software provides a Limit directive, which allows the Web administrator to restrict which optional access features (such as New, Delete, Connect, Head, and Get) are associated with each Web content file. The Apache Require directive allows the Web administrator to restrict available content to authenticated users or groups.

Many directives or commands can be overridden on a per-directory basis. The convenience of being able to make local exceptions to global policy is offset by the threat of a security hole being introduced in a distant subdirectory, which could be controlled by a hostile user. The Web administrator should disable a subdirectory's ability to override top-level security directives unless that override is absolutely necessary.

In most cases, Web server file directory listings should be disabled. The HTTP specifies that a URL ending in a slash character be treated as a request for a listing of the files in the directory with that name. Web servers should be prohibited from responding to such requests, even if the public can read all of the directory files. Such requests often indicate an attempt to locate information by means other than those intended by the Web administrator or Webmaster. Users may attempt this if they are having difficulty navigating through the site or if a link appears to be broken. Intruders may attempt this to locate information hidden by the Web site's interface. Web administrators should investigate requests of this type found in the Web server log files (see Section 6).

4.2.3 Controlling Web "Bots" Impact on Web Servers

Web bots (a.k.a., agents or spiders) are software applications used to collect, analyze and index Web content. Web bots are used by a numerous of organizations for many purposes. Some examples are as follows:

- Scooter, Slurp, and Googlebot slowly and carefully analyze, index, and record Web sites for Web search engines such as Alta Vista and Google.
- ArchitextSpider gathers Internet statistics.
- Hyperlink "validators" are used by Webmasters to automatically validate the hyperlinks on their Web site.
- EmailSiphon and Cherry Picker are bots specifically designed to crawl Web sites for electronic mail (e-mail) addresses to add to unsolicited advertising e-mail ("spam") lists. These are a common example of a bot that may have a negative impact on a Web site or it users.

Unfortunately, bots can present a challenge to Webmasters and their servers:

- Web servers often contain directories that do not need to be indexed.
- Organizations might not want part of their site appearing in search engines.
- Web servers often contain temporary pages that should not be indexed.
- Organizations operating the Web server are paying for bandwidth and want to exclude robots and spiders that do not benefit their goals.
- Bots are not always well written or well intentioned and can hit a Web site with extremely rapid requests, causing a reduction in or outright DoS for legitimate users.
- Bots may uncover information that the Webmaster would prefer would remain secret or at least unadvertised (e.g., e-mail addresses).

Fortunately, there is a way for Web administrators or the Webmaster to influence the behavior of most bots on their Web site. A series of agreements called the Robots Exclusion Standard (REP) has been created. Although REP is not an official Internet standard, it is supported by most well written and well-intentioned bots, including those used by most major search engines.

Web administrators who wish to limit bots' actions on their Web server need to create a plain text file named "robots.txt." The file must always have this name, and it must reside in the Web server's root document directory. In addition, only one file is allowed per Web site. Note that the robots.txt file is a standard that is voluntarily supported by bot programmers.

There is no requirement that it be used. Thus, malicious bots (such as EmailSiphon and Cherry Picker) will ignore this file.[9]

Robots.txt is a simple text file that contains some keywords and file specifications. Each line of the file is either blank or consists of a single keyword and its related information. The keywords are used to tell robots which portions of a Web site are excluded.

The following keywords are allowed:

- **User-agent**—is the name of the robot or spider. A Web administrator may also include more than one agent name if the same exclusion is to apply to each specified bot. The entry is not case sensitive (in other words “googlebot” is the same as “GOOGLEBOT” and “GoogleBot”).

  A “*” indicates this is the “default” record, which applies if no other match is found. For example, if you specify "GoogleBot" only, then the "*" would apply to any other robot.

- **Disallow**—tells the bot(s) specified in the user-agent field which sections of the Web site are excluded. For example, /images informs the bot not to open or index any files in the images directory or any subdirectories. Thus, the directory "/images/special/" would not be indexed by the excluded bot(s).

  Note that /do will match any directory beginning with "/do" (e.g. /do, /document, /docs, etc.), whereas /do/ will match only a directory named "/do/".

  A Web administrator can also specify individual files. For example, the Web administrator could specify /mydata/help.html to prevent only that one file from being accessed by the bots.

  A value of just “/” indicates that nothing on the Web site is allowed to be accessed by the specified bot(s).

  At least one disallow per user-agent record must exist.

There are many ways to use the robots.txt file. Some simple examples are as follows:

- To disallow all (compliant) bots from specific directories:

```
User-agent: *
Disallow: /images/
Disallow: /banners/
Disallow: /Forms/
Disallow: /Dictionary/
Disallow: /_borders/
Disallow: /_fpclass/
Disallow: /_overlay/
```

[9] Other methods for controlling malicious bots exist; however, they are changing constantly as the malicious bot operators and Web administrators develop new methods of counteracting each other’s techniques. Given the constantly changing nature of this area, discussion of these techniques is beyond the scope of this document.

```
Disallow: /_private/
Disallow: /_themes/
```

- To disallow all (compliant) bots from the entire Web site:

```
User-agent: *
Disallow: /
```

- To disallow a specific bot (in this case the Googlebot) from examining a specific Web page:

```
User-agent: GoogleBot
Disallow: tempindex.htm
```

Note that the robots.txt file is available to everyone. Thus, a Web administrator should not specify the names of sensitive files or folders. If these must be excluded, it is better to use password-protected pages that cannot be accessed by bots. Password protection is the only reliable way to exclude noncompliant bots. See Section 6 for more information on Web based authentication methods.

4.3 Using File Integrity Checkers

A file integrity checker is an application that computes and stores a checksum for every guarded file and establishes a database of file checksums. It allows a system administrator to easily recognize changes to critical files, particularly unauthorized changes. Checksums should be recomputed regularly to test the current value against the stored value to identify any file modifications. A file integrity checker capability is often included with host-based intrusion detection systems (see Section 7.2.2) and is also available separately (see Appendix E).

Although an integrity checker is a useful tool that does not require a high degree of human interaction, it needs to be used carefully to ensure that it is effective. To create the first reference database a file integrity checker requires a system that is known to be in a secure state. Otherwise, cryptographic hashes of a compromised system may be created and therefore create a false sense of security for the tester. The reference database should be stored off line so that an attacker cannot compromise the system and modify the database to hide tracks of the attack. A file integrity checker can also generate false positive alarms. Each file update and system patch implementation changes the file and will therefore require an update of the checksum database. Thus, keeping the database up-to-date may be difficult. However, even if the integrity checker is run only once (when the system is first installed), it can still be a useful activity for determining which files have been modified in case of a suspected compromise. Finally, attackers have demonstrated an ability to modify a file in ways the commonly used 32-bit cyclic redundancy check (CRC) checksum could not detect. Therefore, stronger checksums are recommended to ensure the integrity of data that is stored in the checksum database.

Integrity checkers should be run nightly on a selection of system files that would be affected by a compromise. Integrity checkers should also be used when a compromise is suspected for determining the extent of possible damage. If an integrity checker detects unauthorized system file modifications, the possibility of a security incident should be considered and investigated according to the organization's incident response and reporting policy and procedures.

4.4 Securely Installing and Configuring the Web Server Checklist

Completed	Action
	Securely installing the Web server
☐	Install the server software on a dedicated host
☐	Install minimal Internet services required
☐	Apply any patches or upgrades to correct for known vulnerabilities
☐	Create a dedicated physical disk or logical partition (separate from operating system and server application) for Web content
☐	Remove or disable all services installed by the Web server application but not required (e.g., gopher, FTP, and remote administration)
☐	Remove all sample documents, scripts, and executable code
☐	Remove all vendor documentation from server
☐	Apply appropriate security template or hardening script to server
☐	Reconfigure HTTP service banner (and others as required) NOT to report Web server and operating system type and version
	Configuring Web server host operating system access controls
☐	Configured so that Web content files can be read but not written by Web service processes
☐	Configured so that Web service processes cannot write the directories where public Web content is stored
☐	Configured so that only processes authorized for Web server administration can write Web content files
☐	Configured so that Web application can write Web server log files, but log files cannot be read by the Web server application
☐	Configured so that temporary files created by Web server application are restricted to a specified and appropriately protected subdirectory
☐	Configured so that access to any temporary files created by Web server application is limited to the Web service process(es) that created these files
☐	Installed with Web content on a different hard drive or logical partition than the operating system and Web application
☐	Configured so that if uploads are allowed to the Web server, a limit is placed on the amount of hard drive space that is dedicated for this purpose
☐	Configured so that log files are stored in a location that is sized appropriately
	Configuring a secure Web content directory
☐	Dedicate a single hard drive or logical partition for Web content and establish related subdirectories exclusively for Web server content files, including graphics but excluding scripts and other programs
☐	Define a single directory exclusively for all external scripts or programs executed as part of Web server content (e.g., CGI, ASP)

Completed	Action
☐	Disable the execution of scripts that are not exclusively under the control of administrative accounts. This action is accomplished by creating and controlling access to a separate directory intended to contain authorized scripts
☐	Create the user groups for the computer.
☐	Disable the use of hard or symbolic links (a.k.a., shortcuts for Windows).
☐	Define a complete Web content access matrix. Identify which folders and riles within the Web server document are restricted and which are accessible (and by whom)
☐	Check the organization's password policy, and set account passwords appropriately (e.g., length, complexity)
☐	Use robots.txt file if appropriate
	Using file integrity checkers
☐	Install a file integrity check to protect Web server configuration files, password files and Web content
☐	Update file integrity checksums whenever an upgrade or content changed occurs
☐	Store checksum on protected write once media
☐	Regularly compare checksums

5. Securing Web Content

The two main components to Web security are the security of the underlying server application and operating systems, and the security of the actual content. Of these, the security of the content is often overlooked. Content security itself has two components. The more obvious is not to place any proprietary, classified, or other sensitive information on a publicly accessible Web server unless other steps have been taken to protect the information via user authentication and encryption (see Section 6). The less obvious component of content security is that the way particular types of content are processed on a server can lead to a compromise.

5.1 Publishing Information on Public Web Sites

Little thought is often given to the security implications of the content placed on the Web site. Few organizations have a Web publishing process or policy that determines what type of information to publish openly, what information to publish with restricted access, and what information should not be published to any publicly accessible repository. This is unfortunate because Web sites are often one of the first places that malicious entities will search for valuable information. For example, hackers often read the contents of a target organization's Web site to gather intelligence before any attacks [Sca01].

A public Web site should not contain the following information:

- Classified records
- Internal personnel rules and procedures
- Sensitive or proprietary information
- Personal information about organization's personnel[10]
  - Home addresses and phone numbers
  - Social Security Numbers (SSN)
- Investigative records
- Financial records (beyond those already publicly available)
- Organization's physical and information security procedures
- Information about organization's network and information system infrastructure
- Copyrighted material without the written permission of the owner
- Privacy or security policies that indicate the types of security measures in place

[10] For federal agencies, this would include all items covered by the Privacy Act of 1974 – (http://www.usdoj.gov/04foia/privstat.htm).

Never use a public Web server to host sensitive information intended to be accessed only by internal users (compromise of the public Web server will invariably lead to the compromise of this data).

To ensure a consistent approach, an organization should create a formal policy and process for determining and approving the information to be published on a Web server. In many organizations, this is the responsibility of the chief information officer (CIO) and/or public affairs officer. Such a process should include the following steps:

1. Identify information that should be published on the Web
2. Identify the target audience (why publish if no audience exists?)
3. Identify possible negative ramifications of publishing the information
4. Identify who should be responsible for creating and publishing this particular information
5. Create or format information for Web publishing
6. Review the information for sensitivity and distribution/release controls (including the sensitivity of the information in aggregate)
7. Determine the appropriate access and security controls
8. Publish information
9. Verify published information.

An area of Web content that is often overlooked is the information sometimes hidden within the source code of a Web page. This can be viewed from any Web browser through the use of the "view source code" menu option. Organizations often do not pay attention to the contents of the source code on their Web site, even though this code can contain sensitive information. The source code can, for example, contain points of contact and reveal portions of the directory structure of the Web server. Attackers will scour not only the obvious content of the Web site but also the hidden source code; thus, Web administrators or Webmasters should periodically review code on their public Web server.

5.2 Regulations Regarding the Collection of Personal Information

Several federal and state laws and regulations exist regarding the collection of user information on publicly accessible government Web sites. In addition, many government agencies have privacy guidelines that address the type of information that could be collected about users. Any governmental organization with a Web site should familiarize itself with the appropriate laws, regulations, and agency guidelines that are applicable to its organization. Private organizations may wish to use these as guidelines and examples of sound security practices but should consult appropriate legal counsel and their privacy officials for the applicable legal and policy implications. However, federal laws, regulations, and applicable agency guidelines may apply to commercial organizations that operate Web sites on behalf of federal agencies. Again, the ever changing legal, regulatory, and contractual issues call for careful consultation and advice from knowledgeable legal and policy experts.

Most federal agencies are prohibited from collecting personally identifying information on publicly accessible Web sites without the explicit permission of the user. This information includes the following:

- Name
- Internet Protocol (IP) address
- E-mail address
- Mailing address
- Telephone number
- SSN
- Financial information.

Federal agencies and many state agencies are also restricted in their ability to use Web browser "cookies" [OMB00a, OMB00b, OMB00c, and MASS99]. A cookie is a small piece of information that may be written to the user's hard drive when he or she visits a Web site. These files can be used to track users and gather a variety of information. There are two principal types of cookies.

Those that cause the most concern are called "persistent" cookies. These cookies can be used to track activities of users over time and across different Web sites. The most common use of persistent cookies is to retain and correlate information about users between sessions. Federal agencies and many state agencies are generally prohibited from using persistent cookies on publicly accessible Web sites.

"Session" cookies, which as their name implies, are cookies that are only valid for a single session (visit) to a Web site. These cookies expire at the end of the session or within a limited time frame. Because these cookies cannot be used to track personal information, they are generally not subject to the prohibition that applies to persistent cookies. However, their use must be clearly stated and defined in the Web site's privacy statement.

5.3 Securing Active Content and Content Generation Technologies

In the beginning of the WWW, most sites presented static information—that is, no interactivity existed between the user and Web site beyond the user clicking on hyperlinks. This information was based on the American Standard Code of Information Interchange (ASCII), often called text-based documents. Soon, interactive elements were introduced that offered users new ways to interact with the Web site. Unfortunately, these interactive elements because they can accept input and perform actions, introduced a raft of new Web related vulnerabilities.[11]

[11] For more extensive guidelines on active content, please see NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code* (http://csrc.nist.gov/publications/nistpubs/index.html).

Active content refers to interactive elements processed at the client (Web browser). If not implemented correctly it can present a serious threat to the end user. For example, active content can take actions without the express permission of the user. A variety of active content technologies exist. Some of the more popular examples include: ActiveX, Java, VBScript, and JavaScript. Organizations considering the deployment of client side active content should carefully consider the risks to their users, as the use of active content often requires the user to reduce the security settings on their Web browser.

Content generators are implemented on the server and thus represent a threat to the Web server itself. The danger in content generators is that they may accept input from users and can take actions on the Web server. If the content generator has not been programmed correctly, an attacker can enter certain types of information that may negatively impact the Web server or compromise its security. For example, one common attack against content generators is a buffer overflow. In this type of attack, a hacker will send large amounts of information to the content generator. The large amount of information will overflow the memory allocated to the content generator and, if formatted appropriately, this information overflow can be used to execute commands or gain unauthorized access to the Web server.

All Web sites that implement active content and content generators should perform additional steps to protect the active content from compromise. These steps, which are discussed in the following sections, may not apply to all installations; therefore, they should be used as guidance in conjunction with appropriate vendor documentation.

Special caution is also required for downloading preprogrammed scripts or executables from the Internet. Many Web administrators and Webmasters are tempted to save time by downloading freely available code from the Internet. Although this is obviously convenient, it is not risk free. There are many examples of malicious code being distributed this way. In general, no third-party scripts should be installed on a Web server until subjected to a thorough code review by a trusted expert.

5.3.1 Client Side Active Content Technologies and Related Vulnerabilities

A wide variety of client-side (Web browser) active content technologies is available. Each technology has its own strengths and weaknesses, and none is perfectly secure. Some of the most popular active content technologies and their associated risks are discussed below. New technologies are being released all the time. Any Web administrator or Webmaster, who is considering deploying a Web site with features that require active content technology at the client side, should carefully weigh the risks and benefits of the technology before implementation. In particular, Web administrators and Webmasters need to be cognizant of the fact that even if their content does not present a threat to the user, active content from other sites may present a threat, and it is unlikely that the user will remember to secure the browser settings when required.

PostScript – is one of the earliest examples of active content still in use today. PostScript is a powerful page description language from Adobe that uses language statements in text files that are translated by the PostScript interpreter to accurately display a page on any host that supports PostScript. This powerful language can be used maliciously to execute commands on the host interpreting the PostScript document. Unfortunately, the best protection against this type of attack is to disable certain commands within the PostScript interpreter, which can negatively affect its overall functionality [NIST01a].

Portable Document Format (PDF) – is a page description language from Adobe for specifying the appearance of the page containing text, graphics, and images, using the same high-level, device-independent image model employed by PostScript. This format is eventually created and read by Adobe Acrobat. Although less susceptible than some other types of active content, a number of vulnerabilities are associated with the PDF and the applications that support it. PDF files can be used to deploy malicious code, and certain versions of the commonly used Adobe Acrobat reader application are susceptible to buffer overflow vulnerabilities that can be used to crash and execute code on client hosts [NIST01a].

Java – is a full-featured, object-oriented programming language compiled into platform independent byte code executed by an interpreter called the Java Virtual Machine (JVM). The resulting byte code can be executed where compiled or transferred to another Java-enabled platform (e.g., conveyed via an HTML Web page as an applet). Java is useful for adding functionality to Web sites. Many services offered by various popular Web sites require the user to have a Java-enabled browser. When the Web browser sees references to Java code, it loads the code and then processes it using the built-in JVM.

The developers of Java tried to address the problem of security and were mostly successful. The Java programming language and runtime environment enforces security primarily through strong safety, by which a program can perform certain operations only on certain kinds of objects. Java follows a so-called sandbox security model, used to isolate memory and method access and to maintain mutually exclusive execution domains. Java code, such as a Web applet, is confined to a sandbox, which is designed to prevent it from performing unauthorized operations, such as inspecting or changing files on a client file system and using network connections to circumvent file protections or user's expectations of privacy.

Hostile applets still pose security threats, even while executing within the sandbox. A hostile applet can consume or exploit system resources inappropriately, or can cause a user to perform an undesired or unwanted action. Examples of hostile applets exploits include DoS, mail forging, invasion of privacy (e.g., exporting of identity, e-mail address, and platform information), and installing backdoors to the system. Because the Java security model is rather complex, it can be difficult for a user to understand and manage, which can increase risk. Moreover, many implementation bugs have also been found, enabling the user to bypass security mechanisms [NIST01a].

JavaScript – is a general purpose, cross-platform scripting language, whose code can be embedded within standard Web pages to create interactive documents. The name JavaScript is a misnomer because the language has little relationship to Java technology and rose independently from it. Within the context of the Web browser, JavaScript is extremely powerful, allowing prepared scripts to perform essentially the same actions as those a user could take. Within that context, JavaScript lacks methods for directly accessing a client file system or for directly opening connections to other computers besides the host that provided the content source. Moreover, the browser normally confines a script's execution to the page with which it was downloaded [NIST01a].

Visual Basic Script (VBScript) – is a programming language developed by Microsoft for creating scripts that can be embedded in Web pages for viewing with the Internet Explorer browser. Netscape Navigator, however, does not support VBScript. Like JavaScript, VBScript is an interpreted language able to process client-side scripts. VBScript, which is a subset of the widely used Microsoft Visual Basic programming language, works with Microsoft ActiveX controls. The language is similar to JavaScript and poses similar risks.

JavaScript and VBScript have similar vulnerabilities. In theory, confining a scripting language to boundaries of a Web browser should provide a relatively secure environment. In practice, this has not been the case. Many browser-based attacks stem from the use of a scripting language in combination with a security vulnerability. The main sources of problems have been twofold: the prevalence of implementation flaws in the execution environment and the close binding of the browser to related functionality, such as an e-mail client. Past exploits include sending a user's URL history list to a remote site, and using the mail address of the user to forge e-mail. The increasing use of HTML and other markup languages as content for e-mail and in push technology services has opened new avenues for exploits through embedded scripts [NIST01a].

ActiveX – is a set of technologies from Microsoft that provide tools for linking desktop applications to the WWW. ActiveX controls are reusable component program objects that can be attached to e-mail or downloaded from a Web site. ActiveX controls also come preinstalled on Windows platforms. Web pages invoke ActiveX controls using a scripting language or with an HTML OBJECT tag.

The ActiveX security model is considerably different from the Java sandbox model. The Java model restricts the permissions of applets to a set of safe actions. ActiveX, on the other hand, places no restrictions on what a control can do. Instead, its author, under a technology scheme called Authenticode, digitally signs ActiveX controls. The digital signatures are verified using identity certificates issued by a trusted certificate authority to an ActiveX software publisher. The Authenticode process ensures that ActiveX controls cannot be distributed anonymously and that tampering with the controls can be detected. This certification process, however, does not ensure that a control will be well behaved. Thus, the ActiveX security model assigns the responsibility for the computer system's security to the user (which is its greatest weakness).

Before the browser downloads an unsigned ActiveX control, or a control whose corresponding publisher's certificate was issued by an unknown certifying authority, the browser presents a dialog box warning the user that this action may not be safe. Users can choose to abort the transfer, or may continue the transfer if they assume the source is trustworthy or they are willing to assume the risk. Many users, if not most, are likely unaware of the security implications of their decision, which may have serious repercussions. Even when users are well informed, attackers may trick them into approving the transfer. Because the security of ActiveX depends on the knowledge and awareness of the end-user, it can be a very risky [NIST01a].

Figure 5.1 shows the relative risk of ActiveX compared with other popular client side active content technologies [NIST01a].

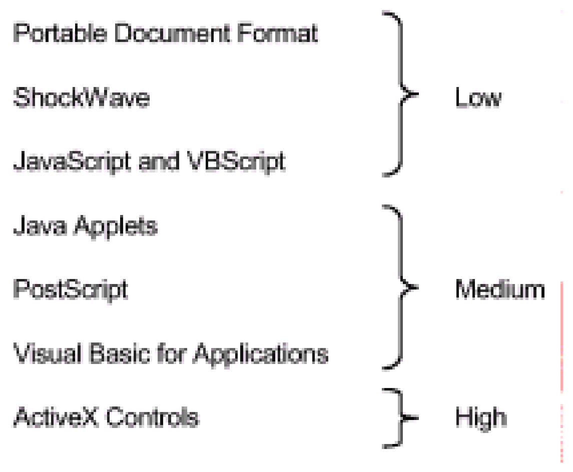


Figure 5.1: Relative Risk of Common Client Side Active Content

5.3.2 Server Side Content Generation Technologies and Related Vulnerabilities

Unlike the above technologies, CGI, ASP, PHP, and other similar server interfaces fall on the (Web) server side of the client-server model. The server side applications can be written in many programming languages to run on a Web server. If scripts are not prepared carefully, however, attackers can find and exercise flaws in the code to penetrate a Web server. Therefore, scripts must be written with security in mind and, for example, should not run arbitrary commands on a system or launch insecure programs. An attacker can find flaws through trial and error and does not necessarily need the source code for the script to uncover vulnerabilities [NIST01a].

Two general areas exist where server side content generator can create security vulnerabilities at the server:

- They may intentionally or unintentionally leak information about the host system that can aid an attacker, for example, by allowing access to information outside the areas designated for Web use.

- When processing user-provided input, such as the contents of a form, URL parameters, or a search query, they may be vulnerable to attacks whereby the user tricks the application into executing arbitrary commands supplied in the input stream.

Ideally, server-side scripts should constrain users to a small set of well-defined functionality and validate the size and values of input parameters so that an attacker cannot overrun memory boundaries or piggyback arbitrary commands for execution. If a script does contain flaws, it should be run only with minimal privileges (i.e., non-administrator) to avoid compromising the entire Web site. However, potential security holes can be exploited, even when applications run with low privilege settings. For example, a subverted script could have enough privileges to mail out the system password file, examine the network information maps, or launch a login to a high numbered port.

The two areas of vulnerability mentioned potentially affect all Web servers. Although these vulnerabilities have frequently occurred with CGI applications, other related interfaces and techniques for developing server applications have not been immune. CGI, being an early and well-supported standard, has simply gained more attention over the years, and the same areas of vulnerability exist when applying similar Web development technologies at the server.

Common Gateway Interface (CGI) – CGI scripts have been the initial mechanism used to make Web sites interact with databases and other applications. However, as the Web evolved, server-side processing methods have been developed that are more efficient and easier to program, for example, Microsoft ASPs for its IIS servers, Sun/Netscape supports Java servlets, and the freeware PHP is supported by on most major Web platforms, including Apache and IIS [NIST01a].

Server Side Includes (SSI) – SSI is a limited server-side scripting language supported by most Web servers. SSI provides a set of dynamic features, such as including the current time or the last modification date of the HTML file, as an alternative to using a CGI program to perform the function. When the browser requests a document with a special file type, such as ".shtml", it triggers the server to treat the document as a template, reading and parsing the entire document before sending the results back to the client (Web browser). SSI commands are embedded within HTML comments (e.g., <!--#include file="standard.html" -->). As the server reads the template file, it searches for HTML comments containing embedded SSI commands. When it finds one, the server replaces that part of the original HTML text with the output of the command. For example, the SSI command given above (i.e., #include file) replaces the entire SSI comment with the contents of another HTML file. This allows the display of a corporate logo or other static information prepared in another file to occur in a uniform way across all corporate Web pages. A subset of the directives available allows the server to execute arbitrary system commands and CGI scripts, which may produce unwanted side effects [NIST01a].

Microsoft Active Server Pages (ASP) – ASP is a server-side scripting technology from Microsoft similar to SSI, which can be used to create dynamic and interactive Web applications. An ASP page is essentially an HTML template that contains server-side scripts that run when a browser requests an ".asp" resource from the Web server. The Web server processes the requested page and executes any script commands encountered before sending the composed result to the user's browser. Both JScript and VBScript are supported scripting languages, but other scripting languages can also be accommodated, provided an ASP-compliant interpreter for that language is installed. For example, scripting engines are available for PERL, REXX, and Python languages from various sources. Scripting capabilities can be extended through the use of ActiveX objects, which can be developed in a variety of languages, including Visual Basic, C++, Cobol, and Java. A script that invokes an ActiveX object causes the object to be created and supplied any needed input parameters. Note that ActiveX (see Section 5.3.1) is an optional technology and not required by ASPs [NIST01a].

Java Servlets – Servlets are based on Java technology (see Section 5.3.1) and are a kind of server-side applet. The Web server first determines whether the browser's request requires dynamically generated information from a servlet. If so, the Web server can then locate or instantiate a servlet object corresponding to the request (e.g., by uploading the code from another server) and invoke it to obtain the needed results. The Web server typically populates itself with the servlet objects, which remain active until invoked. Thus, no startup overhead is associated with execution of the servlet objects. A Web server may also offload the handling of servlets to another server. By relying on Java portability and observing a common applications program interface, servlet objects can run in nearly any server environment. Servlets support an object-oriented environment on the Web server, which is flexible and extendible. Moreover, untrusted servlet objects can be executed in a secure area, with the dynamically generated information being passed from the secure area into the remaining server environment [NIST01a].

PHP (Hypertext Preprocessor) – PHP is a scripting language used to create dynamic Web pages. With syntax from C, Java, and Perl, PHP code is embedded within HTML pages for server side execution. PHP is commonly used to extract data from a database and present it on the Web page. Most major NT and UNIX Web servers support the language, and it is widely used with the mySQL database [NIST01a].

5.3.3 Server Side Content Generator Security Considerations

When examining or writing an active content executable or script, consider the following questions:

- How complex is the executable script? The longer it is, the more likely it is to have problems.
- Does it read or write files on the host system? Programs that read files may inadvertently violate access restrictions or pass sensitive system information. Programs that write files may modify or damage documents or introduce Trojan horses.
- Does it interact with other programs? For example, many CGI scripts send e-mail in response to form input by opening up a connection with the sendmail program. Is this performed in a secure manner?
- Does it run with suid (set-user-id) privileges? This is not recommended and should not be permitted if at all possible.
- Does the author validate user inputs from forms? This demonstrates that the author is thinking about security by preventing the possibility of buffer overflows.
- Does the author use explicit path names when invoking external programs? Relying on the PATH environment variable to resolve partial path names is not recommended.
- Has the Web server employing active content been scanned with a vulnerability scanner? Web servers (whether or not they employ active content) should be scanned periodically for vulnerabilities (see Section 8.4.1). There are several scanners that scan for content generator vulnerabilities (e.g., Whisker and Retina, see Appendix E).

When considering a server side content generator, it is important to review public vulnerability and security databases (such as the ICAT Metabase, http://icat.nist.gov) to determine the relative risk of various technologies under consideration. Although the historical record will not be a perfect indicator of future risk, it does indicate which technologies appear to be more vulnerable.

Various organizations research network and system security topics and periodically publish information concerning recently discovered vulnerabilities in service software. This includes Web server software and supporting technologies, such as scripting languages and external programs. External programs that are in wide-scale use are regularly analyzed by researchers, users, and security incident response teams and by members of the intruder community.

Intruders will often publish exploit scripts that take advantage of known vulnerabilities in Web service software and external programs commonly used by public Web servers. Web administrators should review public information sources frequently and be aware of all security-relevant information about any external programs that they are considering.

5.3.4 Location of Server Side Content Generators

The location of active content on the Web server is critical. If located in an incorrect directory or in a directory with the wrong permissions, it can quickly lead to the compromise of the Web server. To avoid this problem:

- Writeable files should be identified and placed in separate folders. No script files should exist in writeable folders. As an example, guest book data is usually saved in simple text files. These files need write permissions for guests to be able to submit their comments.
- Executable files (e.g., CGI, .EXE, .CMD, and PL) should be placed in separate folder(s). No other readable or writeable documents should be placed in these folders.
- Script files (e.g., ASP, PHP, and PL) should have separate folder(s).
- Include files (e.g., INC, SHTML, SHTM, and ASP) created for code reusability should be placed in separate directories. SSI should not generally be used on public Web servers. ASP include files should have an .asp extension instead of .inc.

5.3.5 Server Side Content Generator Security

The scripts themselves also should be programmed securely to avoid potential vulnerabilities. The following items should be considered when creating or analyzing a script for a security:

- For data entry forms, determine a list of expected characters and filter out unexpected characters from input data entered by a user before processing a form. For example, on most forms, expected data falls in these categories: letters a-z, A-Z, and 0-9. Device names such as AUX, COM, LPT, NUL, and PRN should also be filtered from the input data.
- Ensure that the dynamically generated pages do not contain dangerous metacharacters. It is possible for a malicious user to place these tags in a database or a file. When a dynamic page is generated using the tampered data, the malicious code embedded in the tags may be passed to the client browser. Then the user's browser can be tricked into running a program of the attacker's choice. This program will execute in the browser's security context for communicating with the legitimate Web server, not the browser's security context for communicating with the attacker. Thus, the program will execute in an inappropriate security context with inappropriate privileges.
- Character set encoding should be explicitly set in each page. Then the user data should be scanned for byte sequences that mean special characters for the given encoding scheme.
- Each character in a specified character set can be encoded using its numeric value. Encoding the output can be used as an alternate for filtering the data. Encoding becomes especially important when special characters, such as copyright symbol, can be part of the

dynamic data. However, encoding data can be resource intensive and a balance must be picked between encoding and other alternates methods for filtering the data.

- Cookies should be examined for any special characters. Any special characters should be filtered out.

- Employ an encryption mechanism to encrypt passwords entered through script forms (see Section 6.5).

- For Web applications that are restricted by username and password, none of the Web pages in the application should be accessible without going through the appropriate login process.

- Many Web servers and some other Web server software install sample scripts or executables during the installation process. Many of these have known vulnerabilities and should be removed immediately. See appropriate vendor documentation or Web site for more information.

5.3.6 Popular Server Side Content generator With Known Security Issues

The following commonly used scripts are known to have security vulnerabilities. If being used, these scripts should be updated or replaced immediately [WWW01]:

- **Matt Wright's TextCounter versions 1.0-1.2 (Perl) and 1.0-1.3 (C++)**. Earlier versions of the TextCounter program, which is used to place page hit counts on pages, fails to remove shell metacharacters from user-provided input. Consequently, remote users can execute shell commands on the server host, which affects the Perl and C++ versions. Upgrade to version 1.21 (Perl) or version 1.31 (C++).

- **Various guest book scripts**. Reports of exploits involving various guest book scripts continue to occur. These exploits take advantage of scripts that do not strip HTML tags from user-provided input and that write the guest book file to a directory that allows SSIs. Guest book scripts should strip HTML tags or replace angle brackets with the &gt; and &lt; and character entities. The files to which they write should not be in a directory that allows SSIs, ASPs, PHP pages, or other HTML template systems.

- **Excite Web Search (EWS) Engine, version 1.1**. The EWS engine stores critical security information (including the encrypted administrative password) in world writable files. This vulnerability allows unprivileged local users to gain access to the EWS administrative front end on both UNIX and NT systems. Note that this bug endangers a Web site only if its has the search engine installed locally. It does not affect sites that link to Excite.com's search pages, or sites that are indexed by the Excite search engine Web bot. A more serious problem is found in unpatched versions of EWS earlier than February 1998 (unfortunately, also called version 1.1). This bug involves the failure to check user-supplied parameters before passing them to the shell, allowing remote users to execute shell commands on the server host. The commands will be executed with the privileges of the Web server.

- **Info2www, versions 1.0-1.1**. Info2www, which converts GNU information files into Web pages, fails to check user-provided filenames before opening them. As a result, it can be tricked into opening system files or executing commands containing shell metacharacters. Versions 1.2 and higher are reported to be free of the problem; however, because of the many extant versions of this script, Web administrators should probably examine the source code before installing it. In addition, the CGI scripts info2html and infogate, which are also based on info2www should be scrutinized.

- **Count.cgi, versions 1.0-2.3**. Contains a stack overflow bug that allows malicious remote users to execute UNIX commands on the server by sending the script carefully crafted query strings. Version s 2.4 and later correct this vulnerability.

- **Microsoft FrontPage Extensions, versions 1.0-1.1**. Under certain circumstances, unauthorized users can vandalize authorized users' files by appending to them or overwriting them. The vulnerability is even more dangerous on a system with SSIs enabled because remote users may be able to exploit this bug to execute commands on the server.

5.4 Securing Web Content Checklist

Completed	Action
	Ensure that none of the following types of information are available on or via a public Web server
☐	Classified or sensitive records
☐	Internal personnel rules and procedures
☐	Confidential or proprietary information
☐	Investigative records
☐	Financial records (beyond those already publicly available)
☐	Organization’s physical and information security procedures
☐	Information about organization’s network and information system infrastructure
☐	Copyrighted material without the written permission of the owner
☐	Privacy or security policies that indicate the types of security measures in place
	Establish an organizational-wide documented formal policy and process for approving public Web content that
☐	Identifies information that should be published on the Web
☐	Identifies target audience
☐	Identifies possible negative ramifications of publishing the information
☐	Identifies who should be responsible for creating and publishing this particular information
☐	Provides guidelines on styles and formats appropriate for Web publishing
☐	Provides for appropriate review the information for sensitivity and distribution/release controls (including the sensitivity of the information in aggregate)
☐	Determines the appropriate access and security controls

Completed	Action
☐	Provides guidance on the information contained within the source code of the Web content
	Web user privacy considerations
☐	Published privacy policy
☐	Prohibition the collection of personally identifying data without the explicit permission of the user
☐	Prohibition on the use of "persistent" cookies
☐	Use of session cookie, if used, is clearly identified in published privacy policy
	Client side active content security considerations
☐	Used only when absolutely required
☐	No actions taken without express permissions of user
☐	No use of high risk client side active content
☐	When possible alternatives are provided (e.g., plain text provided along with PDF)
	Server side active content security considerations
☐	Simple easy to understand code
☐	Limited or no reading or writing of files
☐	Limited or no interaction with other programs (e.g., sendmail)
☐	No requirement to run with suid privileges
☐	Use of explicit path names (i.e., does not rely on path variable)
☐	No directories have both write and execute permissions
☐	All executable files are placed in a dedicated folders
☐	SSIs are disabled
☐	All user input is validated
☐	Dynamically created pages do not create dangerous metacharacters
☐	Character set encoding should be explicitly set in each page
☐	User data should be scanned for byte sequences that mean special characters for the given encoding scheme
☐	Cookies should be examined for any special characters
☐	Encryption mechanism is used to encrypt passwords entered through scripts forms
☐	For Web applications that are restricted by username and password, none of the Web pages in the application should be accessible without going through the appropriate login process
☐	All sample scripts are removed
☐	No third-party scripts or executable code are used without verifying the source code

6. Authentication and Encryption Technologies

Public Web servers often support a range of technologies for identifying and authenticating users with differing privileges for accessing information. Some of these technologies are based on cryptographic functions that can provide an encrypted channel between a Web browser client and a Web server that supports encryption.

Without user authentication, Web administrators will not be able to restrict access to specific information to authorized users. All information that resides on a public Web server will then be accessible by anyone with access to the server. In addition, users of the public Web server will not be able to determine if the server is the "authentic" Web server or a counterfeit version operated by a malicious entity.

Encryption can be used to protect information traversing the connection between a Web browser client and a public Web server. Without encryption, anyone with access to the network traffic can determine, and possibly alter, the content of sensitive information, even if the user accessing the information has been authenticated carefully. This may violate the confidentiality and integrity of critical information.

6.1 Determining Authentication and Encryption Requirements

A Web administrator should periodically examine all information accessible on the public Web server and determine the necessary security requirements. While doing so, the Web administrator should identify information that shares the same security and protection requirements. For sensitive information, the Web administrator should determine the users or user groups that have a legitimate need and should have access each set of resources.

For information that requires some level of user authentication, the Web administrator should determine which of the following technologies or methods will provide the appropriate level of authentication and encryption. Each has its own unique benefits and costs that should be weighed carefully with client and organizational requirements and policies. It may be desirable to use some authentication methods in combination.

6.2 Address-Based Authentication

The simplest authentication mechanism that is supported by most Web servers is address-based authentication. Access control is based on an IP address and/or host name of the host requesting information. Although easy to implement for small groups of users, address authentication can be unwieldy for Web sites that have a large potential user population (i.e., most public Web servers). It is also not very secure because it is susceptible to several types of attacks, including IP spoofing and Domain Name Service (DNS) spoofing. It should be used only in conjunction with stronger authentication methods as a second line of defense.

6.3 Basic Authentication

The basic authentication technology uses the Web server content's directory structure. Typically, all files in the same directory are configured with the same access privileges. A requesting user provides a recognized user identification and password for access to files in a given directory. More restrictive access control can be enforced at the level of a single file within a directory, given that the Web server software provides this capability. Each vendor's Web server software has its own method and syntax for defining and using this basic authentication mechanism.

From a security perspective, the main drawback of this technology is that all password information is transferred in an encoded, rather than an encrypted, form. Anyone who knows the standardized encoding scheme can decode the password after capturing it with a network sniffer. Furthermore, any Web content is transmitted as unencrypted plaintext, so this content also can be captured, violating confidentiality. Basic authentication is supported by standard-compliant Web browsers [CERT01]. Basic authentication is useful for protecting information from malicious bots (see Section 4.2.3).

6.4 Digest Authentication

Because of the drawbacks with basic authentication, an improved technique, known as digest authentication, was introduced in the publication of version 1.1 of the HTTP protocol. Digest authentication uses a challenge-response mechanism for user authentication. Under this approach a nonce or arbitrary value is sent to the user, who is prompted for an ID and password as with basic authentication. However, in this case, the information entered by the user is concatenated and a cryptographic hash of the result formed, which is again concatenated with the nonce and the requested URL and then rehashed as a response value that is sent to the user.

Because the user's password is not sent in the clear, it cannot be sniffed from the network. Moreover, the user's password is not needed by the server to authenticate the user, only the hashed value of the user ID and password, which provides further security. Because the nonce can be constructed from the current date and time information, replay attacks are also thwarted. Unfortunately, all other data is sent in the clear (i.e., unencrypted), and this is vulnerable to interception and alteration. Thus, digest authentication is only marginally more secure than basic authentication. Like basic authentication, digest authentication is useful for protecting information from malicious bots (see Section 4.2.3).

6.5 SSL/TLS

The SSL and Transport Layer Security (TLS) protocols provide server and client authentication and encryption of communications.[12] SSL was first introduced by Netscape Communications in 1994 and was revised twice (SSL version 3 is the latest version). In 1996, the Internet Engineering Task Force (IETF) established the TLS working group to formalize and advance the SSL protocol to the level of Internet standard. The TLS protocol version 1.0 is formally specified in the IETF RFC 2246,[13] which was published in 1999 based in large part on SSL version 3. For this document, SSL version 3 and TLS version 1 are essentially identical and will be discussed together. Most major Internet components, such as Web browsers, now support the use of either SSL or TLS.

[12] Proper understanding of SSL and the information presented in this section requires at least a basic understanding of cryptographic algorithms, message digest functions, digital signatures, symmetric encryption algorithms, and asymmetric encryption algorithms. For an introduction to cryptography, see NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure* (http://csrc.nist.gov/publications/nistpubs/index.html).

[13] http://www.ietf.org/rfc/rfc2246.txt

The Transmission Control Protocol/Internet Protocol (TCP/IP) governs the transport and routing of data over the Internet. Other protocols, such as the HTTP, Lightweight Directory Access Protocol (LDAP), or Internet Message Access Protocol (IMAP), run "on top of" TCP/IP in that they all use TCP/IP to support typical application tasks, such as displaying Web pages or running e-mail servers. Thus, SSL/TLS can support more than just secure Web communications. Figure 6.1 shows how SSL/TLS fits between the application and network/transport layers of the Internet protocol suite.

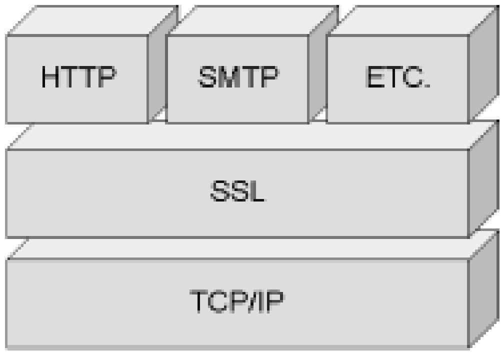


Figure 6.1: SSL/TLS Location within the Internet Protocol Stack

6.5.1 SSL/TLS Capabilities

SSL/TLS provides the following capabilities to HTTP and other application layer protocols [SSL98]:

- **Server Authentication**—SSL/TLS allows a Web client (user) to confirm a Web server's identity. SSL/TLS-enabled Web clients (e.g., Internet Explorer, Netscape, and Opera) can employ standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be important if the user, for example, is sending a credit card number over the network and wants to confirm the receiving server's identity.

- **Client Authentication**—SSL/TLS allows a Web server to confirm a user's identity. This is accomplished using the same techniques as those used for server authentication. SSL-enabled Web server software can confirm that a client's certificate and public ID are valid and have been issued by a CA listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank that is sending confidential financial information to a customer and wants to confirm the recipient's identity.

- **Communication Encryption**—SSL/TLS can encrypt most of the information being transmitted between a Web browser (client) and a Web server or even between two Web servers. This encryption provides a high degree of confidentiality. In addition, all data sent over an encrypted SSL connection is protected with a mechanism for detecting tampering—that is, for automatically determining whether the data has been altered in transit.

6.5.2 Weaknesses of SSL/TLS

Several limitations are inherent with SSL/TLS. Packets are encrypted at the TCP layer so IP layer information is not encrypted. Although this protects the Web data being transmitted, a person monitoring communications of an SSL/TLS session can determine both the sender and receiver via the unencrypted IP address information. In addition, SSL/TLS protects only data while it is being transmitted. It is not encrypted when stored at either end-point. Thus, the data is still vulnerable while in storage (e.g., a credit card database) unless additional safeguards are taken at the end-points.

SSL/TLS are also vulnerable to the "man in the middle" attack. This occurs when a malicious entity intercepts all communication between the Web client and the Web server with which the client is attempting to communicate via SSL/TLS. The attacker intercepts the legitimate keys that are passed back and forth during the SSL/TLS handshake (see Section 6.5.3), substitutes his or her own, and makes it appear to the Web client that he or she is the Web server and to the Web server that he or she is the Web client [SSL98].

The encrypted information exchanged at the beginning of the SSL/TLS handshake is actually encrypted with the malicious entity's public key or private key, rather than the Web client's or Web server's real keys. The attacker program ends up establishing one set of session keys for use with the real Web server, and a different set of session keys for use with the Web client. This allows the attacker program not only to read all the data that flows between the Web client and the real Web server, but also to change the data without being detected. Therefore, it is extremely important for the Web client to check that the domain name in the server certificate corresponds to the domain name of the server with which a client is attempting to communicate. Most modern browsers automatically do this, but users can override the warning and thus need to be educated of the dangers.

6.5.3 Example SSL/TLS Session

The SSL/TLS protocols use a combination of public-key and symmetric key encryption. Symmetric key encryption is much faster than public-key encryption, but public-key encryption provides better authentication techniques. An SSL/TLS session always begins with an exchange of messages called the SSL/TLS handshake. The handshake allows the server to authenticate itself to the client using public-key techniques; this allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server.

The exact programmatic details of the messages exchanged during the SSL/TLS handshake are beyond the scope of this document. However, the basic steps involved can be summarized as follows [SSL98]:

1. "The client sends the server the client's SSL/TLS version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL/TLS."

2. "The server sends the client the server's SSL/TLS version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL/TLS. The server also sends its own certificate and, if the client is

requesting a server resource that requires client authentication, requests the client's certificate."

3. "The client uses some of the information sent by the server to authenticate the server. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to Step 4."

4. "Using all data generated in the handshake to this point, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server's public key (obtained from the server's certificate, sent in Step 2), and sends the encrypted premaster secret to the server."

5. "If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case, the client sends both the signed data and the client's own certificate to the server, along with the encrypted premaster secret."

6. "If the server has requested client authentication, the server attempts to authenticate the client. If the client cannot be authenticated, the session is terminated. If the client can be successfully authenticated, the server uses its private key to decrypt the premaster secret, then performs a series of steps (which the client also performs, starting from the same premaster secret) to generate the master secret."

7. "Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL/TLS session and to verify its integrity—that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL/TLS connection."

8. "The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished."

9. "The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished."

10. "The SSL/TLS handshake is now complete, and the SSL/TLS session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity."

6.5.4 SSL/TLS Encryption Schemes

The SSL/TLS protocols support the use of a variety of different cryptographic algorithms for operations such as authenticating the Web server and Web client to each other, transmitting certificates, and establishing session keys. Web clients and Web servers may support different cipher suites, or sets of ciphers; depending on factors such as the version of SSL/TLS they support, organizational policies regarding acceptable encryption strength; and government restrictions on export, import, and use of SSL/TLS-enabled software. Among its other functions, the SSL/TLS handshake protocols determine how the Web server and Web client negotiate which cipher suites they will use to authenticate each other, to transmit certificates, and to establish session keys. Tables 6.1 and 6.2 provide a list of the more popular cipher suites, their recommended usage, and their relative strength [SSL98 and Cho02].

Table 6.1: SSL/TLS Encryption Schemes with Diffie-Hellman or RSA Key Exchange

Recommended Use	Cipher Suites
Banks and commercial or governmental organizations that handle sensitive data.	
Highest Security:	Encryption: Advanced Encryption Standard (AES) 256-bit encryption
	Authentication & Digest: Digital Signature Standard (DSS) or RSA with 1024-bit keys, and Secure Hash Algorithim-1 (SHA-1)
Security and Performance:	Encryption: AES 128-bit encryption
	Authentication & Digest: DSS or RSA with 1024-bit keys, and SHA-1
Security and Compatibility:	Encryption: Triple Data Encryption Standard (3DES) 168-bit encryption (note: 3DES is considerably slower than AES)
	Authentication & Digest: DSS or RSA with 1024-bit keys, and SHA-1
Situations where confidentiality of data (encryption) is not a concern, but where authentication or tamper detection is desired.	Authentication & Digest: DSS or RSA with 1024-bit keys and SHA-1

Choosing an appropriate encryption algorithm depends on several factors that will vary with organization. Although at first glance it might appear that the strongest encryption available should always be used, that is not always true. The higher the level of the encryption, the greater impact it will have on the Web server's resources and communications speed. Furthermore, a number of countries still maintain restrictions on the export, import, and/or use of encryption. Patents and licensing issues may affect which encryption schemes can be used in a particular country. Common factors that influence the choice of encryption algorithm are as follows:

- Required security
  - Value of the data (to either the organization and/or other entities—the more valuable the data, the stronger the required encryption)
  - Time value of data (if data are valuable but for only a short time period [e.g., days as opposed to years] then a weaker encryption algorithm can be used—for example, passwords that are changed daily because the encryption needs to protect the password for only a 24-hour period)
  - Threat to data (the higher the threat level, the stronger the required encryption)
  - Other protective measures (if other protective measures are in place, they may reduce the need for stronger encryption—an example would be using protected methods of communications such as dedicated circuits as opposed to the public Internet).

- Required performance (higher performance requirements may necessitate weaker encryption)
- System resources (less resources [e.g., process, memory] may necessitate weaker encryption)
- Import, export, or usage restrictions
- Encryption schemes supported by Web server application
- Encryption schemes supported by Web browsers of expected users.

6.5.5 Implementing SSL/TLS

To implement SSL/TLS on a Web server, it will need a certificate (a.k.a., digital certificate). A certificate which is the digital equivalent of an ID card, is used in conjunction with a public key encryption system. Certificates can be issued by trusted third parties, known as Certificate Authorities (CA) or can be "self-signed." Which one is superior depends on an organization's requirements.

Although the sequence of steps is not identical for all Web servers, the implementation of a third-party signed certificate for a Web server generally includes at least three steps:

- Generating and submitting a certificate-signing request (CSR)
- Picking up a signed SSL/TLS certificate from a CA
- Installing the certificate and configuring the Web server to use SSL/TLS for any specified resources.

A CSR consists of three parts:

- Certification request information
- Signature algorithm identifier
- Digital signature over the certification request information.

Although the specific steps to generate a CSR may differ somewhat for each Web server, Figure 6.2 shows a sample CSR.

```
-----BEGIN CERTIFICATE REQUEST-----
AQAwejELMAkGA1UEBhMCQ0ExEzARBgNVBAgTClRFc3QgU3RhdGUxETA
vbG9yYWR0MRswGQYDVQQKExJDYW5hZGlhbiBUZXN0IE9yZy4xEjAQBg
9mZmljZTESMBAGA1UEAxMJd3d3LmV4LmNhMIGfMA0GCSqGSIb3DQEBA
QKBgQD5PIij2FNa+Zfk1OHtptspcSBkfkfZ3jFxYA6ypo3+YbQhO3PL
WyvoNvL8Gnp1GUPgiw9GvRao603yHebgc2bioAKoTkWTmW+C8+Ka42w
DnDWOSBWWR1L1j1YkQBK1nQnQzV3U/h0mr+ASE/nV7wIDAQABoAAwDQ
EEBQADgYEAAAhxY1dcw6P8cDEDG4UiwB0DOoQnFb3WYVl7d4+6lfOtK
QoVpOICF3gfAF6wcAbeg5MtiWwTwvXRtJ2jszsZbpOuIt0WU1+cCYiv
rD4s2ZJytkzDTAcz1Nmiuh93eqYw+kydUyRYlOMEIomNFIQ=
-----END CERTIFICATE REQUEST-----
```

Figure 6.2: Sample CSR

Web servers that are SSL/TLS enabled provide specific instructions for the generation of a CSR.[14] There are two major types of CSRs. The most popular is the encoded Public Key Cryptography Standard (PKCS) #10, Certification Request Syntax Standard, which is used by newer Web servers [RSA00]. The other CSR type, based on the Privacy Enhanced Mail (PEM) specification, is called either PEM Message Header or Web site Professional format. The use of this CSR is generally limited to older Web servers.

Many of the more recent Web servers generate PKCS #10 compliant CSRs similar to the example CSR shown previously. A CSR provides not only additional information about a given entity, or a "challenge password" by which the entity may later request certificate revocation but also attributes for inclusion in X.509 certificates [RSA00].

When supplying any necessary information during the CSR generation process, it is important to check spelling and punctuation. The URL that is supplied must exactly match the URL for which the certificate is used, or SSL/TLS clients are configured to generate an error. In some instances, a user may acknowledge this error in an alert box and proceed ahead despite it.

Once the CSR has been generated, it must be submitted to a CA. The CA's role is to fulfill the CSR by authenticating the requesting entity and verifying the entity's signature. If the request is valid, the CA constructs an X.509 certificate from the DN and public key, the issuer name (or more commonly referred to as the common name [CN]), and the CA's choice of serial number, validity period, and signature algorithm.

Upon receiving a submitted CSR, the CA must verify the CSR and create a signed X.509 certificate. At this point, most CAs will then alert the applicant by phone, e-mail, etc., that the X.509 certificate is available. Once notified, applicants will be able to download their certificates, through an SSL/TLS-protected Web-based interface. Figure 6.3 shows what a certificate looks like unencoded. Similar to the CSR, when supplying a certificate to a configuration wizard or even saving it to hard drive, the lines "BEGIN CERTIFICATE" and "END CERTIFICATE" are vital. Without them, the Web server application will be unable to interpret the encoded contents of the certificate.

[14] For CSR generation methods of Web servers, see: http://www.thawte.com/getinfo/products/keygen/contents.html.

```
-----BEGIN CERTIFICATE-----
AwIBAgIBAzANBgkqhkiG9w0BAQQFADCBzzELMAkGA1UEBhMCQ0ExEDAOBg
FyaW8xETAPBgNVBAcTCFdhdGVybG9vMR8wHQYDVQQKExZVbml2ZXJzaXR5
bG9vMSswKQYDVQQLEyJJbmZvcm1hdGlvbiBTeXN0ZW1zIGFuZCBUZWNobm
YDVQQDExxVVy9JU1QgQ2VydGlmaWNhdGUgQXV0aG9yaXR5MSYwJAYJKoZI
c3QtY2FAaXN0LnV3YXRlcmxvby5jYTAeFw05ODA4MjcxNjE0NDZaFw05OT
ZaMIHGMQswCQYDVQQGEwJDQTEQMA4GA1UECBMHT250YXJpbzERMA8GA1UE
b28xHzAdBgNVBAoTFlVuaXZlcnNpdHkgb2YgV2F0ZXJsb28xKzApBgNVBA
F0aW9uIFN5c3RlbXMgYW5kIFRlY2hub2xvZ3kxGTAXBgNVBAMTEGlzdC51
Y2ExKTAnBgkqhkiG9w0BCQEWGndlYm1hc3RlckBpc3QudXdhdGVybG9vLm
qGSIb3DQEBAQUAA4GNADCBiQKBgQCw8Sc7X4EeAxBxTPgmTd4Utau0BIqY
n2A7G5MtkMHj0triXoineuRxW9MQSQW8jMAv+xznMaL6OxnG+txyBjYx1z
81kgbypp5Usf18BonsqSe9Sl2P0opCCyclGr+i4agSP5RM5KrycTSVoKHE
MH4wOgYJYIZIAYb4QgEEBC0WK2h0dHA6Ly9pc3QudXdhdGVybG9vLmNhL3
NhLWNybC5wZW0wLQYJYIZIAYb4QgENBCAWHklzc3VpbmcgQ0EgYXNzdW1l
bGl0eTARBglghkgBhvhCAQEEBAMCAEAwDQYJKoZIhvcNAQEEBQADgYEADZ
IMOSbqTQK1LUjn4uHN3BLmqxznIzdiMu4RXyxne5Uq9EA7LbttutH7fIoO
FoU1dtEvovXmA6m5G+SN8A9tIAvRGjNmphB82xGkwEXuLN0afYz5XaFo3Z
hPTgNIyYEiiSp6Qfc=
-----END CERTIFICATE-----
```

Figure 6.3: Sample Encoded SSL/TLS Certificate

Whatever format the SSL/TLS certificate is delivered in, administrators should take extreme caution in securing their certificate and encryption keys. The following are tips for security of the certificate:

- Create and store a backup copy of the certificate on read-only media in case the original certificate is deleted accidentally. If the certificate or key is lost and cannot be recovered from backup media, a new key and certificate must be created.

- Store the original certificate in a folder or partition accessible by only Web or system administrators and secured by appropriate authentication mechanisms.

- Consider running data integrity scanner (e.g., Tripwire) on the Web server (see Section 7.2.2) and ensure that it is monitoring for any changes to the certificate.

- Examine system logs regularly to validate and ensure prevention of unauthorized system access.

If a malicious user gains unauthorized access to a Web server, the integrity of the entire server is lost immediately once the encryption key pair is modified. Once a key in an SSL/TLS certificate is compromised, it can remain compromised because no mechanism exists for consulting the root of a CA to confirm the key in question has not been revoked.

For many organizations, a certificate issued by a third-party CA is not required. In those instances, the organization may wish to "self-sign" their Web server certificate. Although this will not provide users the same verification provided by a third-party CA, it avoids the cost of purchasing and renewing a certificate.

A self-signed certificate uses its own certificate request as a signature rather than the signature of a third-party CA. Two limitations with self-signed certificates must be considered:

- Browsers will not automatically recognize the certificate and allow a secure connection to be made, without first prompting the user. Organizations can configure the Web browsers used by their employees to recognize the self-signed certificate, but the public at large will still get the notification.

- When CAs issue a signed certificate, they are guaranteeing the identity of the organization and the Web server that is providing the Web pages to the browser. Thus, use of a third-party CA will often be required for sensitive transactions with the public at large, such as e-commerce and e-government applications.

Although the sequence of steps is not identical for all Web servers, the implementation of a third-party signed certificate for a Web server includes at least two steps:

- Generating a public and private key pair

- Creating a self-signed certificate.

Once a certificate has been collected from the CA or self generated it will be necessary to enable and configure SSL. Some steps are common to all Web servers:

- Indicate location of SSL/TLS certificate/instruct server to start using SSL/TLS. In certain cases, the Web server must be instructed to begin using SSL/TLS, and even to the exact location of the SSL/TLS certificate and private keys if they were stored as files on the hard drive.

- Instruct server to listen to TCP port 443. This is the default TCP port from which SSL/TLS resources are accessed by clients (other ports can be used). In most cases, if the server was not previously using SSL/TLS, this port would be disabled for security reasons. It will probably be necessary to configure any network infrastructure supporting the Web server to allow SSL/TLS traffic (see Section 7.2).

- Configure the server to protect the necessary resources (directories and/or files) using SSL/TLS. Configure the Web server application so that the appropriate resources are protected with SSL/TLS. These resources are then accessible only from a URL that starts with https://.

Newer versions of the HTML standard have even been amended to include a response to inform clients when they requested a file that is available only via SSL/TLS or vice versa. The HTTP status code 403.4 indicates that a HTTP GET request must be prefixed with an https:// because the resource requested is protected with SSL/TLS. For more information, consult the HTTP RFC 1945.[15]

6.5.6 SSL/TLS Implementations

Although some Web servers come packaged with SSL capabilities already integrated, many do not. This section discusses various commercial and open-source SSL/TLS implementations, in use today. Some of these packages contain the functionality to generate SSL certificates without the need of a CA. The following list illustrates some of the SSL toolkits available:

[15] http://www.ietf.org/rfc/rfc1945.txt

- SSLRef 3.0 is the sample (or "reference") implementation from Netscape (http://www.netscape.com/newsref/std/sslref.html).
- SSLava is an implementation of SSL written in Java from Phaos Technology (http://www.phaos.com/solutions.html).
- SSLeay is a free noncommercial implementation of SSL 2.0 written by Eric Young. It includes cryptographic code, which because of patent issues can be used royalty-free only outside of the United States (ftp://ftp.psy.uq.oz.au/pub/Crypto/SSL/).
- OpenSSL is an open source implementation of SSL/TLS for UNIX and Linux platforms (http://www.openssl.org).

6.6 Web Authentication and Encryption Technologies Checklist

Completed	Action
	Web authentication and encryption technologies
☐	For Web resources that require minimal protection and for which there is a small, clearly defined audience, configure address-based authentication
☐	For Web resources that require additional protection but which for which there is a small, clearly defined audience, configure address-based authentication as a second line of defense
☐	For Web resources that require minimal protection but for which there is no clearly defined audience, configure basic or digest authentication (better)
☐	For Web resources that require protection from malicious bots (see Section 4.2.3), configure basic or digest authentication (better)
☐	For Web resources that require maximum protection, configure SSL/TLS
	Configuring SSL/TLS
☐	For configurations that require minimal authentication but require encryption, use self-signed certificate
☐	For configurations that require server authentication and encryption, use third-party issued certificate
☐	For configurations that require a medium level of client authentication, configure server to require username and password via SSL/TLS
☐	For configurations that require a high level of client authentication configure server to require client certificates via SSL/TLS
☐	For government and commercial organizations that require a medium level of encryption, use DES
☐	For government and commercial organizations that require a high level of encryption, use RC4 (high) or 3DES (highest)
☐	For government organizations that require a highest level of encryption, use Fortezza
☐	Configure file integrity checker to monitor Web server certificate

Completed	Action
☐	If only SSL /TLS is to be used on the Web server, ensure access via TCP port 80 is disabled
☐	If most traffic to the Web server will be via encrypted SSL/TLS, ensure that appropriate logging and detection mechanisms are employed on the Web server (because network monitoring is ineffective against encrypted SSL/TLS sessions)

7. Implementing a Secure Network for a Web Server

The network infrastructure that supports the Web server plays a critical role in the security of the Web server. In most configurations, the network infrastructure will be the first line of defense between the Internet and a public Web server. Although considerations of network infrastructure are influenced by many factors other than security (e.g., cost, performance, and reliability), this section will primarily address security issues.

Network design alone, however, cannot protect a Web server. The frequency, sophistication, and even variety of Web attacks perpetrated today, support the idea that Web security must be implemented through layered and diverse defense mechanisms (defense in depth). This section discusses those network components that can support and protect Web servers to further enhance their overall security.

7.1 Network Location

An organization has many choices when selecting a networking location, and security may not be the principal factor in deciding between those options. Network location is the first and in many respects most critical networking decision that affects Web server security. Network location is important for several reasons. Network location determines what network infrastructure can be used to protect the Web server. For example, if the Web server is located behind the organization's firewall, then the firewall cannot be used to control traffic to and from the Web server. Network location also determines what other portions of the network are vulnerable if the Web server is compromised. For example, if the Web server is located on the internal production network, then the internal network is subject to attack from the compromised Web server. An organization may choose not to have the Web server located on its network at all and to outsource the hosting to a third-party.

7.1.1 Internal Network

Some organizations choose to locate their public Web servers on their internal production networks, that is, they locate their Web server on the same network as their internal users and servers. This location is not recommended because it exposes the internal network to unnecessary risk of compromise. The principal weakness of this configuration is that Web servers are often the target of choice for an attacker. If they manage to compromise the Web server, they will be on the internal network and can more easily compromise internal hosts. Figure 7.1 shows an example of this time of network configuration.

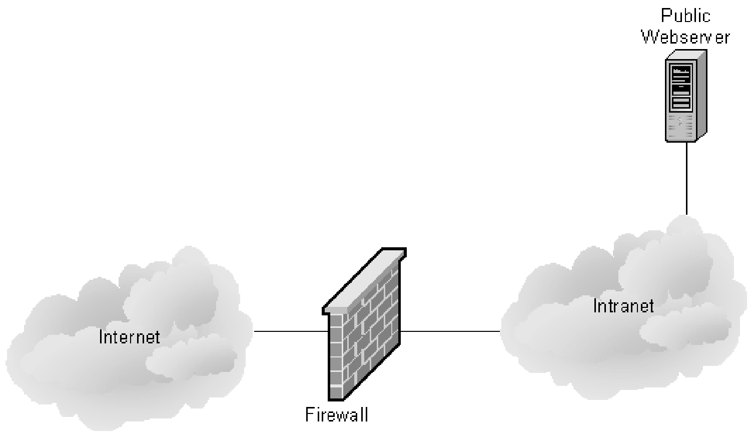


Figure 7.1: Public Web Server on Internal Network

The advantages of locating the Web server on the internal network from a security standpoint are as follows:

- Web server can be protected by a firewall.
- Easy to administer the Web server and update Web server content.

The disadvantages of locating the Web server on the internal network from a security standpoint are as follows:

- Compromise of the Web server directly threatens the internal network.
- DoS attacks aimed at the Web server will affect the internal network.

Locating the Web server on the internal network is extremely risky and should not be done except in exceptional circumstances and only with a full understanding of the risks.

7.1.2 External to Firewall

Another network location that is not generally recommended is placing the Web server before an organization's firewall or router that provides IP filtering. In this type of the configuration the network can provide little, if any, protection to the Web server. All security has to be provided by the Web server itself, which provides a single point of failure. To be even somewhat secure in this location, the Web server operating system and application has to be well hardened (all unnecessary and insecure services disabled) and with all necessary security patches applied. To maintain the "security" of the setup, the Web administrator must stay up-to-date on all vulnerabilities and related patches. Another limitation of this location is that it is difficult in this type of configuration to provide any sort of secure remote administration or content update capability. This type of network configuration is demonstrated in Figure 7.2.

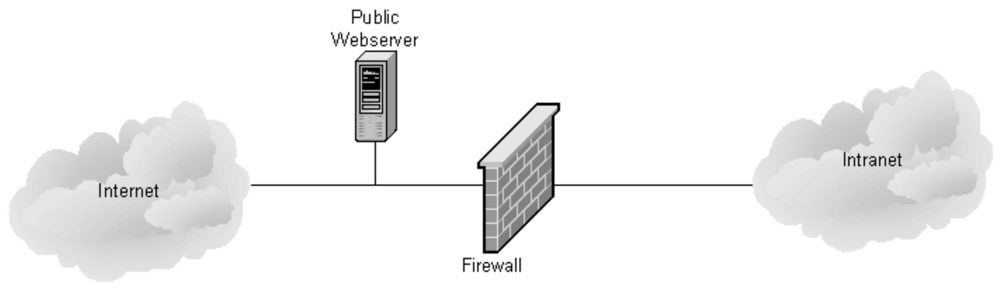


Figure 7.2: Web Server External to Firewall

The advantages of locating the Web server external to the firewall from a security standpoint are as follows:

- Compromise of the Web server does not directly threaten the internal network.
- DoS attacks aimed at the Web server will not (generally) affect the internal network.

The disadvantages of locating the Web server external to the firewall from a security standpoint are as follows:

- Web server cannot be protected by a firewall.
- Web server and operating system configuration has to be perfect to be (relatively) secure.
- Difficult to securely administer the Web server and update its content from the intranet.
- Difficult to monitor network traffic to and from the Web server.

Locating the Web server external to the firewall is extremely risky and should not be done except in exceptional circumstances and only with a full understanding of the risks.

7.1.3 Demilitarized Zone

A Demilitarized Zone (DMZ) can be defined as a host or network segment inserted as a "neutral zone" between an organization's private network and the Internet. It prevents outside users of the Web server from gaining direct access to an organization's internal network (intranet). A DMZ mitigates the risks of locating a Web server on an internal network or exposing it directly to the Internet. It is a compromise solution that offers the most benefits with the least amount of risk for most organizations. The DMZ allows access to the resources located within it to both internal and external users.

In creating a DMZ, an organization will place a firewall between its border router and its internal network (in fact the border router itself may act as the firewall). The new segment of network that is created by this action is where a Web server is placed. Figure 7.3 illustrates an example of a DMZ.

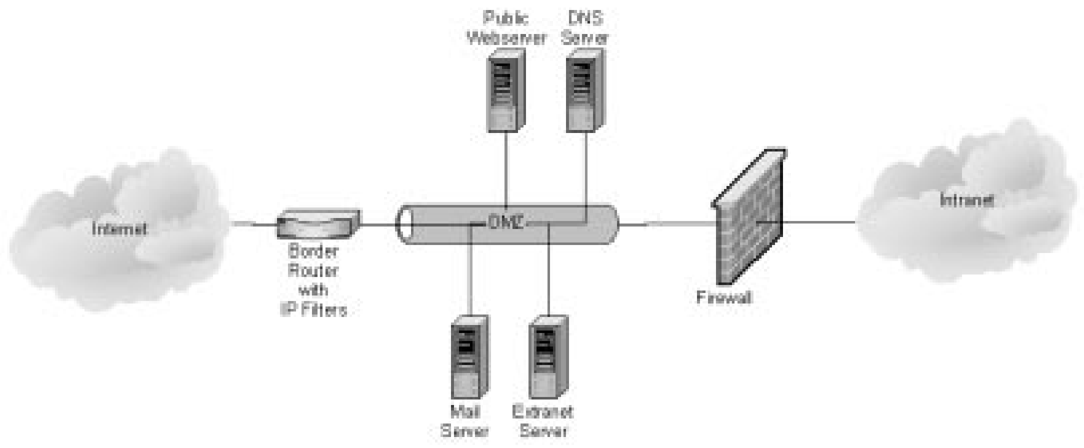


Figure 7.3: Simple Single Firewall DMZ

One type of DMZ network configuration that is not recommended for use with Web servers is the so-called "service leg" DMZ. In this configuration, a firewall is constructed with network interfaces. One network interface attaches to the border router, another interface attaches to the internal network, and a third network interface connects to the DMZ (see Figure 7.4).

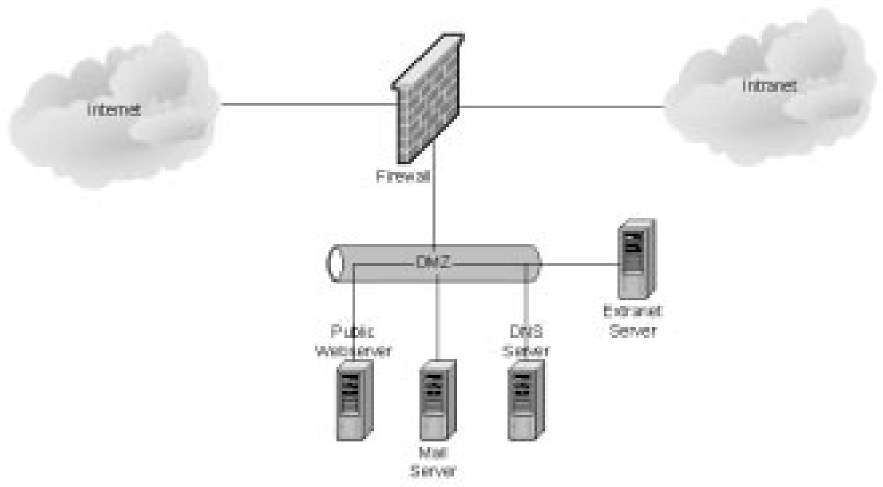


Figure 7.4: Three Interface Firewall DMZ

This configuration subjects the firewall to an increased risk of service degradation during a DoS attack aimed at the Web server. In a standard DMZ network configuration (discussed above), a DoS attack against the Web server will generally only affect the Web server. In a service-leg DMZ network configuration, the firewall bears the brunt of any DoS attack because it must examine any network traffic before the traffic reaches the Web server (or any other DMZ or internal network resource). This processing can overwhelm the firewall and slow all traffic, including that destined for the internal network [NIST02].

The advantages of a DMZ from a security standpoint are as follows:

- Web server can be well protected and network traffic to and from the Web server can be monitored.
- Compromise of the Web server does not directly threaten internal production network.

- Greater control over the security of the Web server.
- Less difficult to securely administer the Web server and update its content from the intranet.
- DMZ network configuration can be optimized to support and protect the Web server(s).

The disadvantages of a DMZ from a security standpoint are as follows:

- DoS attacks aimed at the Web server may have an effect on the internal network.
- Depending on the traffic allowed to and from the DMZ and internal network, it is possible that the Web server can be used to attack or compromise hosts on the internal network.

For organizations that support their own Web server, a DMZ is almost invariably the best option. It offers protection for the Web server and other externally accessible servers without exposing the internal network.

7.1.4 Outsourced Hosting

Many organizations choose to outsource the hosting of their Web server to a third-party (e.g., an Internet Service Provider [ISP], Web hosting service, or other government agency). In this case, the Web server would not be located on the organization’s network. The hosting service network would have a dedicated network that hosts many Web servers (for many organizations) operating on a single network (see Figure 7.5).

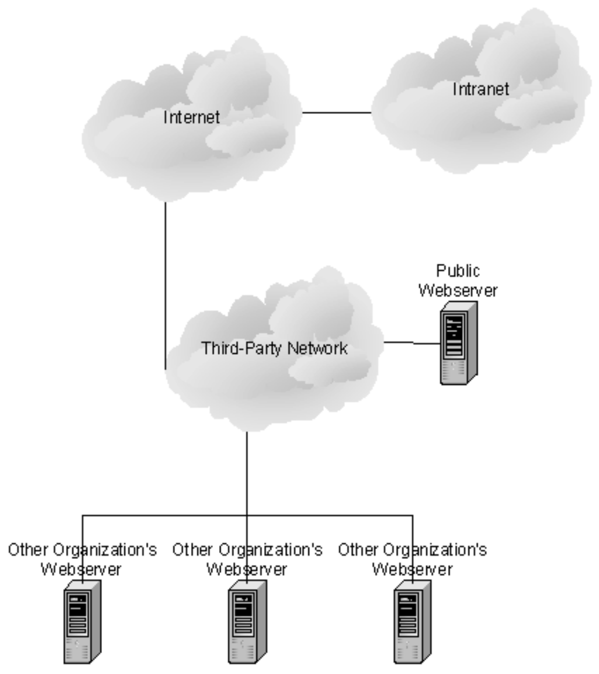

Internet
Intranet
Public Webserver
Third-Party Network
Other Organization's Webserver
Other Organization's Webserver
Other Organization's Webserver

Figure 7.5: Outsourced Web Server Hosting

The advantages of outsourcing from a security standpoint are as follows:

- DoS attacks aimed at the Web server have no effect on the organization's production network.
- Compromise of the Web server does not directly threaten internal production network.
- Outsourcer may have greater knowledge in securing and protecting Web servers.
- Network optimized solely for the support and protection of Web servers.

The disadvantages of outsourcing from a security standpoint are as follows:

- Difficult or impossible to remotely administer the Web server or remotely update Web server content.
- Less control over the security of the Web server.
- Web server may be affected by attacks aimed at other Web servers hosted by the outsourcer on the same network.

Outsourcing often makes sense for smaller organizations that cannot afford the necessary expertise to support the necessary Web server staff. It may also be appropriate for larger organizations that do not wish to host their own Web server for whatever reason. It usually does not make sense for organizations that wish to maintain tight control over the Web server.

7.2 Network Element Configuration

Once the Web server has been located in the network, it will be necessary to configure the network infrastructure elements to support and protect the Web server. The elements of network infrastructure that affect Web server security include firewalls, routers, intrusion detection systems, and network switches. Each has an important role to play and is critical to the overall strategy of protecting the Web server through defense in depth. Unfortunately, when it comes to securing a Web server there is no single "silver bullet" solution. A firewall or Intrusion Detection System (IDS) alone cannot adequately protect a public Web server from all threats or attacks.

7.2.1 Firewall Configuration

Firewalls are devices or systems that control the flow of network traffic between networks. They protect Web servers from vulnerabilities inherent in the TCP/IP suite. They also help reduce the security issues associated with insecure applications and operating systems. There are several types of firewalls. These firewalls include border routers, which can provide access control on IP packets; to stateful firewalls, which can also control access based not only on IP but also TCP and User Datagram Protocol (UDP) protocols; and the most powerful firewalls, which can understand and filter Web content.[16]

[16] For more information of Firewalls, see NIST Special Publication 800-41, *Guide to Firewall Selection and Policy Recommendations* (http://csrc.nist.gov/publications/nistpubs/index.html).

A common misperception of firewalls is that they eliminate all risk and can protect against the misconfiguration of the Web server or poor network design. Unfortunately, this is not the case. Firewalls themselves are vulnerable to misconfiguration and (sometimes) software vulnerabilities. Web servers in particular are vulnerable to many attacks, even when located behind a secure, well-configured firewall. For example, the recommended configuration for a firewall that is protecting a Web server is to block all access to the Web server from the Internet except for HTTP (TCP port 80) and/or HTTPS (TCP port 443). Even with this configuration, many Web server applications are vulnerable to attack via TCP port 80. Thus, a firewall is the critical first line of defense for a Web server. However, to be truly secure, the organization will need to practice defense in depth for its Web server (and network). Most importantly, organizations should strive to maintain all systems in a secure manner and not depend solely on the firewall(s) (or any single component) to stop attackers.

The most basic type of firewall is a network layer (also called packet filter) firewall, which is usually a router with access control lists (network administrator configured security rules) installed. A network layer firewall can provide filtering based on several pieces of information [NIST02]:

- Source IP address
- Destination IP address
- Traffic type
- TCP/UDP port number (sometimes).

The strengths of network layer firewalls are as follows:

- Speed
- Cost (most organizations already have a border router than can be configured to provide network layer firewall capabilities)
- Mature, secure technology.

The weakness of network layer firewalls are as follows:

- Susceptible to application layer attacks (e.g., cannot examine Web content)
- Difficult to configure
- Limited logging capabilities
- Susceptible to IP spoofing attacks
- Limited rule set and filtering capabilities.

Network layer firewalls are useful as the first line of defense but should not be used as an organization's only firewall. The only possible exception to this rule is for Web servers that have had the operating system and application locked down and hardened and which face a small threat level. Few enterprise-level firewalls are pure network layer firewalls. About the only pure network layer firewalls that are available today are small office home office (SOHO) firewall appliances and personal firewalls [NIST02].

Stateful inspection firewalls are network layer firewalls that incorporate additional "awareness" of TCP protocol. Stateful inspection firewalls maintain internal information about the state of connections passing through them, the contents of some of the data streams, and so on. This allows for better and more accurate rules sets and filtering. Stateful inspection firewalls have the same strengths and weaknesses as network layer firewalls, except that they are somewhat more secure. Most enterprise level network layer firewalls are stateful inspection firewalls [NIST02]. These firewalls may be appropriate for Web servers that are appropriately hardened and that require the throughput inherent in these firewalls.

Application layer firewalls (sometimes called Application-Proxy Gateway firewalls) are advanced firewalls that combine network and transport layer access control with application layer functionality. Application layer firewalls permit no traffic directly between the Internet and the internal network, or between two networks. These components can usually perform extensive logging and access control.

Application layer firewalls are considered the most secure type of firewall and have numerous advantages over network layer and stateful inspection firewalls:

- Logging capabilities
- Filtering capabilities (can filter specific types of Web content and specific HTTP commands)
- Easy to configure
- Resistant to IP spoofing attacks
- Provide user authentication.

Application layer firewalls also have some disadvantages as compared with network layer and stateful inspection firewalls:

- Slower
- Limited support for obscure and new protocols.

Although not strictly a limitation, application layer firewalls tend to be implemented on a workstation running a general-purpose operating system (e.g., Windows, Linux, and UNIX). This introduces an added layer of complexity because that general-purpose operating system must also be secured in addition to the firewall software itself. Network layer firewalls run on specialized operating systems, thus eliminating most of this risk.

To successfully protect a Web server using a firewall, ensure that it is capable of and configured to:

- Control all traffic between the Internet and the Web server

- Block all inbound traffic to the Web server except TCP ports 80 (HTTP) and/or 443 (HTTPS)
- Block (in conjunction with the intrusion detection system [see Section 7.2.2]) IP addresses or subnets that the IDS reports are attacking the organizational network
- Notify the network or Web administrator of suspicious activity through an appropriate means (e.g., page, e-mail and network trap)
- Provide content filtering
- Protect against of service attacks
- Detect malformed or know attack URL requests
- Log critical events including the following details:
  - Time and date
  - Interface IP address
  - Vendor-specific event name
  - Standard attack event (if one exists)
  - Source and destination IP address
  - Source and destination port numbers
  - Network protocol used by attack.
- Patched to the latest or most secure level (firewall application and underlying operating system).

Most firewall devices available in hardware and software perform some type of logging of the traffic they receive. For most firewalls, the default-logging configuration is suitable, provided logging is enabled. Administrators should consult their vendor documentation if they believe they require additional information logged. Certain brands of hardware-based firewalls include an ability to track and log information for each firewall policy. This ability enables accountability to a very specific extent.

One common feature that is available in many firewalls is the ability to selectively decide what information to log. If a firewall receives a series of similar packets from the same location it may decide not to log any additional packets after the first one. Although this is a valuable feature, consider the consequences: each packet that is dropped and not logged is potential evidence of a malicious intent. The principle of logging, which is a fundamental aspect of accountability, is discussed in greater detail in Section 8.1.

As with operating systems and other security-enforcing elements, a firewall may not necessarily be perfect; it may require updates. Although more prevalent in software implementations of firewall technology, hardware and router firewalls contain an ability to update their firmware. Specific instructions on how to update a firewall are found within the vendor documentation. Administrators should check for firewall updates at least once a week.

7.2.2 Intrusion Detection Systems

An IDS is an application that monitors system and network resources and activities and, using information gathered from these sources, notifies the network administrator when it identifies a possible intrusion or penetration attempt.[17]

The two principal types of IDSs are host-based and network based. Host-based IDSs must be installed on each individual computer system that is to be monitored or protected. Host-based IDSs are very closely integrated with the operating system they protect. Thus, a host-based IDS must be designed specifically for each operating system. These types of IDSs monitor network traffic to and from the host, the use of system resources, and the system log files.

Host-based IDSs are useful when most of the network traffic to and from the Web server is encrypted (e.g., when SSL/TLS is in use [see Section 6.5]) because the functionality and capability of network-based IDSs (see below) is severely limited when network traffic is encrypted. Furthermore, because Host-based IDSs are located on the server, they can detect some attacks and penetration attempts not recognized by network-based IDSs.

Host-based IDS can have a negative effect on host performance. In general, the greater the detection capabilities, the greater the negative impact on the performance of the host. Host-based IDSs may not detect some network-based attacks such as certain DoS attacks [NIST01b]. If a host-based IDS is on a Web server that is compromised, it is very likely that the attacker will also compromise the IDS itself.

Network-based IDS are implemented as protocol analyzers with the capability to recognize particular events. These devices monitor all network traffic on a network segment, scrutinizing it for signs of attack or penetration attempts. Most network IDSs rely on predefined "attack signatures" to detect and identify attacks. Attack signatures are a series of events that usually indicate that a particular attack or penetration attempt is in progress. When the IDS detects a series of events that matches one if its attack signatures, it assumes that an attack is in progress and notifies the network administrator.

Unlike host-based IDSs, network-based IDSs can monitor multiple hosts and even multiple network segments simultaneously. They can usually detect more network based attacks and can more easily provide a comprehensive picture of the current attacks against a network. Because network-based IDS are installed on a dedicated host, they do not have a negative effect on the performance of the Web server host and are not immediately compromised by a successful attack on the Web server.

Network-based IDSs do have some limitations. The timing of an attack can have a significant impact on the ability of a network-based IDS to detect an attack. For example, if an intruder spreads out the timing of his or her attack, the attack may not be detected by the IDS. In addition, the attacker can format the method of his attack (e.g., fragment packets, alter attack

[17] For more information about IDSs see NIST Special Publication 800-32, *Intrusion Detection Systems* (http://csrc.nist.gov/publications/nistpubs/index.html).

patter so that it does not match the attack signature) so that it is not recognized by the network-based IDS.

Network configuration, especially the use of switches (see Section 7.2.3) can have a negative impact on the ability of a network-based IDS to detect attacks. Network-based IDS are also more susceptible to being disabled by DoS attack (even those not directly targeted at the IDS).

Both host-based IDSs and network-based IDSs share some weaknesses. The most significant is the fact that no IDS today can detect all—and in many cases even most—of the attacks that exist today. Furthermore, IDSs require frequent updates to their attack signature databases in order to recognize new attacks. An IDS that is not updated frequently will fail to recognize the latest (and often most popular) attacks.

Although not strictly speaking IDSs, the following applications have some IDS capabilities and are a useful complement to an IDS:

- **Honey Pot**—is a host(s) that is (are) placed on a network for the strict purpose of attracting and detecting intruders. A honey pot will divert the attacker's attention from the "real" information system resources and will allow an organization to monitor the attacker's actions without risking "real" organizational information and resources.

- **File Integrity Checker**—computes and stores a checksum for every guarded file and establishes a database of file checksums. It provides a tool for the system administrator to recognize changes to files, particularly unauthorized changes (see Section 4.3). These are often included with host-based IDSs. See Appendix E for a listing of commonly available file integrity checkers.

To successfully protect a Web server using an IDS, ensure that it is capable of and configured to accomplish the following tasks:

- Monitor network traffic before any firewall or filter router (network based)
- Monitor traffic network traffic to and from the Web server
- Monitor changes to critical files on Web server (host-based or file-integrity checker)
- Monitor the system resources available on the Web server host (host-based)
- Block (in conjunction with the firewall) IP addresses or subnets that are attacking the organizational network
- Notify the network or Web administrator of attacks through appropriate means
- Detect port scanning probes
- Detect DoS attacks
- Detect malformed URL requests
- Log events including the following details:
  - Time and date

  - Sensor IP address
  - Vendor specific attack name
  - Standard attack name (if one exists)
  - Source and destination IP address
  - Source and destination port numbers
  - Network protocol used by attack.

- Updated with new attack signatures frequently (weekly basis).

Although not perfect, IDSs are a critical early warning system that can provide the Web administrator with the information necessary to defend the Web server from attack.

7.2.3 Network Switches

Network switches are devices that provide connectivity between two or more hosts located on the same network segments. They are similar to hubs in that they allow communications between hosts except that, unlike hubs, switches have more "intelligence" and send communications to only those hosts to which the communications are addressed. In other words, switches isolate the communications of hosts on a network segment from each other. One benefit of this isolation is that it may reduce the impact of a DoS attack on other hosts on the network.

The benefit of this from a security standpoint is that when switches are employed on a network, it is much more difficult to eavesdrop on communications between other hosts on the network segment. This benefit is extremely important when a Web server is on a network segment that is used by other hosts. For example, if a hub is used and the Web server is compromised, an attacker may be able to eavesdrop on the communications of other hosts possibly leading to the compromise of those hosts or the information they communicate across the network. A primary example of this would be e-mail servers, which are often located with the Web servers, and which, in their default configurations, receive unencrypted passwords. In this instance, the compromise of the Web server would lead to the eventual compromise of the mail server unless a switch was being used. A switch would prevent, or at least hinder, the attacker from sniffing mail server passwords from the compromised Web server.

Many switches include specific security settings that further enhance of the security of the network by making it difficult for a malicious entity to "defeat" the switch. Some examples include the ability to minimize the risk of ARP spoofing and ARP poisoning attacks. If a switch has these security capabilities, they should be enabled (see appropriate vendor documentation).

7.3 Network Infrastructure Checklist

Completed	Action
	Network location
☐	The Web server is located in a DMZ or outsourced to an organization that appropriately protects the firewall
☐	The DMZ is not located on the third (or more) interface of the firewall
	Firewall configuration
☐	Web server is protected by a firewall
☐	Web server if it faces a higher threat or if it is more vulnerable, is protected by an application layer firewall
☐	Firewall controls all traffic between the Internet and the Web server
☐	Firewall blocks all inbound traffic to the Web server except TCP ports 80 (HTTP) and/or 443 (HTTPS)
☐	Firewall blocks (in conjunction with IDS) IP addresses or subnets that the IDS reports are attacking the organizational network
☐	Firewall notifies the network or Web administrator of suspicious activity through an appropriate means
☐	Firewall provides content filtering
☐	Firewall configured to protect against of service attacks
☐	Firewall detects malformed or known attack URL requests
☐	Firewall logs critical events
☐	Firewall and firewall operating system patched to latest or most secure level
	Intrusion detection systems (IDS)
☐	Host-based IDS used for Web servers that operate primarily SSL/TLS
☐	IDS configured to monitor network traffic before any firewall or filter router (network-based)
☐	IDS configured monitor traffic network traffic to and from the Web server after firewall
☐	IDS configured to monitor changes to critical files on Web server (host-based or file-integrity checker)
☐	IDS blocks (in conjunction with the firewall) IP addresses or subnets that are attacking the organizational network
☐	IDS notifies the network or Web administrator of attacks through appropriate means
☐	IDS configured to detect port scanning probes
☐	IDS configured to detect DoS
☐	IDS configured to detect malformed URL requests
☐	IDS configured to log events

Completed	Action
☐	IDS updated with new attack signatures frequently (weekly basis)
☐	IDS configured to monitor the system resources available on the Web server host (host-based)
	Network switches
☐	Network switches are used on Web server network segment to protect against network eavesdropping
☐	Network switches are configured in high-security mode to defeat ARP spoofing and ARP poisoning attacks
☐	Network switches are configured to send all traffic on network segment to IDS host (network-based)

8. Securely Administering a Web Server

8.1 Logging

Logging is the principal component of securely administering a Web server. Logging the appropriate data and then monitoring and analyzing those logs are critical. Web server host logs are important, particularly for encrypted traffic, where network monitoring is far less effective. Reviewing logs is mundane and many Web administrators have a difficult time fitting log file analysis into their hectic schedules. This is unfortunate as log files are often the best and/or only record of suspicious behavior. Failure to enable the mechanisms to record this information and use them to initiate alert mechanisms will greatly weaken or eliminate the ability to detect intrusion attempts and to determine whether they succeeded. Similar problems can result from not having the necessary procedures and tools in place to process and analyze the log files.

System and network logs can alert the Web administrator that a suspicious event has occurred and requires further investigation. Web server software can provide additional log data relevant to Web-specific events. If the Web administrator does not take advantage of these capabilities, Web-relevant log data may not be visible or may require a significant effort to access.

Web server logs provide the following:

- Alerts to suspicious activities that require(s) further investigation
- Tracking of an intruder’s activities
- Assistance in the recovery of the system
- Assistance in the post-event investigation
- Required information for legal proceedings.

The selection and implementation of specific Web server software will determine which set of detailed instructions the Web administrator should follow to establish logging configurations. Some of the guidance contained in the steps below may not be fully applicable to all vendors' Web server software products.

8.1.1 Identifying the Logging Capabilities of a Web Server

Each Web server application supports a different logging capability. Depending on the Web server application, one or more of the following logs may be available [CERT00]:

- **Transfer Log**—Each transfer is represented as one entry showing the main information related to the transfer.
- **Error Log**—Each error is represented as one entry, including an explanation of the reason for this error report.

- **Agent Log**—Contains information about the user client software used in accessing Web content.

- **Referrer Log**—Collects information relevant to HTTP access. This includes the URL of the page that contained the link that the user client software followed to initiate the access to the Web page.

Most Webs servers support the Transfer Log and it is usually considered the most important. Several log formats are available for Transfer Log entries. Typically, the information is presented in plain American Standard Code for Information Interexchange ASCII without special delimiters to separate the different fields [CERT00]:

- **Common Log Format**—This format stores the following information related to one transfer (Transfer Log) in the indicated order:

  - Remote host
  - Remote user identity in accordance with RFC 1413[18]
  - Authenticated user in accordance with the basic authentication scheme (see Section 6.3)
  - Date
  - URL requested
  - Status of the request
  - Number of bytes actually transferred.

- **Combined Log Format**—Contains the same seven fields above. It also provides information normally stored in the Agent Log and the Referrer Log, along with the actual transfer. Keeping this information in a consolidated log format may support more effective administration.

- **Extended Log Format**—Provides a way to describe all items that should be collected within the log file. The first two lines of the log file contain the version and the fields to be collected, and they appear in the log file as follows:

```
#Version: 1.0
#Fields: date time c-ip sc-bytes time-taken cs-version
1999-08-01 02:10:57 192.0.0.2 6340 3 HTTP/1.0
```

This example contains the date, time, originating address, number of bytes transmitted, time taken for transmission, and the HTTP version.

- **Other Log File Formats**—Some server software provides log information in different file formats, such as database formats or delimiter-separated formats. Other server

[18] See the Internet Engineering Task Force Web site: http://www.ietf.org/rfc/rfc1413.txt?number=1413.

software provides the capability for an administrator to define specific log file formats in the Web server configuration file using a particular syntax (if the default CLF format is insufficient).

8.1.2 Identifying Additional Logging Requirements

If a public Web server supports the execution of programs, scripts, or plug-ins, the Web administrator should determine whether specific logging data should be captured regarding the performance of these features. If a Webmaster develops his or her own programs, scripts, or plug-ins, it is strongly recommended that they define and implement a comprehensive and easy-to-understand logging approach based on the logging mechanisms provided by the Web server hosting operating system. Log information associated with programs, scripts, and plug-ins can add significantly to the typical information logged by the Web server.

8.1.3 Recommended Generic Logging Configuration

The following configuration is a good starting point for logging on public Web servers [CERT00]:

- Use the Combined Log Format for storing the Transfer Log or manually configure the information described by the Combined Log Format to be the standard format for the Transfer Log.
- Enable the Referrer Log or Agent Log if the Combined Log Format is unavailable.
- Establish different log file names for different virtual Web sites that may be implemented as part of a single physical Web server.
- Use the Remote User Identity as specified in RFC 1413.

Some Web server software provides a capability to enforce or disable the checking of specified access controls during program startup. This level of control may be helpful to, for example, avoid inadvertent alteration of log files as a result of errors in file access administration. Web administrators should determine the circumstances under which he or she may want to enable such checks (assuming the Web server software supports this feature).

8.1.4 Reviewing and Retaining Log Files

Reviewing log files can be time-consuming and boring. Log files are an inherently reactive security measure: they inform of events that have already occurred. Accordingly, they are often useful for corroborating other evidence, whether it is a central processing unit (CPU) utilization spike or anomalous network traffic reported by an IDS. When a log is used to corroborate other evidence, a focused review is in order. For example, if an IDS reported an outbound FTP connection from the Web server at 8:17 a.m., then a review of the logs generated just before 8:17 a.m. is appropriate. Web server logs should also be reviewed for indications of attacks. The frequency of the review will depend on the following factors:

- Traffic the server receives

- General threat level (certain sites, in particular the Federal Government and certain commercial institutions receive many more attacks than other sites and thus should review their logs more frequently)
- Specific threats (at certain times specific threats arise that may require more frequent log file analysis as a result)
- Vulnerability of the Web server
- Value of data and services provided by Web server.

Reviews should take place on a daily to weekly basis and when a suspicious activity has been noted or a threat warning has been issued. Obviously, the task could quickly become burdensome to a Web administrator. To reduce this burden, automated log file analysis tools have been developed (see Section 8.1.5).

In addition, long-term and more in-depth analyses of the logs are needed. Because a typical Web attack can involve hundreds of unique requests, an attacker may attempt to disguise a Web attack by increasing the interval between requests. In this case, reviewing a single day's or week's logs may not show recognizable trends. However, when trends are analyzed over the course of a week, a month, or a quarter, multiple attacks from the same host are more easily recognized.

Log files should be protected to ensure that if an attacker does compromise a Web server, he or she cannot alter the log files to cover his or her actions. Although encryption can be useful in protecting log files, the best solution is to store log files on a host separate from the Web server(s). This is often called a log or syslog host.

Log files should be backed up and archived regularly. Archiving log files for a period of time is important for several reasons. They can be important for certain legal actions and they are often useful in troubleshooting problems with the Web server. The retention period for archived log files depends on a number of factors, including the following:

- Legal requirements
- Organizational requirements
- Size of logs (which is directly related to the traffic of the site and the number of details logged)
- Value of Web server data and services
- Threat level.

8.1.5 Automated Log File Analysis Tools

Most public Web servers receive significant amounts of traffic, and the log files quickly become voluminous. To ease the burden on the Web administrator, it is generally necessary to install one or more automated log file analysis tools. These tools analyze the entries in the Web server log file and identify suspicious and unusual activity.

Many commercial and public domain tools are available to support regular analysis of Transfer Logs. Most operate on either the Common or the Combined Log Formats.

These tools can identify IP addresses that are the source of high numbers of connections and transfers.

Error Log tools indicate not only errors that may exist within available Web content (such as missing files) but also attempts to access non-existing URLs. Such attempts could indicate the following:

- Probes for the existence of vulnerabilities to be used later in launching an attack
- Information gathering
- Interest in specific content such as databases.

The automated log file analyzer should forward any suspicious log file events to the responsible Web administrator or security incident response team as soon as possible for follow-up investigation. A list of some commonly used log file analyzers is provided in Appendix E.

8.2 Web Server Backup Procedures

One of the most important functions of a Web server administrator is to maintain the integrity of the data on the Web server. This is vitally important because a Web server is often the most exposed server on an organization's network and thus is often the most susceptible to malicious actions and possible hardware and software failures. There are two principal components to backing up data on a Web server: regular backup of the data and operating system on the Web server and maintaining a separate protected authoritative copy of the organization's Web site(s).

8.2.1 Web Server Backup Policies and Strategies

The Web administrator needs to perform backups of the Web server regularly. This is critical for several reasons. If the Web server fails either because of a malicious or unintentional act or a hardware or software failure, the Web server can be restored in a timely manner. In addition, federal and state governmental organizations are governed by regulations on the backup and archiving of Web server data. Certain commercial organizations may also wish to back up their Web server data regularly for legal or financial reasons.

All organizations need to create a Web server data backup policy. The contents of this policy will be influenced by three factors:

- Legal requirements
  - Applicable laws and regulations (federal, state, and international)
  - Litigation requirements
- Business requirements

- Contractual
- Common industry practices
- Criticality of data to organization

- Organizational guidelines and policies

Although each organization's Web server backup policy will be different to reflect its particular environment, it should address the following issues:

- Purpose of the Web server backup policy
- Who is affected by the Web server backup policy
- Which Web servers are covered by the backup policy
- Define key terms, especially legal and technical
- Describe the requirements in detail from the legal, business, and organization's perspective
- Outline frequency of backups
- Outline the procedures for ensuring data is properly retained and protected
- Outline the procedures for ensuring data is properly destroyed or archived when no longer required
- Clearly document the litigation exception process and how to respond to discovery requests
- List the responsibilities of those involved in data retention, protection and destruction activities
- Build a table showing the information type and it corresponding retention period
- Document the specific duties of a central/organizational data backup team if one exists.

  Two primary types of backups exist. Full backups, as their name implies, are a full backup of the operating system, applications, and data stored on the Web server (i.e., an image of every piece of data stored on the Web server hard drive[s]). The advantage of a full backup is that it is easy to restore the entire Web server to the state (configuration, patch level, data, etc.) that existed when the backup was performed. The disadvantage of full backups is that they take considerable time and resources to perform. Incremental backups reduce the effect by backing up only data that has changed since the previous backup (either full or incremental). Generally, full backups are performed less frequently (weekly to monthly or when a significant change occurs) than incremental backups (daily to weekly). The frequency of backups will be determined by several factors:

- Volatility of information on the Web site

  - Static Web content (less frequent backups)
  - Dynamic Web content (more frequent backups)
  - E-commerce/E-government (very frequent backups)
- Amount of data to be backed up
- Backup device and media available
- Time available for dumping backup data
- Criticality of data
- Threat level faced by Web server
- Effort required to data reconstruction without data backup.

8.2.2 Maintain an Authoritative Copy of Organizational Web site(s)

All organizations should maintain an authoritative (i.e., verified and trusted) copy of their public Web sites on a host that is inaccessible to the Internet. This is a supplement to, but not replacement for, an appropriate backup policy (see Section 8.2.1). For simple relatively static Web sites, this could be as simple as a copy of the Web site on a read-only medium (e.g., CD-R). However, for most organizations, the authoritative copy of the Web site is maintained on a secure host. This host is usually located behind the organization's firewall on the internal network and NOT on the DMZ (see Section 7.1.3). The purpose of the authoritative copy is to provide a means of restoring information on the public Web server if it is compromised as a result of an accident or malicious action. This authoritative copy of the Web site allows organization to rapidly recover from Web site defacement, which is the most common Web server attack.

To successfully accomplish the goal of providing and protecting an authoritative copy of the Web server content, the following requirements must be met:

- Protect authoritative copy from unauthorized access
  - Use write once media (appropriate for relatively static Web sites)
  - Locate host with authoritative copy behind firewall, and ensure there is no outside access to host)
  - Minimize users with authorized access to host
  - Control user access in a granular manner as possible
  - Employ strong user authentication
  - Employ appropriate logging and monitoring procedures
  - Consider additional authoritative copies at different physical locations for further protection.

- Establish appropriate authoritative copy update procedures
  - Update authoritative copy first (any testing on code should occur before updating the authoritative copy)
  - Establish policies and procedures for who can authorize updates, perform updates, and when updates can occur, etc.
- Establish a process for copying authoritative copy to production Web server
  - Data can be transfer using a secure physical media (e.g., encrypted and/or write once media such as a CD-R)
  - Network transfer using a secure protocol (e.g., SSH).
- Include the procedures restoring from the authoritative copy in the organizational incident response procedures (see Section 8.3)
- Consider automatic updates from authoritative copy to Web server periodically (quarter hourly, hourly, daily, etc.) because this will overwrite a Web site defacement automatically.

8.3 Recovering From a Security Compromise

Most organizations will eventually face a successful compromise of one or more hosts on their network. The first step in recovering from a compromise is to create and document the required policies and procedures for responding to successful intrusions *prior* to an intrusion[19]. The response procedures should outline the actions that are required to respond to a successful compromise of the Web server and the appropriate sequence of these actions (sequence can be critically important). These response procedures would be contained within the organization's security policy.

A Web administrator should take the following steps once a successful compromise is identified:

- Consult the organization's security policy
- Disconnect compromised system(s) from networks or take steps to contain attack so additional evidences can be collected
- Investigate other "similar"[20] hosts to determine if the attacker also has compromised other systems

[19] For more information on this area see*:* NIST Special Publication 800-3, *Establishing a Computer Security Incident Response Capability* and NIST Special Publication 800-18, *Guide to Developing Security Plans for Information Technology Systems* (http://csrc.nist.gov/publications/nistpubs/index.html).

[20] "Similar" would include hosts in the same IP address range, that have the same or similar passwords, that share a trust relationship, and/or that have the same operating system and/or applications.

- Consult with management, legal counsel, and law enforcement as appropriate (contact law enforcement immediately if prosecution is desired)
- Analyze the intrusion, including:
  - Modifications made to the system's software and configuration
  - Modifications made to the data
  - Tools or data left behind by intruder
  - Review system logs, intrusion detection and firewall log files.
- Restore the system
  - Two options exist:
    - Install clean version of operating system
    - Restore from backups (more risky).
  - Disable unnecessary services
  - Apply all patches
  - Change all passwords (even on uncompromised hosts) as required
  - Reconfigure network security elements (e.g., firewall, router, IDS) to provide additional protection and notification
- Reconnect system to network
- Test system to ensure security
- Monitor system and network for signs that the attacker is attempting to access the system or network again
- Document lessons learned.

System administrators should consider the following when deciding whether to reinstall the operating system of a compromised system as opposed to restoring from a backup:

- Level of access that the intruder gained (e.g., root, user, guest, system)
- Type of attacker (internal or external)
- Purpose of compromise (e.g., Web page defacement, illegal software repository, platform for other attacks)
- Method of system compromise

- Actions of hacker during and after compromise (e.g., see log files, intrusion detection reports)
- Duration of compromise
- Extent of compromise on network (i.e., the number of hosts compromised)
- Results of consultation with management and legal counsel.

The lower the level of access gained by the intruder and the more the Web administrator knows about the hacker's actions, the less risk there is in restoring from a backup and patching the vulnerability. The less known about the intruder's actions, the more highly recommended it is to reinstall all software on the host.

If legal action is pursued, system administrators need to be aware of the guidelines for handling a host after a compromise. For more information see the National Infrastructure Protection Center's (NIPC) Web site (http://www.nipc.gov).

8.4 Security Testing Web Servers

Periodic security testing of public Web servers is critical. Without periodic testing, there is no assurance the current protective measures are effective or that the security patch just applied is functioning as advertised. Although a variety of securities testing techniques exist, the most commonly used on Web servers are vulnerability scanners. Vulnerability scanning assists a Web administrator in identifying vulnerabilities and verifying whether the existing security measures are working.[21]

8.4.1 Vulnerability Scanning

Vulnerability scanners are automated tools used to identify vulnerabilities and misconfiguration of hosts. Many vulnerability scanners also provide information about mitigating discovered vulnerabilities.

Vulnerability scanners attempt to identify vulnerabilities in the hosts scanned. Vulnerability scanners can help identify out-of-date software versions, vulnerabilities, applicable patches, or system upgrades, and validate compliance with, or deviations from, the organization's security policy. To accomplish this task, vulnerability scanners identify operating systems and major software applications running on hosts and match them against a catalogue of known vulnerabilities associated with commonly used operating systems and applications. Two principal types of vulnerability scanners are of use to Web administrators. General-purpose vulnerability scanners (e.g., CyberCop and Nessus) scan for numerous application, network, and operating system vulnerabilities including some Web vulnerabilities. There are also specific vulnerability scanners (e.g., WebInspect or Whisker) that address Web server or some subcomponent of Web server vulnerabilities (e.g., CGI scripts).

However, vulnerability scanners have some significant weaknesses. Usually they identify only surface vulnerabilities and are unable to address the overall risk level of a scanned Web server.

[21] For information about other testing techniques, see NIST Special Publication 800-42, *Guideline on Network Security Testing* (http://csrc.nist.gov/publications/nistpubs/index.html).

Although the scan process itself is highly automated, vulnerability scanners can have a high false positive error rate (reporting vulnerabilities when none exist). This means an individual with expertise in Web server security and administration must interpret the results. Vulnerability scanners cannot usually identify vulnerabilities in custom code or applications.

Vulnerability scanners rely on periodic updating of the vulnerability database to recognize the latest vulnerabilities. Before running any scanner, Web administrators should install the latest updates to its vulnerability database. Some vulnerability scanner databases are updated more regularly than others (the frequency of updates should be a major consideration when choosing a vulnerability scanner).

Vulnerability scanners are often better at detecting well-known vulnerabilities at the expense of more esoteric ones because it is impossible for any one product to incorporate all known vulnerabilities in a timely manner. It is also because of the manufacturers' desire to keep the speed of their scanners high (more vulnerabilities detected require more tests, which slows the overall scanning process). Therefore, vulnerability scanners may be of little use to Web administrators operating less popular Web servers, operating systems or custom-coded applications.

Vulnerability scanners provide the following capabilities:

- Identifying active hosts on network
- Identifying active services (ports) on hosts and which of these are vulnerable
- Identifying application and banner grabbing
- Identifying operating systems
- Identifying vulnerabilities associated with discovered operating systems and applications
- Testing compliance with host application usage/security policies.

Organizations should conduct vulnerability scanning to validate that operating systems and Web server applications are up to date regarding security patches and software versions. Vulnerability scanning is a reasonably labor-intensive activity that requires a high degree of human involvement with interpreting the results. It may also be disruptive to network operations by taking up bandwidth and slowing response times. However, vulnerability scanning is extremely important for ensuring that vulnerabilities are mitigated as soon as possible, before they are discovered and exploited by adversaries. Vulnerability scanning should be conducted at least quarterly.

Vulnerability scanning results should be documented and discovered deficiencies corrected. Network and host-based vulnerability scanners are available for free or for a fee. Appendix E contains a list of readily available vulnerability scanning tools.

8.5 Remotely Administering a Web Server

An item of Web server administration that needs to be very carefully considered is whether to enable the capability to remotely administer and/or update content on a Web server. The most secure configuration is to disallow any remote administration or content updates, although that might not be viable for all organizations. The risk of enabling remote administration or content updates varies considerably depending on the location of the Web server on the network (see Section 7.1). For example, if the Web server is located external to the organization's firewall or IP filtering router, then in no circumstances should remote administration or updating be implemented. For a Web server that is located behind a firewall, then remote administration or content updating can be implemented relatively securely from the internal network. Remote administration or content updating should not be allowed from host located outside the organization's network.

If an organization determines that it is necessary to remotely administer or update content on a Web server, following these steps should ensure that it is implemented in a secure manner as possible:

- Use a strong authentication mechanism (e.g., public/private key pair, two factor authentication, etc.)
- Restrict hosts that can be used to remotely administer or update content on the Web server
  - Restrict by IP address (not hostname)
  - Restrict to hosts on the internal network
- Use secure protocols (e.g., SSH, HTTPS), not insecure protocols (e.g., Telnet, file transfer protocol [FTP], network file system (NFS) or HTTP). Secure are those protocols that provide encryption of both passwords and data.
- Enforce the concept of least privilege on the remote administration and content updating (i.e., attempt to minimize the access rights for the remote administration/update account[s]).
- Do not allow remote administration from the Internet through the firewall.
- Change any default accounts or passwords form the remote administration utility or application.
- Do not mount any file shares on the internal network from the Web server or vice versa.

8.6 Securely Administering a Web Server Checklist

Completed	Action
	Logging
☐	Use the Combined Log Format for storing the Transfer Log or manually configure the information described by the Combined Log Format to be the standard format for the Transfer Log
☐	Enable the Referrer Log or Agent Log if the Combined Log Format is unavailable
☐	Establish different log file names for different virtual Web sites that may be implemented as part of a single physical Web server
☐	Use the Remote User Identity as specified in RFC 1413

Completed	Action
☐	Store logs on a separate (syslog) host
☐	Archive logs according to organizational requirements
☐	Review logs daily
☐	Review logs weekly (for more long-term trends)
☐	Use automated logfile analysis tool(s)
	Web server backups
☐	Create a Web server backup policy
☐	Back up Web server incrementally on a daily to weekly basis
☐	Back up Web server fully on a weekly to monthly basis
☐	Periodically archive backups
☐	Maintain an authoritative copy of Web site(s)
	Recovering from a compromise
☐	Consult the organization’s security policy (this should take precedence over the recommendations provided here)
☐	Disconnect compromised system(s) from network or take steps to contain attack so additional evidences can be collected
☐	Investigate other “similar ” hosts to determine if the attacker also has compromised other systems
☐	Consult with management, legal counsel, and law enforcement as appropriate (contact law enforcement immediately if prosecution is desired)
☐	Analyze the intrusion
☐	Restore the system
☐	Reconnect system to network
☐	Test system to ensure security
☐	Monitor system and network for signs that the attacker is attempting to access the system or network again
☐	Document lessons learned
	Security testing
☐	Periodically conduct vulnerability scans on Web server and network supporting network
☐	Update vulnerability scanner prior to testing
☐	Correct any deficiencies identified by the vulnerability scanner
	Remote administration and content updating
☐	Use a strong authentication mechanism (e.g., public/private key pair, two factor authentication)
☐	Restrict hosts that can be used to remotely administer or update content on the Web server by IP address and to the internal network
☐	Use secure protocols (e.g., secure shell, HTTPS)

Completed	Action
☐	Enforce the concept of least privilege on the remote administration and content updating (i.e., attempt to minimize the access rights for the remote administration/update account[s])
☐	Change any default accounts or passwords from the remote administration utility or application
☐	Do not allow remote administration from the Internet through the firewall
☐	Do not mount any file shares on the internal network from the Web server or vice versa

Appendix A. Securing Apache Web Server

Apache is an open source (collaborative, consensus-based) software development effort aimed at creating a commercial-grade and freely available version of a Hypertext Transfer Protocol (HTTP) (i.e., Web) server. The project is jointly managed by a group of volunteers located around the world, using the Internet to communicate, plan, and develop the server and its related documentation. Hundreds of users have contributed ideas, code, and documentation to further the development of Apache.

A.1 Installation

A.1.1 Install and Secure the Host Operating System

If the operating system underlying the Web server is not secured, then the Web server is insecure regardless of its configuration. Before installing any Web server, ensure that the host computer and operating system are completely secure and hardened. Although beyond the scope of this document for the most part, hardening an operating system includes the following steps:

- Apply latest patches to operating system
- Disable or remove all unnecessary services and applications
- Apply appropriate permissions to system resources
- Use appropriately strong identification and authentication mechanisms.

A strong password should always be used for the root or administrator account on the operating system that underlies Apache. A weak password can result in an otherwise hardened platform to be hacked easily. The only additional services needed on a Web server should be Secure Shell (SSH) for remote administration and Network Time Protocol (NTP) to synchronize systems clocks so that log file correlation is easier. The Web server should never have a compiler or X Windows or other remote administration system installed.

Transmission Control Protocol/Internet Protocol (TCP/IP) packet filtering using ipchains or iptables should also be used to restrict traffic only to what should be allowed and expected. The outside interface (the interface on which public traffic arrives) of the Web server should allow only TCP ports 80 and 443 incoming traffic and greater than or equal to TCP port 1024 for outgoing traffic. The inside interface should allow only TCP ports 22, 80, and 443 incoming and greater than or equal to 1024 outgoing. DNS resolution should be allowed via User Datagram Protocol (UDP) to allow for logging of hostnames, conducting reverse look-ups, and running identd.

A.1.2 Create an Unprivileged User Account for the Apache Web server

To reduce the risk of exposure when operating an Apache Web server, use a unique and unprivileged userid and group created solely for the Web server application. In many instances, the userid and group "nobody" is used for these purposes. Although this is acceptable, creating a unique user and group for the Web server is preferable.

When the Apache server first initializes, root privileges are required to open its log files, start the Web server daemon, and open the appropriate TCP ports. Once this startup sequence has been completed, the Web server daemon should be configured to switch to the unprivileged user and group created above.

The unprivileged account that is created for Apache should be configured not to allow interactive logins (e.g., it should not allow users to login via that account). This can be verified by checking the entry for the Apache account in /etc/shadow password file. The password for the user should indicate that the account is locked and cannot be used to log into.

A.1.3 Install the Apache Server Software

To secure the Web server, the server daemon and content should be installed on separate hard disk partitions. The Web content should be installed in the DocumentRoot. The DocumentRoot is the directory structure in Apache within which all the Web content is stored. If possible, this directory should be stored on a separate partition or hard drive from the server root or chroot (see below). Separating the content from the Apache Web server application files makes it much more difficult for an attacker to compromise the Web server.

The htdocs directory included in the Apache distribution should not be used as the DocumentRoot. The htdocs directory contains Apache documentation that should not be made available to the public. This directory contains information about the system that could be used in an attack.

The Web server daemon should also be installed in a chroot jail. Chroot is a contained environment outside of the regular file structure. This contained environment is used run the server so if any compromise of the server occurs, it is contained within the chroot "jail" and the attacker cannot "escape" to other areas of the server. This contained environment effectively blocks the attacker who is exploiting the server from accessing the remainder of the server's file system, limiting the damage of the attack.

The src directory included in the Apache distribution should not be installed on the server because this directory contains the source code for compiling the Web server executable and is not needed.

A.1.4 Set Permissions for the Web server Directories and Flies

All commands that are executed by root should be protected at all times. Ensure that all non-root users are unable to modify the commands that root executes. It is important to protect the files, directories, and parents of all directories. To protect these, it is important to make them writable only by root.

If users with less than root privileges are able to modify any files that root executes or writes, then the system becomes open to many vulnerabilities and exploits. In normal usage of Apache as a Web server, Apache is started as root and then switches to the user defined in the configuration of Apache (see above). This is to protect the system from the risk of root becoming compromised while serving Web pages.

Apache Web server software groups file by function into subdirectories. Table A.1 lists the directories that should be in the ServerRoot or CHroot jail.

Table A.1: Apache Web server Directories

Name	Function	Contents
conf	Web server configuration	httpd.conf, srm.conf, access.conf
logs	Web server logs	Access_log, agent_log, error_log
cgi-bin	Web executables	CGI Scripts
icons	Icons	
support	Tools	Utilities for Administering

A.1.5 Delete All Vulnerable or Unknown CGI Scripts

All Computer Gateway Interface (CGI) scripts that are included with the Apache distribution should be removed. In addition, no CGI scripts should be installed until they have been tested thoroughly and found to be safe. CGI scripts and other active contents have significant potential to cause vulnerabilities that may compromise the Web server (see Section 5.2).

CGI scripts, if absolutely required, should also be run from a strictly controlled cgi-bin directory that does not allow the use of shell scripts. This privilege should be given to only users who can be trusted. Although the users may be trusted, any CGI scripts or programs should be tested for potential vulnerabilities (see Section 5.2). Although security holes may not have been added intentionally by programmers, a potential exists for holes to be introduced inadvertently. Unchecked CGI input should never be passed onto the UNIX command line. This is important so that malicious users cannot add malicious code or commands to an input that could possibly get executed as root.

A.1.6 Delete All Unnecessary Files From the HTML Document Tree

Within the DocumentRoot directory, all unnecessary files should be removed. This is especially important with files that may contain information that should not be accessible to the public. Although there may not be any links to the document in question, it may still be possible for the public or malicious entities to access the document(s).

A.1.7 Protect System Settings

All users should be prevented from creating .htaccess files. These files have the potential to override the configured security settings. The best method for preventing this is to add the following entry to the server configuration file:

```
<Directory />
AllowOverride None
Options None
Allow from all
</Directory>
```

This will stop all overrides, includes, and accesses in all directories from occurring.

A.2 Initial Configuration

A.2.1 Make Working Copies of the Server Configuration Files

The initial server configuration files and any later baseline configurations should be backed up and archived before the server is accessible to the public or other changes are made. This allows for the rapid restoration of service after an attack or inadvertent misconfiguration.

A.2.2 Disable Automatic Directory Listings and Symbolic Links

The httpd.conf file controls many critical security-related configuration settings of the Web server. The directory directive for htdocs should have the path changed to match that of DocumentRoot.

To prevent the server from listing all the files within a directory, the entry for Indexes should be changed from the default FollowSymLinks to IncludesNoExec. This action prevents not only automatic directory listings from occurring, but also the use of symbolic links that could allow access to files or directories that are outside the Web server's DocumentRoot. Server Side Includes (SSI) are still allowed (see below), but execution of code using an SSI is prevented. CGI execution will be restricted to the ScriptsAlias directory.

A.2.3 Server Side Includes

SSIs are directives placed inside Hypertext Markup Language (HTML) pages that are evaluated when the pages are served. SSIs enable the Webmaster to add dynamically generated content to an existing HTML page without having to serve the whole page with a CGI program or other method. SSI is a good method for adding small amounts of information to a static page. SSIs should not be used if large amounts of information are generated.

SSIs can also be configured to allow malicious users to execute arbitrary programs on the server. This feature of SSIs should be disabled using the IncludesNoExec option within the Options directive. For more information on the risks associated with active content, see Section 5.2.

A.2.4 Default Apache Locations and Formats

Apache logs are stored in the default location /usr/local/apache/logs. The most useful log is the access_log, but other files such as ssl_request_log and ssl_engine_log can also provide valuable information. By default, the Apache access_log contains eight fields:

- Client Internet Protocol (IP) address
- Unique personal id
- Username
- Date
- Method
- Uniform Resource Identifier (URI) stem

- HTTP status
- Number of bytes transferred.

A.2.5 Apache Log Options

For Apache, logging is controlled within the file httpd.conf, using the LogFormat and CustomLog directives. The LogFormat controls which attributes are logged. The CustomLog directive names only the log file access_log by default. There are many options for LogFormat. A complete list of these options can be found at the Apache Web site (http://httpd.apache.org/docs/mod/mod_log_config.html). Table z-2 provides a partial list.

Table A.2: Apache Logging Options

Option	Action
%a	Remote IP address
%A	Local IP address
%B	Bytes sent, excluding HTTP headers
%f	Filename
%h	Remote host
%P	The process ID of the child that serviced the request
%T	The time, in seconds, taken to serve the request
%U	The URL path requested

A.2.6 Configure Access Control and Authentication

Several capabilities exist within Apache for controlling access and authenticating users or host before allowing access. This can include allowing or denying connections from specified IP addresses or subnets in addition to authenticating the user by requiring a username and password.

Apache has three methods of determining if the resources requested by a user will actually be served to that user: authentication, authorization, and access control.

- **Authentication**—is the process by which the user's claimed identity is verified. This process is usually accomplished though the use of usernames and passwords. The process can also be accomplished though the use of biometric or smart card devices.
- **Authorization**—is the process by which a user's permission to access a particular resource is determined to be valid. The criteria by which permissions are granted or denied can vary depending on the resource and the rules associated with that resource.
- **Access Control**—is a process similar to authorization. The difference is that access control is not dependent on the user attempting to access particular resources. Access to resources is granted based on attributes that are not based on the user.

-

Within Apache, authentication can occur though Basic Authentication, Digest Authentication, and Database Authentication Modules.

A.2.7 Basic Authentication

When resources are protected using basic authentication, any time a request is made for the protected resource, the user is required to send credentials in the form of a username and password to the server. If the username is within the approved list and the password is correct, access is granted to the protected resource. Because HTTP is a stateless protocol, the credentials must be sent with every request for the protected resource. This action is usually taken care of automatically within the client software or Web browser that is being used to access the protected resource.

Although passwords are stored in an encrypted manner when using basic authentication, they are not encrypted when they are transmitted from the client to the server. The fact that the password are transmitted unencrypted, allows third parties that have access to networking equipment, along with the path the username and password is sent, to easily intercept and compromise the user's credentials. The weakness is further compounded in that the username and password are sent with every communication, making it much easier for a malicious entity to compromise the credentials. In addition, basic authentication does not provide any encryption of the data transmitted to the user from the server, so that too is susceptible to interception and compromise. Consequently, it is not recommended that this form of authentication be used for any public Web server.

A.2.8 Digest Authentication

A second type of authentication is Digest Authentication. The difference between Digest Authentication and Basic Authentication is that with Digest Authentication the password is never sent across the wire or network in the clear. The password is sent using a Message Digest 5 (MD5) encrypted hash of the user's password. Although an attacker can still intercept the password, he or she will have to expend additional time and resources to crack (unencrypt) the password hash.

Although the password is not sent in clear text, the digest could still be used to access any protected information by a skilled hacker. In addition, the protected data is transferred in clear text so that data could still be compromised. In addition, not all Web browsers support digest authentication.

A.2.9 Database Authentication Modules

Database Authentication Modules allow usernames and passwords to be checked more rapidly. When large numbers of username and password combinations reside on a server, the amount of time required to authenticate a user each time that a user is accessing protected data can become time consuming and cumbersome. The time required to match the username and password in a flat file exists because every time a user requests data, the server must search through the entire flat file until the username is found. On average, this means that the flat file is searched through line by line for half the number of usernames in the file. The use of a database to store the usernames and passwords drastically reduces the search time to confirm the username and password.

A.2.10 Access Control

Access control allows the protected data to be served to users based on criteria other than whom the user claims to be. These criteria may be characteristics such as the location of the IP address or domain of the host the client is using. Access control is configured by the fields of the Order and Allow directives. The default configuration of the server allows everyone full access to the contents of the DocumentRoot. The Allow and Deny directives within Apache allow access based on the host name or host address of the machine of the user is using requesting the data. The Satisfy directive within Apache allows several criteria to be used in deciding whether a user or machine is granted access to protected data.

To restrict access to a local intranet, for example 192.168.0.0/16, the following setting would accomplish this:

```
order deny, allow
deny from all
allow from 192.168
```

The above setting would allow anyone within the 192.168.0/0/16 subnet access to the Web server's content while denying everyone else. The deny from and allow from directives can use host, domain names, IP addresses, IP addresses with subnet mask, or IP address with Classless InterDomain Routing (CIDR) mask size. If possible, IP addresses should be used over domain names to prevent Apache from conducting double-reverse lookups on the domain names.[22]

A.2.11 Authorization

Authorization is the process by which a user's permission to access a particular resource is validated. The criteria by which permissions are granted or denied may vary depending on the resource and the rules associated with that resource.

Different files on a server can be protected in various ways. Each file or directory structure can have a different set of access controls or authentication methods. With the use of directives and the methods of access controls and authentication, many different possibilities exist. For instance, one could require a valid username and password for all access from the Internet, yet no username or password for users that have been authenticated as being located on the organizational intranet. This can be accomplished using the Satisfy directive. The Satisfy Any directive would allow a user access if either the Allow or Deny directives were passed or the Require directive was passed. An example of requiring a password for all access from outside the intranet would be —

```
order deny, allow
deny from all
```

[22] Apache always employs a double-reverse lookup when dealing with host or domain names in all situations related to security. A double-reverse lookup involves translating the host or domain name to an IP address and then translating that IP address back to a list of names. If both translations are not successful, then this will be considered a failure to match the host or domain name.

```
allow from 192.168
AuthType Basic
AuthName “Password Protected”
AuthUserFile /usr/local/Web/apache/passwd/passwords
Require valid-user
Satisfy Any
```

Once it has been determined that a user is from within the intranet, the username and password will not be required. If users from the intranet were required to have a valid usernames and passwords were required, then the Satisfy Any directive (above) would be changed to Satisfy All. This directive would require both directives to be met instead of either one.

If security and privacy are paramount, then the above methods should not be employed. In those instances, Secure Socket Layer (SSL) or its replace Transport Layer Security (TLS) should be used. In most instances, SSL/TLS are the most appropriate authentication and encryption method for public Web servers.

A.2.12 SSL and TLS Authentication

SSL and TLS are the preferred means for authenticating users and encrypting data for Web servers and content. Both are standards that are supported by most Web servers and browsers, and they provide a level of security not attainable with most other Web authentication schemes. SSL/TLS protocols are discussed in detail in Section 6.5, but the particulars of implementing them with Apache are discussed here.

To support SSL, Apache contains the Mod_SSL module that provides strong cryptography using SSL versions 2 and 3 and the newer TLS, version 1. The mod_SSL package, originally created in 1998, is available under a license that allows it to be used at no cost for either commercial or non-commercial uses. Currently, the module provides a strong 128-bit cryptography (for worldwide use) and supports both RSA and Diffie-Hellman ciphers.

Although SSL and TLS are very similar (TLS is based on SSL version 3), some differences exist that may be important for particular applications. SSL is a protocol that provides communications privacy over the Internet. The following are key features that the SSL protocol provides:

- Private connections and encrypted data
- Authentication of peer communicated to server
- Reliable connection.

After an initial handshake, SSL uses the negotiated encryption protocol to generate a secret key. Symmetric cryptography is then used for data encryption. Public key, or asymmetric cryptography, is used to authenticate the peer’s identity and exchange the negotiate symmetric encryption key. Message integrity is provided by the Message Authentication Code (MAC), providing a reliable connection.

TLS, an extension of SSL, is a security protocol designed to provide privacy and data integrity between applications during communication. The key features of the TLS protocol are that it provides the following:

- Private connections
- Reliable connections
- Peer identity authentication
- Secure negotiation of a shared secret
- Reliable negotiation.

A private connection is ensured using symmetric cryptography to encrypt the data. The keys for the symmetric cryptography, which are generated uniquely for each connection, are based on a secret negotiated by the TLS protocol.

The connection is ensured for reliability using a keyed MAC, whereas the identity of the peer is ensured through authentication using asymmetric, or public key, cryptography.

The negotiation of a shared secret is secure from sniffing and interception from any user able to place himself or herself in the middle of the transfer of the data (e.g., it eliminates the "man-in-the-middle attack"). Any modification of the data in a secure communication is detectable by the peers communicating, ensuring a reliable negotiation.

A.2.13 Restrict Remote Operations PUT and POST

Remote users should not be able to upload files to the DocumentRoot directory structure. This effort is usually accomplished through the PUT and POST commands. This can be prevented using the file and directory permissions of the DocumentRoot. Failure to do this may allow malicious entities to deface or otherwise compromise the Web site.

A.3 Patching Apache

The greatest risk posed in a hardened Web server is over ports 80 and 443. The attacks possible are as follows:

- URL-sending metacharacters, long URLs, and vast amounts of data
- Application-sending metacharacter and vast amount of data
- SSL data causing unexpected action.

Web vulnerabilities can and should be scanned for using a security-scanning program, such as CyberCop, Nessus, and Whisker, to test for known Web vulnerabilities. The Web server administrator should be sure to keep up-to-date on the latest vulnerabilities for all applications and operating systems that are used with or on the Web server. For more information about research and applying security patches, see NIST Special Publication 800-40, *Applying Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

A.4 Maintaining a Secure Apache Configuration

A.4.1 Check Web Server Logs

Web server logs, in addition to the operating system logs, should be checked daily for any data that could alert to a potential problem. Items to look for might include those below:

- Invalid login attempts
- Attempts to access restricted files
- Attempts to access files that do not exist
- Attempts to PUT files to the server when not allowed
- Multiple attempts from similar IP addresses in small amount of time
- Unexpected stops or starts of the Web server.

From more information about logs, see Section 6.

A.4.2 Archive and Flush Web Server Logs Periodically

To prevent the disk from filling up and to keep the log files manageable, the Web Server log files should be archived and flushed periodically. Transfer of the logs can be automated by piping the log file into a log file rotation utility. A utility such as rotatelogs, which can be found in the /src/support/ directory, is useful in rotating the logs daily.

Another useful, and a bit more advanced, program for log file rotation is cronolog, available at http://www.ford-mason.co.uk/resources/cron0log. Cronolog is a program that reads log messages from its input and writes them to a set of output file constructed using a template and the current date and time. Cronolog was designed for use with a Web server such as Apache to split the continuous logs into daily logs.

A.4.3 Perform Regular Backups of System Data

Anytime the Web administrator edits or makes changes to any of the system configuration files, he or she should first make a backup, especially when multiple modifications are made. This ensures that if a misconfiguration occurs that renders the server inoperable, the previous working state can be restored in a timely manner.

Tape backups of the entire Web server should also be made regularly. There should be a daily and weekly schedule for partial and full server backups. Both full and incremental tape backups should occur within this regular schedule.

Test all backup and recovery plans that are developed. Ensure that these plans are successful before an actual problem or outage occurs. Any changes to the plan should be made and then retested.

A.4.4 Maintaining the Secure Configuration

Once the server, operating system, and network are secured, they should be maintained and updated to ensure that they remain secure, by monitoring and implementing security updates available from many organizations, including Apache, which maintain vulnerability lists and patches.[23]

In addition to monitoring and implementing security and vulnerability updates, vulnerability scanning and assessment and other security testing should also be implemented to ensure that new issues have not been introduced to any part of the system without the knowledge of the administrator.[24]

A.4.5 Apache Security Checklist

Completed	Action
	Installing and securing the host operating system
☐	Apply latest patches to operating system
☐	Disable or remove all unnecessary services and applications
☐	Apply appropriate permissions to system resources
☐	Use appropriately strong identification and authentication mechanisms
	Installing the Apache Web server software
☐	Install Apache daemon and Web server content on separate hard drive partitions
☐	Web content should be stored in the DocumentRoot
☐	DocumentRoot and htdocs should be installed in separate directories
☐	Remove all Apache and other vendor documentation from DcoumentRoot and htdocs directories
☐	Install Apache in a chroot jail
☐	Do not install or remove src directory included in the Apache distribution
	Set permission for the Web server directories and files
☐	Ensure that all commands executed by root are protected from non-root users
☐	Create an unprivileged user account for the Apache Web server
	Remove all unnecessary scripts, executables, and services
☐	Disable or remove unnecessary services and applications
☐	Remove all sample content

[23] From more information on these resources, see NIST Special Publication 800-40, *Applying Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

[24] For more information on Security Testing, see NIST Special Publication 800-42, *Guideline on Network Security Testing* ((http://csrc.nist.gov/publications/nistpubs/index.html).

Completed	Action
☐	Remove all unnecessary scripts and executables (e.g., CGI, PHP, etc.)
☐	Delete all unnecessary files from the HTML document tree
☐	Prevent users from creating .htacess files
	Initial configuration
☐	Make working copies of the server configuration files
☐	Disable automatic directory listing
☐	Disable symbolic links
☐	Disable server side includes
☐	Enable logging
☐	Configure user identification and authentication as appropriate
☐	Restrict remote operations put and post
	Maintaining a secure Apache configuration
☐	Apply all necessary patches to the Apache application (check for new patches on (at least) a weekly basis)
☐	Employ vulnerability scanners to test Apache install, underlying operating system and network defenses (when installed and at least quarterly thereafter)
☐	Monitor logs on a daily basis
☐	Archive and flush Web server logs periodically
☐	Perform regular backups of system data and Web server content

Appendix B. Securing Internet Information Server

Appendix B discusses specific procedures, techniques, and methods for securing Microsoft Internet Information Server (IIS). Although the information within this section is designed to provide security best practices for IIS, because IIS is an integrated service with the Microsoft Windows operating system, no discussion of IIS security is complete without mentioning the relevant operating system security configurations. Because this document cannot provide a complete set of configurations for security of the Microsoft Windows operating system, readers should consult the appropriate operating system hardening procedures (see Section 3.4). Note that this section addresses only IIS 4.0 and 5.0. Versions of IIS prior to version 4.0 are not recommend for use on public Web server and should be upgraded immediately.

Administrators who do not wish to audit and configure IIS manually can now use Microsoft IIS Lockdown Tool to audit and configure IIS without having to check and change each setting manually. All administrators should consider running this application on their IIS Web server because it offers access to a variety of configuration settings that are difficult or impossible to change without use of the tool (e.g. changing the Hypertext Transfer Protocol [HTTP] and File Transfer Protocol [FTP] application banners). In addition, versions of the IIS Lockdown Tool after version 2.0 include URLScan. This tool screens all incoming requests to the IIS server and filters them based on rules set by the administrator (with assistance from the Lockdown Tool). This secures the server by ensuring that only valid requests are processed.

URLScan is effective in protecting Web servers because most attacks share a common characteristic—they involve the use of a request that is unusual in some way. For instance, the request might be extremely long, request an unusual action, be encoded using an alternate character set, or include character sequences that are rarely seen in legitimate requests. By filtering out all unusual requests, URLScan prevents them from reaching the server and potentially causing damage.[25]

B.1 IIS Overview

IIS is a combination of services available for Microsoft Windows NT, Windows 2000, and Windows XP providing capabilities such as HTTP and FTP. IIS is considered integrated into the Microsoft Windows server architecture executing as an enterprise service. IIS has access to capabilities provided by other components of the Windows operating system and simultaneously uses a number of the technologies inherent to the Windows architecture to function, such as the Microsoft proprietary technology, component object model (COM). Consequently, any security vulnerability that is discovered in Microsoft Windows also should be considered a vulnerability with IIS.

IIS introduced numerous new and extended technologies for Web servers such as Internet Server Application Program Interface (ISAPI) applications and Active Server Pages (ASP). IIS also serves as a basis for other Microsoft technologies, such as Site Server, and for Windows XP, Microsoft SharePoint. All these technologies must be considered when securing IIS.

[25] The IIS Lockdown Tool/URLScan application can be downloaded from Microsoft at http://www.microsoft.com/Downloads/Release.asp?ReleaseID=33961.

Web services are also available to Windows 9x clients through Microsoft Peer Web Services (PWS). PWS uses the Microsoft Winsock network programming library. This section does not focus at all on PWS security.

IIS is administered with the Internet Services Manager (ISM). This is a Microsoft Management Console (MMC) Snap-in that is accessible from the IIS menu option on the Option Pack menu in Windows NT 4.0 and from the Administrative Tools option on the Start menu in Windows 2000. From here, administrators can selectively start and stop Web and FTP sites and can configure advanced options for Web sites, even particular folders for a Web site.

B.2 IIS Overview

IIS is included by default in the installation of Windows NT 4.0 Workstation and Server, Windows 2000 Professional and Server, and Windows XP, but administrators may choose against its installation during this time and return to install it later. The process of installing IIS after the operating system is the same process a user would follow to add a component of Windows from the operating system compact disc (CD). Windows NT 4.0 attempts to install IIS version 3.0 during the operating system installation,[26] whereas the Windows NT Option Pack provides IIS 4.0 with additional components. Administrators who already installed IIS on their Windows NT machine can perform an upgrade to IIS 4.0 when installing the Windows NT Option Pack (NTOP) 4.0. For upgrade tips, consult the following Microsoft Knowledge Base article Q224831:

http://support.microsoft.com/support/kb/articles/Q224/8/31.asp

Windows 2000 attempts to install IIS 5.0 with the operating system and Windows XP attempts to install IIS 5.1 by default, although the exact version of IIS available for Windows XP is not completely clear.[27] Microsoft indicates within the ISM that the version of IIS for Windows XP is actually 6.0. By default, IIS 5.0 installs the following components:[28]

- Required IIS common files
- IIS documentation
- FTP Server
- FrontPage 2000 Server Extensions
- Internet Information Services snap-in
- Internet Services Manager (HTTP)
- World Wide Web server (WWW).

[26] Note that it is not recommended to use IIS prior to version 4.0 because of a number of vulnerabilities. It is also not recommended that one install later versions of IIS over previous a version, because a number of vulnerabilities that existed on the previous version will remain even with the upgrade.

[27] http://www.activewin.com/reviews/previews/windowsxp/server/iis.shtml

[28] Ivens, Kathy, and Gardinier, Kenton. The Complete Reference – Windows 2000

The minimum installation for IIS 4.0 and the NTOP 4.0 includes the following components:[29]

- Internet Information Server 4.0
- Internet Service Manager (MMC snap-in version)
- Microsoft ASPs
- Microsoft Management Console
- MMC:
  - ActiveX Data Objects with Remote Data Service
  - Remote Data Service
  - Open Database Connectivity (ODBC)
  - Object Linking and Embedding Database (OLE-DB).

Windows NT 4.0 administrators may consult the following location for additional tips on installing IIS 4.0:

http://www.iisanswers.com/Top10FAQ/t10-installiis.htm

Windows 2000 administrators may consult the following Microsoft Knowledge Base article (Q266115) for additional tips on installing and using IIS 5.0:

http://support.microsoft.com/support/kb/articles/Q266/1/15.ASP

B.3 Where to Install IIS

Any administrators installing IIS within their infrastructure should consider where the server will reside physically. The option of which physical server it is installed on and where it resides on a network affects the performance and security. Ideally, IIS should be installed on a machine whose only function is a dedicated Web server. Do not install IIS on any domain controller, primary (PDC) or backup (BDC); if possible, do not install IIS on a server that already has another dedicated function, such as electronic mail or database services. A Windows domain controller is constantly processing authentication requests. Running IIS on a domain controller can decrease overall performance. In addition, installing IIS on these machines puts their contents and hardware at risk from the outside world through unauthorized access to sensitive information or corrupting critical information stores.

Both IIS 4.0 and 5.0 include options to host multiple Web sites with a single network interface card (NIC) and IP address. These options allow organizations to consolidate resources while providing the same functionality.

[29] http://www.windowsitlibrary.com/Content/405/14/2.html

If possible, do not make IIS a member of a Windows domain. No accessible way should exist for an attacker to springboard across platforms within an enterprise. Capabilities such as domain trusts and cached login credentials that exist on domain members can afford an attacker this chance.

It is a common practice to insert all publicly accessible machines like an IIS Web server on a network demilitarized zone (DMZ). This is designed to isolate the resources on this machine from any other machine in the enterprise. This design assumes that organizations have implemented some type of firewall and packet filtering technology at their network perimeter (for more information about locating a Web server within a network, see Section 7).

B.4 Configuring Windows Operating System

The first step in securing IIS is addressing operating system security. The information presented in this section is designed to address security configurations that are common to all versions of Microsoft Windows that run IIS. Any differences among the versions are noted where applicable. Securing the underlying operating system is a basis for a high-security Web server. The following steps should be taken in this area:

- Implement physical security for computer
- Update operating system
- Apply security templates (if applicable)
- Restrict anonymous registry access
- Remove and disable unnecessary services
- Limit user accounts
- Harden the file system.

As stated previously, although the focus of this discussion on IIS security includes operating system security, this document cannot provide all the configurations necessary to create a secure operating system environment (see additional resources provided in Appendix A). The references presented in the introduction provide additional locations that readers can consult for more information. This section describes some of the areas of the Windows operating system that require additional configuration for security when running IIS.

B.4.1 Protect Server Environment

Ensure that the server is physically protected from unauthorized persons and environmental extremes. Examples of this type of environment are as follows:

- Locked cabinets
- Access controlled data centers
- Environmentally controlled area.

The criticality of the data and services provided by the servers will affect the nature and types of physical protections required.

B.4.2 Remove Unnecessary Hardware

Remove or disable all unnecessary hardware from the machine. Examples are as follows:

- 3D accelerators
- Sound cards
- Compact Disc Read Only Memory (CD-ROM) and floppy drives.

Components that may be necessary to install the operating system, such as a CD-ROM drive, should be removed once the installation is completed. Be sure to follow the hardware uninstallation process that Windows provides because simply removing the computer hardware without deleting associated files and settings can result in instability in the operating system. Removing unnecessary hardware improves reliability and security. For example, many programs allow a user to boot up a machine from a floppy and bypass most operating system-based security mechanisms.

B.4.3 Update Operating System

Although many administrators consider this to be a one-step process, updating or patching is a crucial step in maintaining the security of the operating system and is a continuous process. Vulnerabilities and bugs in software such as operating systems are discovered on a daily basis in some cases. Unfortunately, there is no single place to turn to receive all necessary patches and updates for an operating system. For more information on patching, consult.[30]

Microsoft Windows administrators should consult several sources to obtain updates for their operating system. Service packs contain orderable or downloadable updates for Microsoft OSs that fix existing problems and, occasionally, product enhancements. Windows administrators should always consider upgrading their operating systems to the latest service pack available from Microsoft. Service packs for most versions of Windows are available from http://windowsupdate.microsoft.com

In addition, Microsoft frequently creates hotfixes to mitigate vulnerabilities discovered in their product line. A hotfix (also referred to as a quick fix engineering [QFE]) is code that fixes a bug in a particular product. Product users are typically notified or can obtain information about current hotfixes at Microsoft's Web site and download the hotfixes they should apply. Although service packs contain all of the latest hotfixes available at the time of their release, new hotfixes are often created between service pack releases. Hotfixes can be downloaded from http://www.microsoft.com/security/

It is important to thoroughly read all supporting documentation included with hotfixes and service packs. This information often describes situations or particular hardware configurations in which the service pack or hotfix will neither work nor cause instability.

---

[30] NIST Special Publication 800-40, *Applying Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

B.4.4 Apply Security Templates

Windows NT provided the Security Configuration Editor (secedit) with Service Pack 4.0 (and later) and Windows 2000 provides with all versions. Secedit allows administrators to do the following:

- Define a template of security configuration settings
- Compare the local machine's settings against a template
- Configure the local machine's settings to match a template.

Microsoft provides a security template, named *hisecWeb.inf*, to establish a baseline of security that is applicable to most Web site installations. The template can be obtained from http://download.microsoft.com/win2000srv/SCM/1.0/NT5/EN-US/hisecWeb.exe

Security templates created with secedit, although extremely useful, do not eliminate the requirement to take the other steps enumerated in this section. Web administrators should consider applying security templates in tandem with other options presented here. Be sure to examine any security template before applying them. Certain configurations might need to be updated for the needs of a particular organization.

B.4.5 Restrict Anonymous Registry Access

By default, the Windows registry is set with relatively open access permissions. To restrict anonymous users from connecting to the Windows registry and performing information-gathering tasks, such as enumerating account names and accessing security identifier (SID) information, add the following key to the Windows registry:

HKLM\System\CurrentControlSet\Control\LSA\RestrictAnonymous=2

This may affect certain network services that require this access to function correctly. If administrators notice their computers behaving incorrectly after adding this key to the registry, they should first ensure that all machines are running with valid user accounts. If this does not work, they should add "winreg" to the "NullPipeSessions" value in

HKLM\System\CurrentControlSet\Services\LanmanServer\Parameters

B.4.6 Remove/Disable Unnecessary Services

Depending on the role that a Windows server will fulfill, not all services that are installed by default are necessary for the server to function. It is considered good practice to limit the number of entry points (services) into a server. Any services that are not required for IIS to function should be disabled. Table B.1 lists those services that are required for IIS to function.

Table B.1: IIS Required Services

Service Name
Event Log
License Logging Service

Service Name
Windows NT Lanman (NTLM) Security Support Provider
Remote Procedure Call (RPC) Service
Windows NT Server or Windows NT Workstation
IIS Admin Service
Microsoft Distributed Transaction Coordinator (MSDTC)
World Wide Web Publishing Service
Protected Storage

Administrators can use the Service Configuration Manager on the Windows Control Panel to stop and disable unnecessary services. The actual services that administrators can successfully disable may vary with each system. For example, if IIS has been installed on a system with a preexisting function (not recommended) within an infrastructure such as e-mail, then those services must also be preserved. The Windows NT and Windows 2000 Service Configuration Manager provides a Service Dependency option for each service. Administrators can use this option to determine if a particular service is dependent on any other services to run. The Service Dependency option is a valuable method to double-check a service that an administrator wishes to disable.

B.4.7 Limit User Accounts

When IIS is installed, it creates two anonymous accounts: *IUSR_computername* and IWAM_*computername*, where *computername* represents the name of the computer on which IIS was installed. The IUSR_*computername* account is used by IIS to grant anonymous access to Web resources. The IWAM_*computername* is used by the Microsoft Transaction Server (MTS) and various IIS entities to provide programmatic and transactional functions.

B.4.8 Set Account Logon to Local

The anonymous IUSR_*computername* account should have logon locally set correctly. Because this account interacts with the IIS service, it acts as if it were physically logged into the server. Administrators can specify an alternate account to use with anonymous access. If an administrator wishes to use an account other than the IUSR_*computername* account, the administrator only needs to ensure that this user is permitted to logon locally.

B.4.9 Harden File System

The processing of hardening the file system includes several steps. Administrators first must decide the file system to use on the server. During the Windows installation process, the administrator must choose the file system with which to format all hard drives. Administrators have an option of choosing between the File Allocation Table (FAT) or NT File System (NTFS). The NTFS file system provides better security, performance, and logging; therefore, it is considered the best file system to use with Windows-based Web servers.

B.4.10 Convert Non-NTFS File System

Administrators who previously formatted their drives with the FAT file system can switch to NTFS using the convert command. The syntax of the convert command is as follows (this example assumes that the WINNT directory has been installed on a partition labeled C):

```
C:\>convert C: /FS:NTFS
```

Once a drive or parition has been converted to NTFS using the CONVERT utility, it cannot be converted back using this utility. If this is required for some reason, third-party utilities exist that will provide that functionality.

If a Web site uses executable files such as active content scripts (e.g., JScript, VBScript) or even CGI programs, place these programs in a separate directory. This action allows more flexibility in assigning access permissions and auditing access. It also prevents Web administrator and Webmasters from having to set ACLs for each file within the Web application. For more information about active content, see Section 5.2.

B.4.11 Synchronize NTFS Permissions With IIS Permissions

For installs of IIS that employ the NTFS, administrators should verify two sets of permissions for file system resources. IIS has Web permissions, and NTFS has its own permissions. IIS permissions really control which HTTP commands are allowed to be executed for HTTP resources. NTFS permissions control which user accounts have what type of access to resources on the hard disk.[31]

When a user attempts to access a protected resource from IIS, both sets of permissions are compared with the credentials the user supplies. If a discrepancy exists between NTFS permissions and IIS permissions, the more restrictive of the two is used.

Those permissions that an administrator configures within IIS should match the permission specified by NTFS (operating system) for the same resource. If not, the resource may be more or less accessible to Web site users than the administrator anticipates or desires (see Section 4.2 for more information).

B.4.12 Secure Default Windows Repair Directory

The Windows Repair directory is located in the WINNT system root folder on the system partition. This folder contains a backup of the SAM password file that is created on installation. By default, this folder gives the Everyone group full control, which would allow any user to obtain a copy of the SAM password file and possibly compromise the host. Administrators should delete this copy of the SAM password file and change the permission of the directory so that only the appropriate accounts have access.

B.5 Configuring IIS

This section presents the configuration options for IIS specifically. These actions describe in detail the basic steps necessary to secure an IIS Web server. Although most of the information

[31] http://msdn.microsoft.com/library/en-us/dnsecure/html/WebsecIISsec.asp?frame=true

that is presented in this section applies to all versions of IIS, any difference among versions is noted as applicable.

B.5.1 Configuring SSL Usage

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) are methods used to cryptographically secure data while in transit between a Web browser and Web server (see Section 6.5). If there is any likelihood that IIS will be handling sensitive information such as user passwords or even credit card information, SSL/TLS should be implemented.

B.5.1.1 Create Certificate Signing Request

To enable IIS to use SSL, administrators must first create a CSR. A CSR is sent to a certificate authority (CA), which processes the request and returns an X.509 server certificate that can be used to create SSL/TLS sessions.

Once the CSR has been created, administrators must choose a CA to submit their CSR. Many organizations have their own CA or may choose to use one of the commercial or federal CAs that issue certificates. The choice of CA will be based on the policies of each organization.

B.5.1.2 Import Server Certificate into IIS

Once the CSR has been created and submitted as described above, the CSR is processed and a server certificate is created. Following these steps, administrators must obtain and import the certificate into IIS. This action allows the Web site(s) that IIS is hosting to employ SSL/TLS capabilities.

B.5.2 Enable Logging

For IIS, the default log file is located in the c:\winntsystem32logfilesw3svc1 directory, and the log file name is based on the current date, as in yymmdd.log. A new log file is generated daily. The default format is the W3C Extended Log File Format, a standard that many third-party utilities can interpret and parse.

Logging is important to maintaining IIS security. It provides an accurate picture of the usage of the Web server and a breakdown of what files were accessed and by whom. Logging also allows the Web administrator to determine if their server is being attacked or probed. For IIS, the recommended format is to use the W3C Extended Log File. Table B.2 lists the recommended fields to log in each IIS log file.

Table B.2: Recommended Fields to Log for IIS

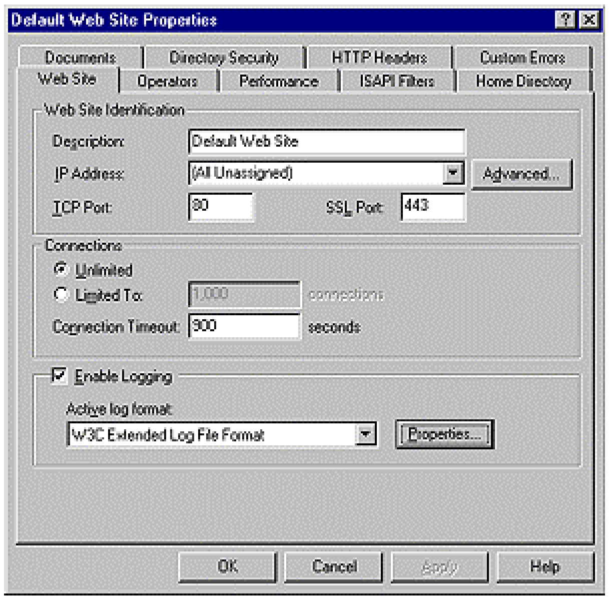
Field Name
Client IP Address
User Name
Method
URI Stem
HTTP Status
Win32 Error

Field Name
User Agent
Server IP Address
Server Port

Administrators should also consider storing copies of their log files offline as a standard backup procedure so that they can be used if the logs on the Web server are compromised. For more information about logging, see Section 6.

B.5.3 IIS Log Options

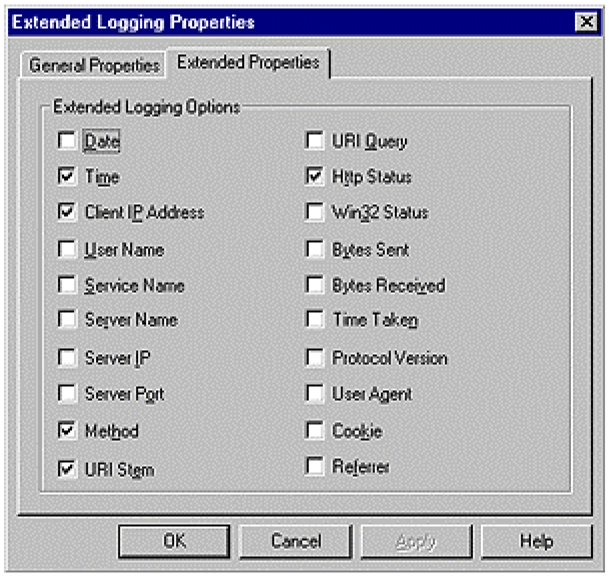
By default, Apache logs are more thorough; however, the IIS logs can be improved to match those of Apache. The following simple procedures will demonstrate how to add these additional logging capabilities to IIS. From the IIS management console, open the Web Site Properties dialog box (Figure B.1).



Description: Default Web Site
IP Address: (All Unassigned)
TCP Port: 80
SSL Port: 443
[✓ Unlimited]
Limited To: 1,000 connections
Connection Timeout: 900 seconds
Active log format: W3C Extended Log File Format

Figure B.1: Default Web Site Properties

From the Default Web Site Properties window, click on the Properties button under the Enabling Logging section. This action will open the Extended Logging Properties Window shown in Figure B.2. This allows a Web administrator to add greater detail to the log files.



Extended Logging Options

Date	URI Query
[✓ Time]	[✓ Http Status]
[✓ Client IP Address]	Win32 Status
User Name	Bytes Sent
Service Name	Bytes Received
Server Name	Time Taken
Server IP	Protocol Version
Server Port	User Agent
[✓ Method]	Cookie
[✓ URI Stem]	Referer

Figure B.2: Default Web Site Properties

B.5.4 Set Log File ACLs

Any log file that IIS generates will be stored in the %SystemRoot%\system32\LogFiles folder on the system partition. If the access control settings for these files are not secured, an attacker may cover up their actions on the server by deleting associated log entries. Table B.3 lists the recommended access control settings for IIS logs.

Table B.3: IIS Log File ACL Settings

User	ACL Setting
Administrator	Full Control
System	Full Control
Everyone	Read, Write, Create

B.5.5 Remove All Sample Applications

IIS installs with a set of sample applications that demonstrate its functionality. Sample code should never exist on a production server. This can be considered to include the software development kit documentation and all sample scripts. Table B.4 lists the default locations for some of the IIS samples. The notation <drive letter> corresponds to the partition or drive where the named folders reside.

Table B.4: IIS Sample Resources to Remove

Technology	Location
IIS	<drive letter>:\Inetpub\iissamples
IIS SDK	<drive letter>:\Inetpub\iissamples\sdk
Administration Scripts	<drive letter>:\Inetpub\AdminScripts

Data Access	<drive letter>:\Program Files\Common Files\System\msadc\Samples

B.5.6 Remove IISADMPWD Virtual Directory

This directory is included by default with IIS. It allows users to reset Windows NT passwords. This type of setup was never intended for public Web sites. This directory should be removed if the IIS server is to be accessible from the Internet.[32]

B.5.7 Remove Unused Script Mappings

IIS can be configured to respond to requests for ASPs or server parsed HTML (SHTML) with various types of active content. When IIS receives a request for one of these types of files, if an associated script mapping with the file, IIS passes the request to the dynamic link library (DLL) that houses that functionality.

If any script mapping within IIS is not being used, it should be removed immediately. Table B.5 provides script mappings that may be removed.

Table B.5: IIS Script Mappings

If IIS Does Not Require	Remove the Mapping
Web-based password reset	.htr
Index Server	.ida
Internet Database Connector	.idc
Server-side includes	.shtm, .stm, .shtml

Note that this list excludes all possible IIS script mappings. Any custom mapping that was implemented should also be removed if it is no longer used.

B.5.8 Disable Parent Paths

Parent paths allow users to use '..' when browsing directories, MapPaths, and the like. Having parent path enabled, allows files in the parent directories to be used. The unfortunate effect is that this can afford an attacker access to files outside the IIS Web root. Several attacks use parents' paths in conjunction with other processes to obtain access to files such as the backup SAM password file and even access to cmd.exe (which allows an attacker execute commands and code on the Web server). Parent paths should be disabled unless absolutely necessary.[33]

[32] Refer to the Microsoft Knowledge Base article Q184619 for more information: http://support.microsoft.com/support/kb/articles/q184/6/19.asp

[33] IIS administrators can find instructions on how to disable parent paths at http://www.windows2000faq.com/Articles/Index.cfm?ArticleID=13996.

B.5.9 Disable Internet Printing Protocol Support

Cited by SANS as one of the five most widely exploited holes in unpatched versions of IIS in 2001,[34] Windows 2000 includes support for the Internet Printing Protocol (IPP) via an ISAPI extension on IIS 5.x. This extension is installed by default on all Windows 2000 systems with IIS.

CERT published an advisory (also referenced by Mitre's CVE system) in May 2001 indicating that through a buffer overflow in the ISAPI extension, remote users could execute arbitrary code in the local system context (essentially the equivalent of administrator), giving the user complete control of the system.[35]

IPP can be disabled by adding the following key to the registry:

HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Printers\DisableWebPrinting

The type of the key is REG_DWORD, and the value should be set to 1. Administrators should note that this effort could be accomplished with a security template as described above.

B.5.10 Disable RDS Support

Remote Data Services (RDS) has the potential to make IIS vulnerable to denial of service (DoS) and arbitrary code execution attacks. Either remove the RDS capability from the IIS server or restrict its capabilities using ACLs. Administrators may refer to the Microsoft references MS98-004, MS99-025, and Q184375 for more information:

http://www.microsoft.com/technet/security/bulletin/MS98-004.asp

http://www.microsoft.com/technet/security/bulletin/MS99-025.asp

http://support.microsoft.com/support/kb/articles/q184/3/75.asp

In addition, check the IIS logs frequently for signs of the attack. A log entry that records an instance of the RDS attack looks similar to:

```
1999-10-24 20:38:12 – POST /msadc/msadc.dll …
```

B.5.11 Set Authentication Methods

Web-based authentication is specific to an application. The only requirement that is common across all forms of authentication is the need to have "strong enough" authentication for the application and data it contains. The following authentication methods are supported by IIS in order of increasing complexity and trust:

---

[34] http://www.sans.org/infosecFAQ/win2000/5threats.htm

[35] The full text of the CERT advisory can be found at http://www.cert.org/advisories/CA-2001-10.html.. The full text of the related Microsoft advisory can be found at http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/bulletin/MS01-023.asp.

- Anonymous. This authentication method was discussed in Section 3.2.3. IIS uses the *IUSR_computername* account to interact with Windows during anonymous authentication.
- Basic. Basic authentication is the only authentication method that is fully supported by all major browser vendors and versions. This method uses clear text credentials and is not secure (see Section 6.3 for more information).
- Windows NT Challenge and Response. Challenge and Response is the newest form of authentication in which no password information is actually transmitted across a network. Although several brands of Web servers support this method, not all major browser versions do.
- Client Certificates. This is the most complex and trustworthy form of authentication. It requires users to have obtained and installed an X.509 certificate into their Web browser. This method has been slow to gain acceptance because of the overhead required by this method.

For more information, consult the following Microsoft Knowledge Base article Q229694:

http://support.microsoft.com/support/kb/articles/Q229/6/94.ASP.

B.5.12 Set Appropriate Directory Permissions

Separating files of a similar type or function into their own directories allows for more granular control over a Web site or Web application. Table B.6 lists recommended ACL settings for various types of Web content.

Table B.6: Recommended Directory Permissions for Web Content

File Type	Recommended ACLs	
	User Group	**ACL Setting**
CGI etc .EXE, .DLL, .CMD, .PL	Everyone	X
	Administrators	Full Control
	System	Full Control
Script files .ASP, etc	Everyone	X
	Administrators	Full Control
	System	Full Control
Include files .INC, .SHTML, .SHTM	Everyone	X
	Administrators	Full Control
	System	Full Control
Static Content .HTML, .GIF, .JPEG	Everyone	R
	Administrators	Full Control
	System	Full Control

The settings above are application dependent. Web administrators should review their environment and requirements before altering the ACLs for their Web sites.

B.6 Configuring Active Server Pages

This section discusses ASP security and how it relates and affects IIS. Web administrators that do not implement ASP over IIS can disregard the information provided here.

ASP is a form of active content created by Microsoft. Because of the complexity of ASP, this section cannot completely cover all aspects that exist concerning ASP. This section is merely meant to show the relationship that ASP has with IIS security and to present some tips to ensure this security stays sound.[36]

ASP security is based on two factors: data integrity and bounds checking. Data integrity refers to protecting the information that ASP requires to operate. This information can be sensitive to the user or organization because it contains proprietary data such as custom-purchased code, or even machine settings that work with database servers. Protecting IIS involves ensuring that all data integrity is maintained. Bounds checking refers to ASP or IIS validating the input that a user supplied to an ASP component in an HTTP POST operation.

B.6.1 Data Integrity

The content provided by ASP is static when an HTTP request returns to a client. The capabilities that were required to render dynamic information can sometimes include custom-purchased code, file names and system configuration settings, and hard-coded user names and passwords. ASP users must ensure that all of the dynamic ASP tags and information are abstracted away from Internet users. Two specific examples of information that must be protected are as follows: the Global.asa ASP configuration file and the Showcode.asp file.

B.6.2 Protecting Global.asa

The Global.asa file is an optional file in which users can specify event scripts and declare session and application objects that can be accessed by every page in an ASP application. The Global.asa file should be stored in the root directory of the ASP application, and each application can have only one Global.asa file. Any ASP pages that use database connectivity may use Global.asa to store identification information, such as ODBC names and database authentication information. Administrators should review Global.asa for any information such as this and remove it from this file.

B.6.3 Protecting Showcode.asp

IIS 4.0 administrators are vulnerable to a serious loss of integrity from Showcode.asp. The Showcode.asp is a sample script included with the default install of IIS 4.0 that is designed to view the source code of the applications via a Web browser. Unfortunately, this file performs inadequate security checking and allows anyone with a Web browser to view the file contents on the Web server. The following Microsoft Knowledge Base article (Q232449) describes the vulnerability:

http://support.microsoft.com/support/kb/articles/Q232/4/49.ASP

[36] For more information about ASP and other active content, see NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code* (http://csrc.nist.gov/publications/nistpubs/index.html).

Administrators of IIS should remove this file immediately from their servers when they remove all sample IIS applications.

B.6.4 Implement Proper Bounds Checking

The second CERT advisory for 2000 described the threat that "malicious html tags embedded in client requests" could have on Web servers.

http://www.cert.org/advisories/CA-2000-02.html

Active Content technologies like ASP are especially vulnerable to this threat. A number of attacks exist where user input is treated incorrectly as valid input and the user could gain access to the server or cause damage. Proper text checking can be performed with either JavaScript (Microsoft refers to it as JScript) and VBScript regular expression capabilities.

The following sample code will strip a string of all invalid characters (characters that are not 0-9a-zA-Z or _): [37]

```
Set reg = New RegExp
reg.Pattern = "\W+" ' One or more characters which
' are NOT 0-9a-zA-Z or '_'
strUnTainted = reg.Replace(strTainted, "")
```

The following sample will strip all text after a | operator:

```
Set reg = New RegExp
reg.Pattern = "^(.+)\|(.+)" ' Any character from the start of
' the string to a | character.
strUnTainted = reg.Replace(strTainted, "$1")
```

Care also should be taken when opening or creating files by using Scripting File System Object. If the filename is based on the user's input, the user may attempt to open a serial port or printer. The following JScript code will strip out invalid filenames:

```
var strOut = strIn.replace(/(AUX|PRN|NUL|COM\d|LPT\d)+\s*$/i,"");
```

B.7 Patching IIS

Microsoft provides several ways to distribute patches and updates for IIS, and has even automated part of the process in one of its tools, the network hotfix checker. This does not diminish the requirement to be constantly aware of new vulnerabilities and patches. Administrators should subscribe to event notification services such as the service provided by Microsoft. Not only do patches provide new functionality and bug fixes but also they contain

[37] All sample code taken from http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/iis5chk.asp

serious security updates. The requirement for administrators to constantly check to ensure that there are no new patches to apply is absolutely crucial.[38]

B.8 Maintaining Secure IIS Configuration

Once a Web server is made more secure, the Web administrator must maintain that security through constant vigilance. Numerous activities must be conducted periodically and in a timely manner to ensure the security of a Web server. The administrator must learn to think and act proactively regarding security.

B.8.1 Review and Back Up IIS Log Files and Windows Event Log Files

IIS should be configured to log the actions of users who visit its Web sites. The steps to implement logging were presented in Section 6. Administrators should develop a habit of reviewing IIS logs daily to search for anomalies. The log files should also be backed up regularly. Depending on the traffic, log files should be backed up daily; for other less frequented sites, the logs may be backed up weekly or even monthly.

IIS includes the ability to log to an ODBC database connection. This provides considerable flexibility for users because they can use the database that IIS creates to perform custom analysis of the log data using software like Microsoft Access or Microsoft Excel.

Again, because IIS is an integrated service with Windows, administrators should also examine the log files for Microsoft Windows. For Web sites that use some type of authentication, failed login attempts will be recorded in the Windows event log.

B.8.2 Review SAM User Database and User Accounts

Administrators should review regularly the user accounts that the SAM database contains on their IIS server. Any unusual activity or account changes should be addressed immediately because once a malicious user can gain unauthorized authenticated access to IIS, the entire computer's integrity is at risk.

Specifically, administrators should note the presence of any new accounts created or major changes in existing accounts. Failed login attempts on user accounts will be noted in the Windows Event log as described above.

B.8.3 Check for Patches and Updates Regularly

The importance of patching and updating IIS and Microsoft Windows cannot be emphasized enough. The section above regarding patching IIS and Windows described several methods to check for any possible updates to IIS and Windows.

Recently, several widespread outbreaks of computer viruses and malicious code that targeted IIS servers could have been avoided had administrators been patching their IIS installations regularly. Depending on the criticality level of IIS within a user's enterprise, the check for updates should be made at least once a week.

---

[38] The importance of patching for software is elaborated in NIST Special Publication 800-40, *Applying Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

As stated in the updating section, the automated tools that Microsoft has developed can ease the tasks of those responsible for IIS.

B.8.4 Back Up IIS Metabase

IIS stored its configuration settings in the IIS Metabase. The Metabase is analogous to the Windows registry, but is specific to IIS. If for some reason IIS needs to be reinstalled or reconfigured, these settings can be used to resurrect IIS in a timely manner.

The backup of the IIS settings should be kept in a secure area inaccessible by all but authorized administrators. IIS 5.0 administrators can consult the following Microsoft Knowledge Base article (Q300672) for more information:

http://support.microsoft.com/support/kb/articles/Q300/6/72.ASP

IIS 4.0 administrators can consult the following Microsoft Knowledge Base article (Q240941) for more information:

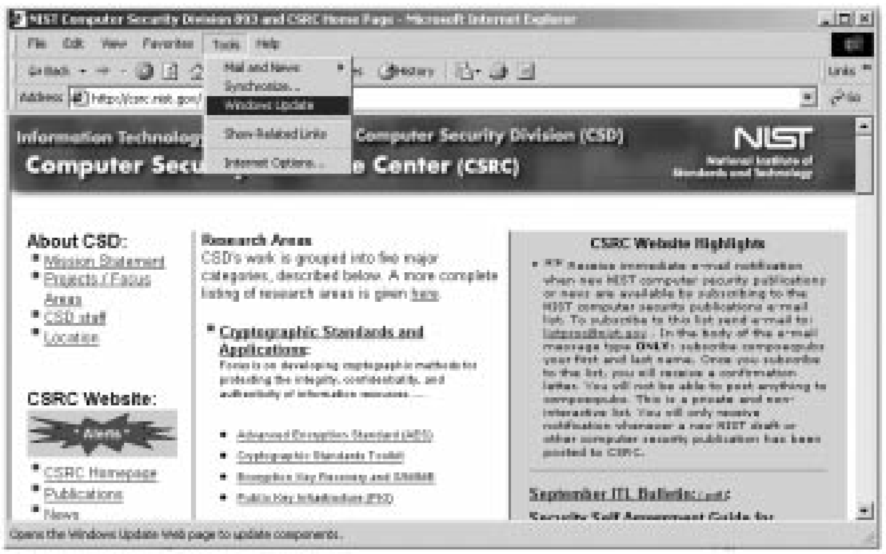
http://support.microsoft.com/support/kb/articles/Q240/9/41.ASP

B.9 Microsoft Patching Tools

Microsoft offers several free tools that can assist a Web administrator in checking and maintaining a secure Web Server configuration. Microsoft Windows Update feature scans a computer(s) to find operating system updates available through Microsoft. This scan will identify any hotfixes or security patches that are needed in addition to listing other software updates that are available. Unfortunately, its greatest limitation for a Web administrator is that although it address the Windows operating system, it does not address the Microsoft IIS Web server application. That oversight is corrected by the Microsoft Network Security Hotfix Checker (HfNetChk). This is a command line tool written by Microsoft to assess not only the patch status for Windows NT 4.0 and Windows 2000 operating systems, but also the status of hotfixes for IIS 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later.

B.9.1 Using Windows Update

Windows Update is a utility provided by Microsoft in most versions of Windows (including some versions of 95 and NT and all versions of 98, ME, 2000, and XP) that allows users to scan their computer to find any updates that are available at that time from Microsoft and other participating vendors. Figures B.3 and B.4 demonstrate two different methods of accessing the Windows Update utility. It is suggested that users close all other applications before initiating the Windows Update feature.


NIST Computer Security Division 893 and CSRC Home Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Mail and News
Synchronize...
Windows Update
Show Related Links
Internet Options...
Address https://csrc.nist.gov/
Information Technology
Computer Security Division (CSD)
Computer Security Resource Center (CSRC)
NIST
About CSD:
Mission Statement
Projects / Focus Areas
CSD staff
Location
CSRC Website:
CSRC Homepage
Publications
News
Research Areas
Cryptographic Standards and Applications
CSRC Website Highlights

Opens the Windows Update Web page to update components.

Figure B.3: Accessing Windows Update Through Internet Explorer

To access the Windows Update from within Internet Explorer browser, click on *Tools* and then *Windows Update* in the pull-down menu.

Alternatively, a user can access the Windows Update from the Start Menu as demonstrated in Figure B.4. From the Windows desktop, click on the *Start* bar. From the menu, click on the *Windows Update* icon.

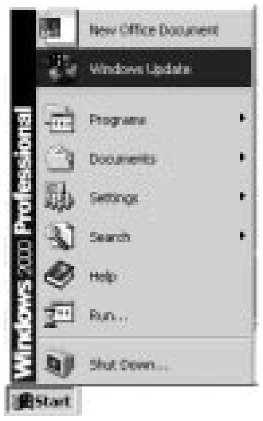
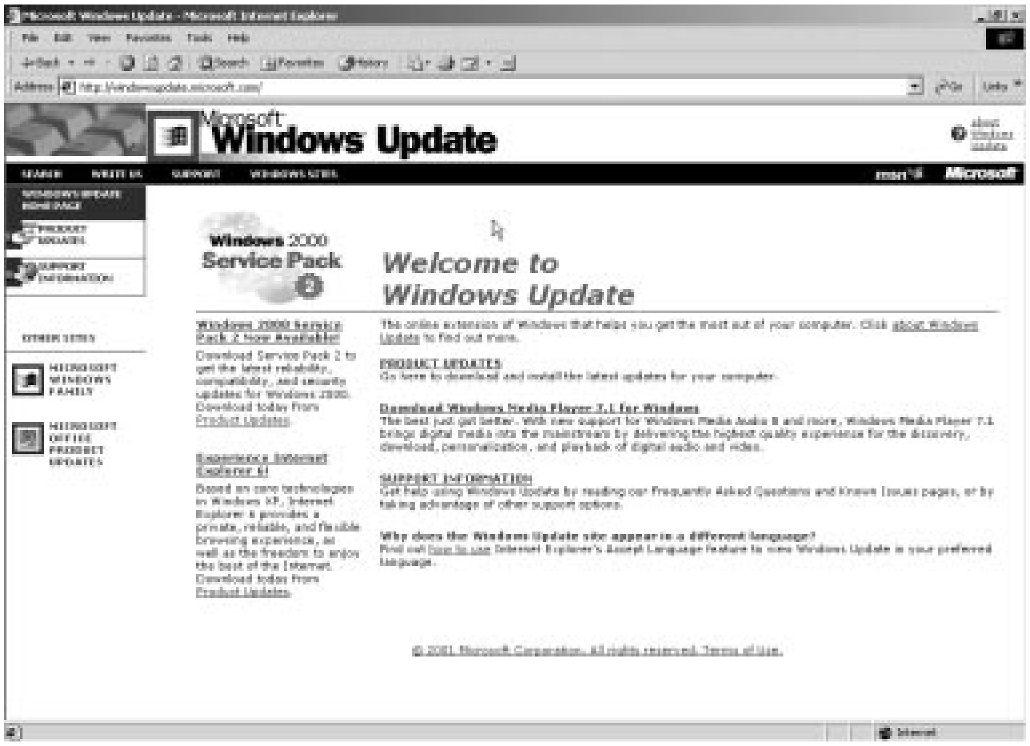


Figure B.4: Accessing Windows Update Though the Start Menu

Either option will launch Microsoft Internet Explorer (if it is not already active) and load the Microsoft Windows Update Web site (htttp://windowsupdate.microsoft.com). See Figure B.5 for the Windows Update homepage.


Microsoft Windows Update - Microsoft Internet Explorer

http://windowsupdate.microsoft.com/

Microsoft Windows Update

Windows 2000 Service Pack 2

Welcome to Windows Update

Windows 2000 Service Pack 2 Now Available!
Download Service Pack 2 to get the latest reliability, compatibility, and security updates for Windows 2000. Download today from Product Updates.

Experience Internet Explorer 6!
Based on core technologies in Windows XP, Internet Explorer 6 provides a private, reliable, and flexible browsing experience, as well as the freedom to enjoy the best of the Internet. Download today from Product Updates.

The online extension of Windows that helps you get the most out of your computer. Click about Windows Update to find out more.

PRODUCT UPDATES
Go here to download and install the latest updates for your computer.

Download Windows Media Player 7.1 for Windows
The best just got better. With new support for Windows Media Audio 8 and more, Windows Media Player 7.1 brings digital media into the mainstream by delivering the highest quality experience for the discovery, download, personalization, and playback of digital audio and video.

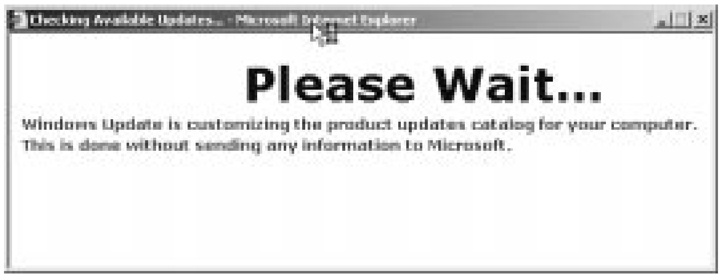
SUPPORT INFORMATION
Get help using Windows Update by reading our Frequently Asked Questions and Known Issues pages, or by taking advantage of other support options.

Why does the Windows Update site appear in a different language?
Find out how to use Internet Explorer's Accept Language feature to view Windows Update in your preferred language.

© 2001 Microsoft Corporation. All rights reserved. Terms of Use.

Figure B.5: Windows Update Homepage

To have the Windows Update scan a computer for updates, click on the "**PRODUCT UPDATES**" link. Note: This action is accomplished without sending any information to Microsoft or transmitting sensitive information on the host over the Internet. The Windows Update utility will commence its scan of the user's computer and derive a customized product update catalog specific to that computer (see Figure B.6). Having Windows Update automatically check the system has several advantages. This check assures that users will get the most up to date and accurate versions of the items chosen for download from the site. In addition, users will not waste time downloading components that are already installed.

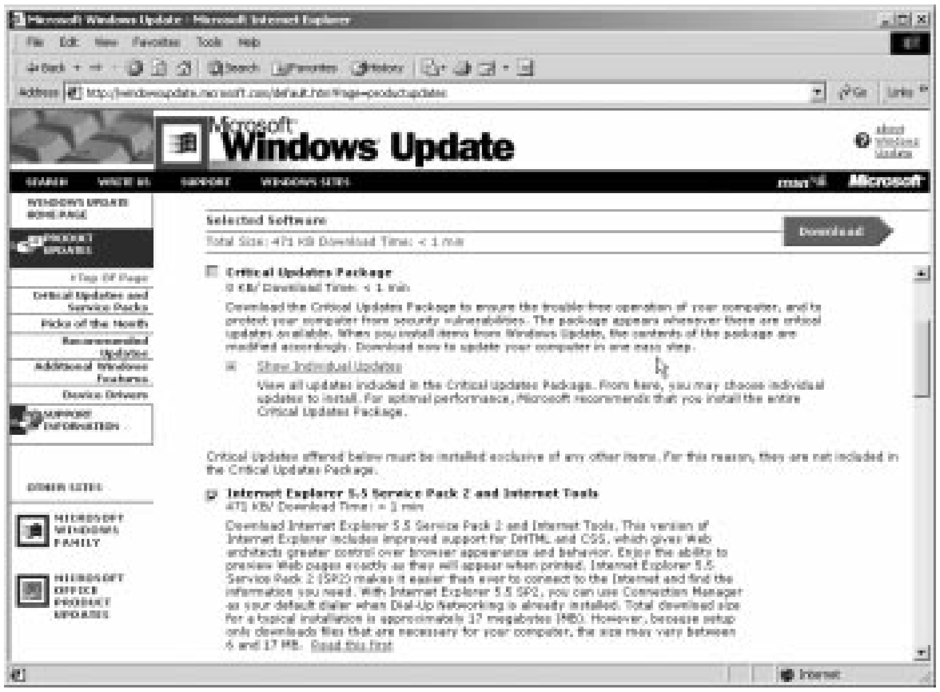

Checking Available Updates... - Microsoft Internet Explorer

Please Wait...

Windows Update is customizing the product updates catalog for your computer.
This is done without sending any information to Microsoft.

Figure B.6: Windows Update Scan

Once Windows Update has finished scanning the user's machine, it will generate a list of recommended updates (see Figure B.7). Users can browse the list, select components, and download the selected components.


Microsoft Windows Update - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://windowsupdate.microsoft.com/default.htm?page=productupdates
Go Links
Microsoft Windows Update
about Windows Update
SEARCH WRITE US SUPPORT WINDOWS SITES
msn Microsoft
WINDOWS UPDATE HOME PAGE
PRODUCT UPDATES
Top Of Page
Critical Updates and Service Packs
Picks of the Month
Recommended Updates
Additional Windows Features
Device Drivers
SUPPORT INFORMATION
OTHER SITES
MICROSOFT WINDOWS FAMILY
MICROSOFT OFFICE PRODUCT UPDATES
Download
Internet

Selected Software
Total Size: 471 KB Download Time: < 1 min

Critical Updates Package
0 KB/ Download Time: < 1 min
Download the Critical Updates Package to ensure the trouble-free operation of your computer, and to protect your computer from security vulnerabilities. The package appears whenever there are critical updates available. When you install items from Windows Update, the contents of the package are modified accordingly. Download now to update your computer in one easy step.

Show Individual Updates
View all updates included in the Critical Updates Package. From here, you may choose individual updates to install. For optimal performance, Microsoft recommends that you install the entire Critical Updates Package.

Critical Updates offered below must be installed exclusive of any other items. For this reason, they are not included in the Critical Updates Package.

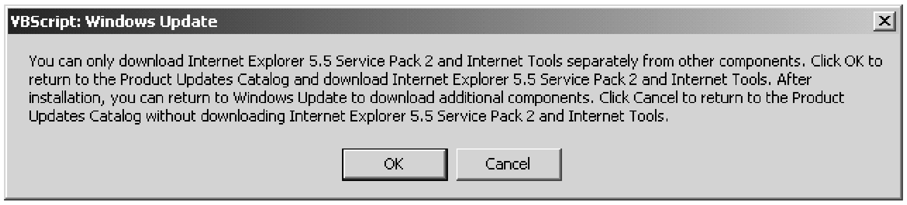
Internet Explorer 5.5 Service Pack 2 and Internet Tools
471 KB/ Download Time: < 1 min
Download Internet Explorer 5.5 Service Pack 2 and Internet Tools. This version of Internet Explorer includes improved support for DHTML and CSS, which gives Web architects greater control over browser appearance and behavior. Enjoy the ability to preview Web pages exactly as they will appear when printed. Internet Explorer 5.5 Service Pack 2 (SP2) makes it easier than ever to connect to the Internet and find the information you need. With Internet Explorer 5.5 SP2, you can use Connection Manager as your default dialer when Dial-Up Networking is already installed. Total download size for a typical installation is approximately 17 megabytes (MB). However, because setup only downloads files that are necessary for your computer, the size may vary between 6 and 17 MB. Read this first

Figure B.7: Windows Update Recommend Updates

The product updates are broken down into five different sections:

- **Critical Updates and Service Packs**—It is suggested that users download all Critical Update Packages as these fix known problems (often security issues) with their specific installation.
- **Picks of the Month**—These are new releases that add functionality to Windows but are not required to fix a known problem.
- **Recommended Updates**—These are older releases that add functionality to Windows but are not required to fix a known problem.
- **Additional Windows Features**—These are updates to other applications that are included with Windows (e.g., Internet Explorer, Media Player).
- **Device Drivers**—Listed here will be any updated device drivers for the computer. A device driver is a program that controls a piece of hardware (such as a printer, monitor, disk drive, or video card) that is attached to the computer. Note: Third parties that manufacture hardware and device drivers for this will not be listed unless the manufacturer has an agreement with Microsoft. A user should go to the appropriate manufacturer's Web site to get device driver updates.

Certain updates can only be downloaded individually. If this is the case, Windows Update will provide notification as shown in Figure B.8. If this happens, the user will have to repeat the process delineated above.


VBScript: Windows Update

You can only download Internet Explorer 5.5 Service Pack 2 and Internet Tools separately from other components. Click OK to return to the Product Updates Catalog and download Internet Explorer 5.5 Service Pack 2 and Internet Tools. After installation, you can return to Windows Update to download additional components. Click Cancel to return to the Product Updates Catalog without downloading Internet Explorer 5.5 Service Pack 2 and Internet Tools.

Figure B.8: Windows Update Warning

After selecting the patches to download, the Download Checklist page loads to confirm the selections (see Figure B-9). At this point, the user may choose to view the instructions, start the download and install, or go back and reselect the software.

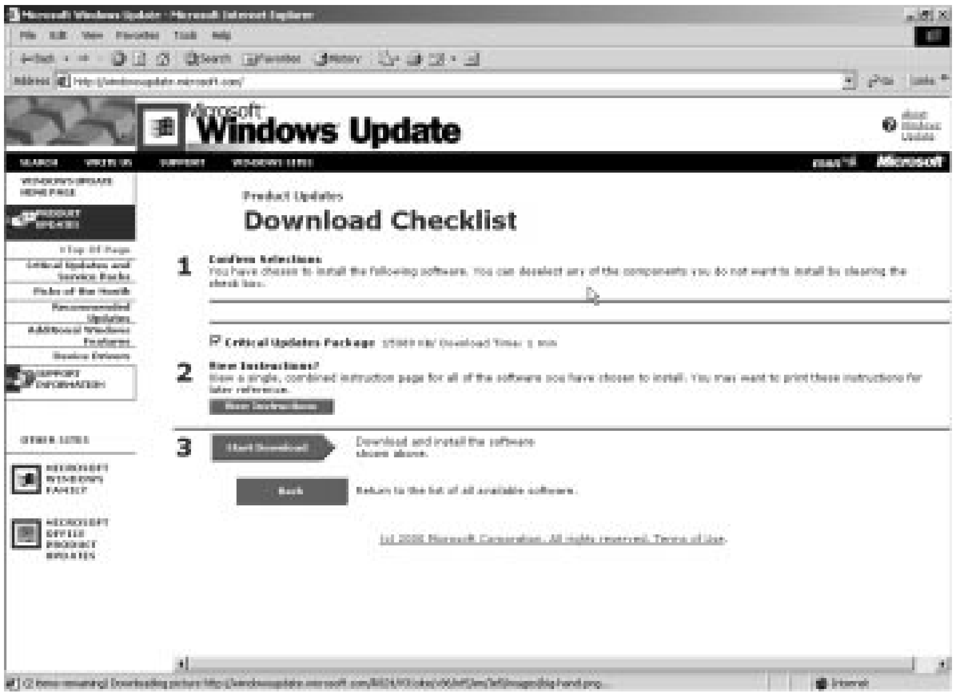
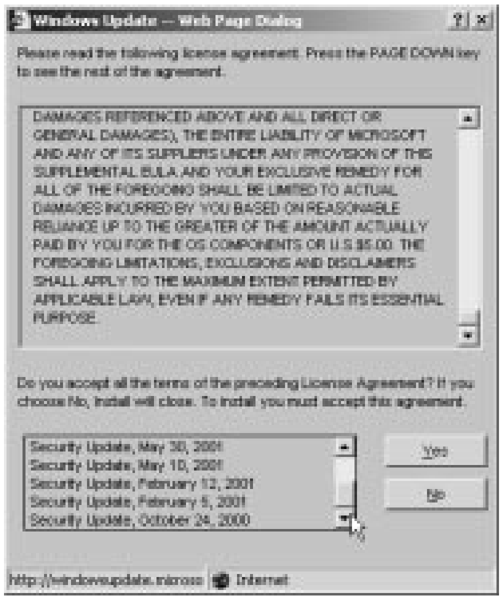


Figure B.9: Windows Update Download Checklist

After selecting "Start Download" from the Download Checklist page, an additional screen pops up to confirm the selection (see Figure B.10). At this point, the user may choose to view the instructions, license agreement, start the download and install (by clicking on the "Yes" button), or go back and reselect the software (by clicking on the "No" button).



Please read the following license agreement. Press the PAGE DOWN key to see the rest of the agreement.

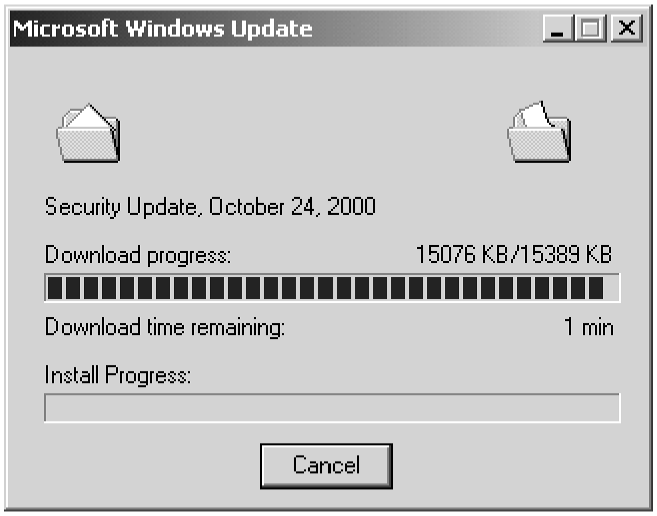
DAMAGES REFERENCED ABOVE AND ALL DIRECT OR GENERAL DAMAGES), THE ENTIRE LIABILITY OF MICROSOFT AND ANY OF ITS SUPPLIERS UNDER ANY PROVISION OF THIS SUPPLEMENTAL EULA AND YOUR EXCLUSIVE REMEDY FOR ALL OF THE FOREGOING SHALL BE LIMITED TO ACTUAL DAMAGES INCURRED BY YOU BASED ON REASONABLE RELIANCE UP TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE OS COMPONENTS OR U.S.$5.00. THE FOREGOING LIMITATIONS, EXCLUSIONS AND DISCLAIMERS SHALL APPLY TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, EVEN IF ANY REMEDY FAILS ITS ESSENTIAL PURPOSE.

Do you accept all the terms of the preceding License Agreement? If you choose No, Install will close. To install you must accept this agreement.

Security Update, May 30, 2001
Security Update, May 10, 2001
Security Update, February 12, 2001
Security Update, February 5, 2001
Security Update, October 24, 2000

Figure B.10: Windows Update Confirmation and License Agreement

Upon acceptance of the license agreement, the selected patches and software will be downloaded (see Figure B.11). The duration of the download will depend on several factors, including the file size and connection speed.



Security Update, October 24, 2000

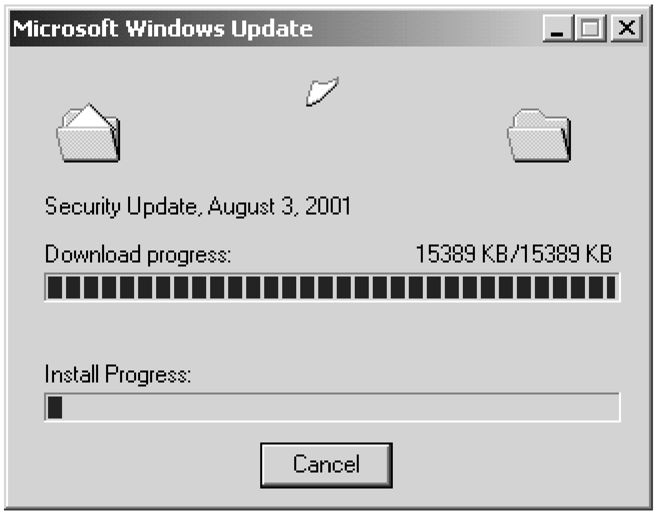
Download progress: 15076 KB/15389 KB

Download time remaining: 1 min

Install Progress:

Figure B.11: Windows Update Download Status Window

After the download is complete, Microsoft Windows Update will start the install process, which may take up to several minutes to complete (see Figure B.12).



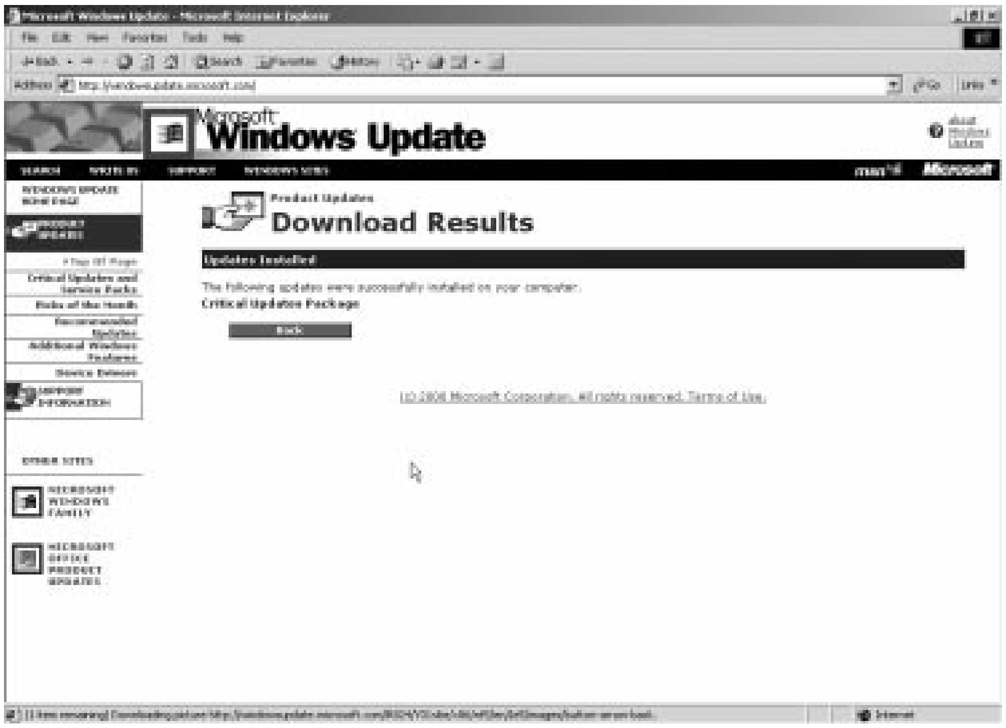
Security Update, August 3, 2001

Download progress: 15389 KB/15389 KB

Install Progress:

Figure B.12: Windows Update Install Status Window

Once the install is successfully completed, the browser window will confirm the success (see Figure B.13).



Product Updates

Download Results

Updates Installed

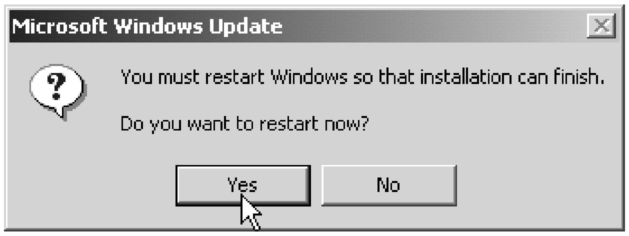
The following updates were successfully installed on your computer.

Critical Update Package

(c) 2000 Microsoft Corporation. All rights reserved. Terms of Use.

Figure B.13: Windows Update Download Results Window.

Often, a reboot may be necessary to activate the updates (see Figure B.14). Click the "Yes" button to restart the computer. Click the "No" button to continue the current Windows session (changes will NOT take effect until the computer has successful rebooted). If Windows Update does not prompt for a reboot, then the changes do not require it and are effective from the time of a successful install.



You must restart Windows so that installation can finish.

Do you want to restart now?

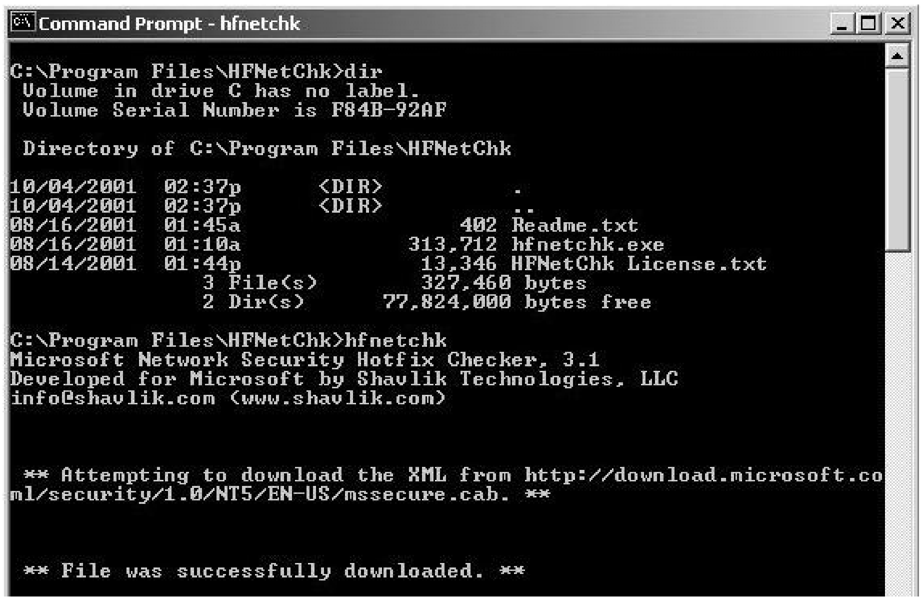
Figure B.14: Windows Update Reboot Dialog Box

If additional patches were required but could not be download simultaneously, repeat the Windows Update process as required.

B.10 Using Microsoft Network Security Hotfix Checker

HfNetChk is a command line tool written by Microsoft to access the patch status for Windows NT 4.0 and Windows 2000 operating systems and hotfixes for IIS 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later. Although less easy to use, it supports more Microsoft products than either Microsoft Windows Update or MPSA.

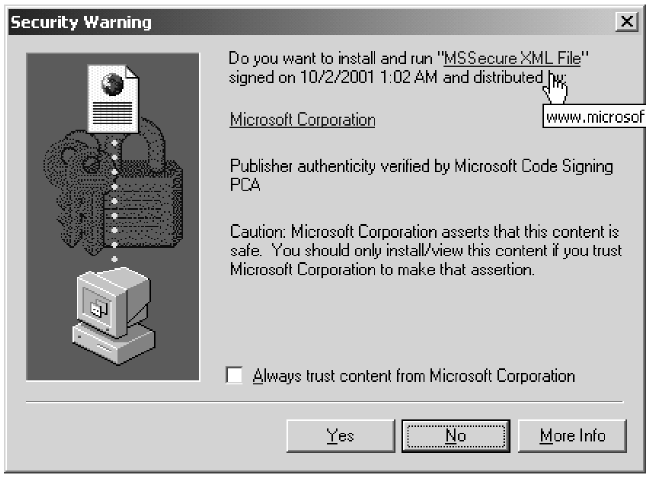
After downloading and installing HfNetChk, run *hfnetchk.exe* from the command line. The program will then attempt to download an Extend Markup Language (XML) file from Microsoft. This XML file contains the information on the current patch and update status of the programs and operating systems being checked (Figure B,15).



```
C:\Program Files\HFNetChk>dir
 Volume in drive C has no label.
 Volume Serial Number is F84B-92AF

 Directory of C:\Program Files\HFNetChk

10/04/2001  02:37p      <DIR>          .
10/04/2001  02:37p      <DIR>          ..
08/16/2001  01:45a                 402 Readme.txt
08/16/2001  01:10a             313,712 hfnetchk.exe
08/14/2001  01:44p              13,346 HFNetChk License.txt
               3 File(s)        327,460 bytes
               2 Dir(s)      77,824,000 bytes free

C:\Program Files\HFNetChk>hfnetchk
Microsoft Network Security Hotfix Checker, 3.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)


 ** Attempting to download the XML from http://download.microsoft.co
m/security/1.0/NT5/EN-US/mssecure.cab. **


 ** File was successfully downloaded. **
```

Figure B.15: Running Hfnetchk

To start the actual scan, agree to the installation and running of the downloaded XML file (see Figure B.16).


Security Warning

Do you want to install and run "MSSecure XML File" signed on 10/2/2001 1:02 AM and distributed by:

Microsoft Corporation

Publisher authenticity verified by Microsoft Code Signing PCA

Caution: Microsoft Corporation asserts that this content is safe. You should only install/view this content if you trust Microsoft Corporation to make that assertion.

☐ Always trust content from Microsoft Corporation

Figure B.16: Installing XML File

After clicking the "Yes" button, the program will load the XML files and scan the computer. The results listed on the command line screen provide limited information (see Figure B.17)

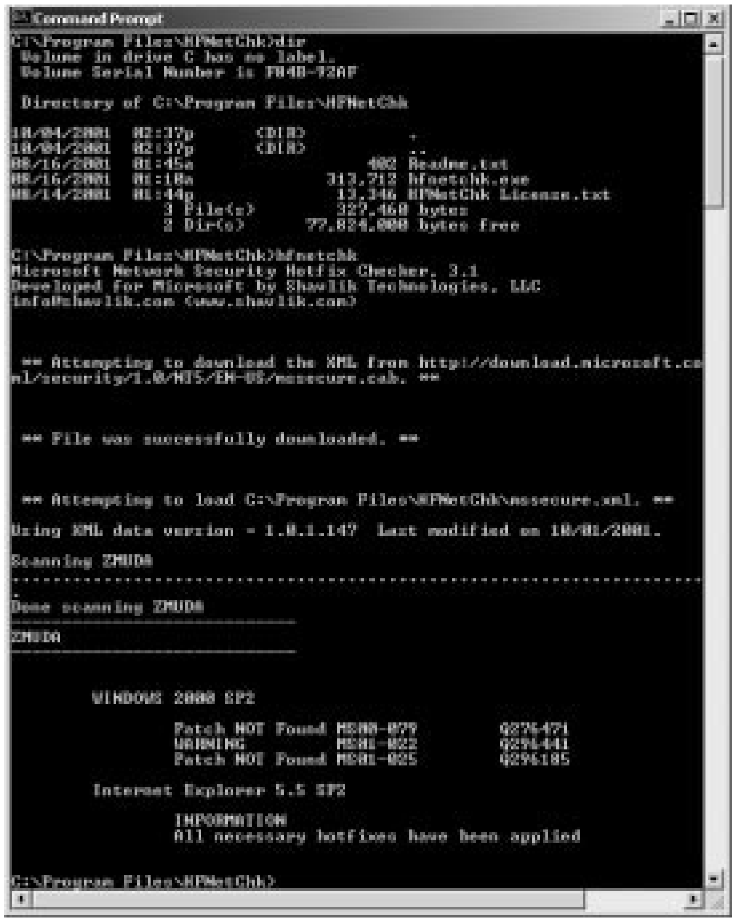
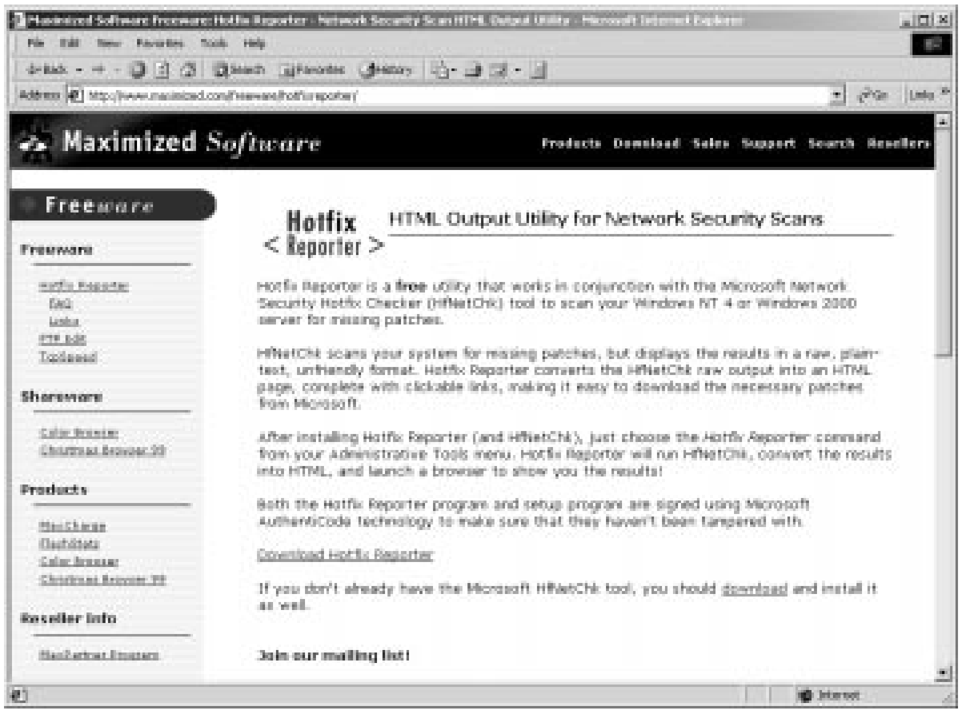

```
Command Prompt
C:\Program Files\HFNetChk>dir
 Volume in drive C has no label.
 Volume Serial Number is F048-92AF

 Directory of C:\Program Files\HFNetChk

10/04/2001  02:37p      <DIR>          .
10/04/2001  02:37p      <DIR>          ..
08/16/2001  01:45a                 402 Readme.txt
08/16/2001  01:10a             313,712 hfnetchk.exe
08/14/2001  01:44p              13,346 HFNetChk License.txt
               3 File(s)        327,460 bytes
               2 Dir(s)      77,824,000 bytes free

C:\Program Files\HFNetChk>hfnetchk
Microsoft Network Security Hotfix Checker, 3.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)


 ** Attempting to download the XML from http://download.microsoft.co
ml/security/1.0/NTS/EN-US/mssecure.cab. **


 ** File was successfully downloaded. **


 ** Attempting to load C:\Program Files\HFNetChk\mssecure.xml. **

Using XML data version = 1.0.1.147  Last modified on 10/01/2001.

Scanning ZMUDA
----------------------------------------------------------------------
.
Done scanning ZMUDA
----------------------------
ZMUDA
----------------------------


        WINDOWS 2000 SP2

                Patch NOT Found MS00-079        Q276471
                WARNING         MS01-022        Q296441
                Patch NOT Found MS01-025        Q296185

        Internet Explorer 5.5 SP2

                INFORMATION
                All necessary hotfixes have been applied

C:\Program Files\HFNetChk>
```

Figure B.17: HfNetChk Output

Although the listed information can be useful, the format and lack of detail is not comprehensive enough for most users. To correct this deficiency, there is a freeware tool

Maximized Software Hotfix Reporter. This utility works in conjunction with HfNetChk to display the results in an easy-to-read HTML format with hyperlinks making it easy to download the associated patches and hotfixes. This utility can be found at http://www.maximized.com/freeware/hotfixreporter/ (see Figure B.18).



HTML Output Utility for Network Security Scans

Hotfix Reporter is a **free** utility that works in conjunction with the Microsoft Network Security Hotfix Checker (HfNetChk) tool to scan your Windows NT 4 or Windows 2000 server for missing patches.

HfNetChk scans your system for missing patches, but displays the results in a raw, plain-text, unfriendly format. Hotfix Reporter converts the HfNetChk raw output into an HTML page, complete with clickable links, making it easy to download the necessary patches from Microsoft.

After installing Hotfix Reporter (and HfNetChk), just choose the Hotfix Reporter command from your Administrative Tools menu. Hotfix Reporter will run HfNetChk, convert the results into HTML, and launch a browser to show you the results!

Both the Hotfix Reporter program and setup program are signed using Microsoft AuthentiCode technology to make sure that they haven't been tampered with.

Download Hotfix Reporter

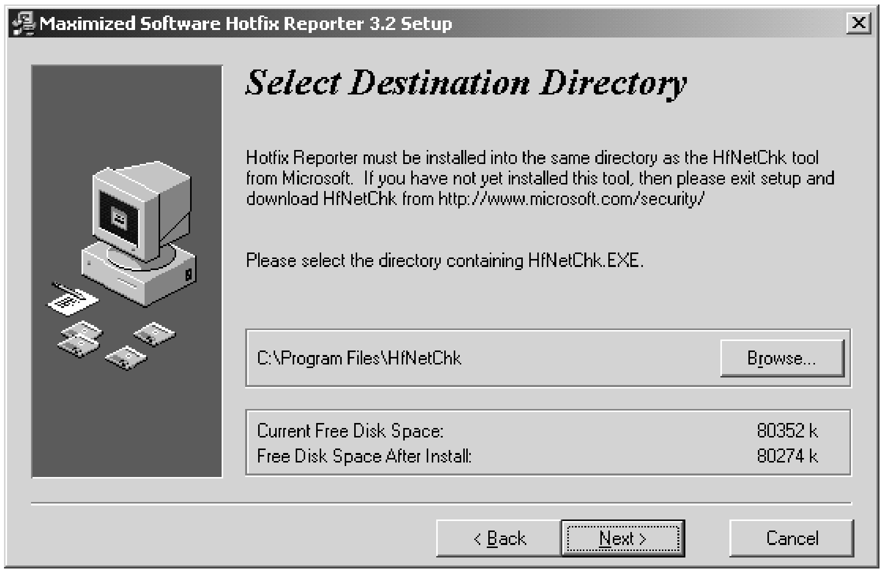
If you don't already have the Microsoft HfNetChk tool, you should download and install it as well.

Join our mailing list!

Figure B.18: Maximized Software Web Site

Download by clicking on the hyperlink Download Hotfix Reporter.

Run the setup program to install Hotfix Reporter. The Hotfix Reporter must be installed in the same directory as HfNetChk (see Figure B.19).



Select Destination Directory

Hotfix Reporter must be installed into the same directory as the HfNetChk tool from Microsoft. If you have not yet installed this tool, then please exit setup and download HfNetChk from http://www.microsoft.com/security/

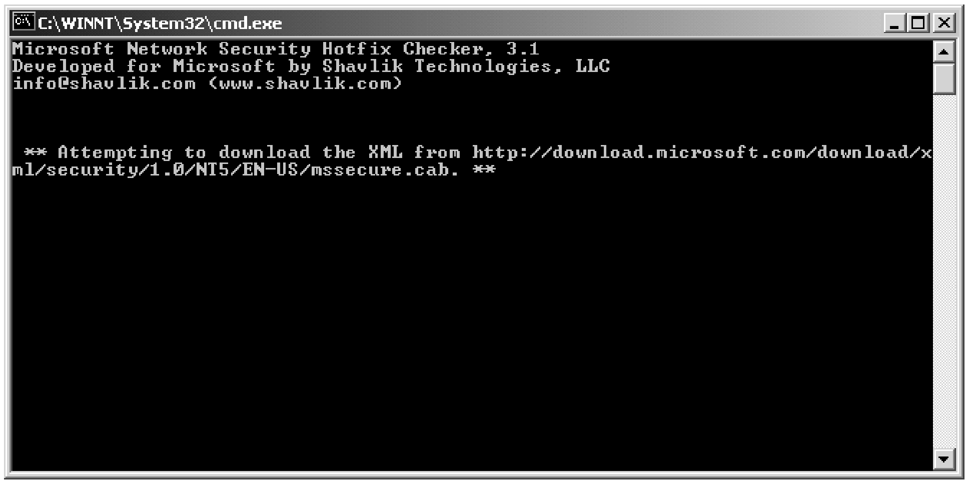
Please select the directory containing HfNetChk.EXE.

C:\Program Files\HfNetChk

Current Free Disk Space: 80352 k
Free Disk Space After Install: 80274 k

Figure B.19: Installing Hotfix Reporter

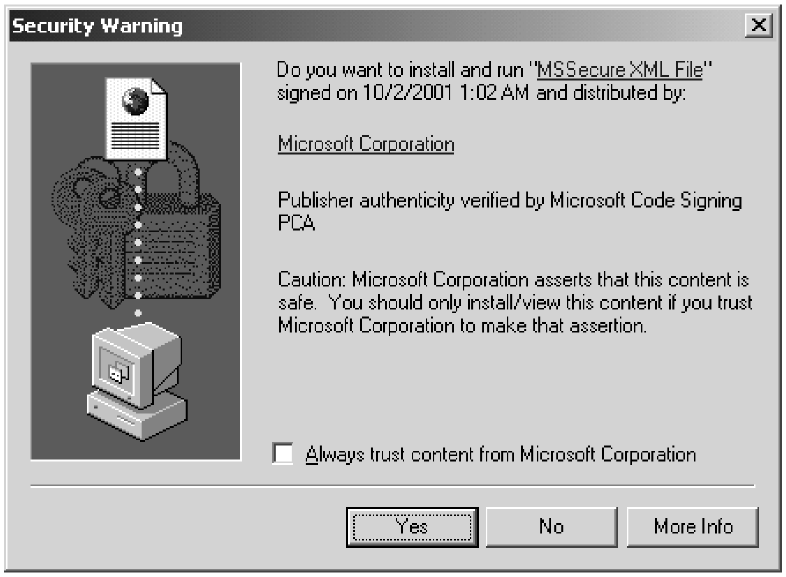
After installation, launch the Hotfix Reporter from the Windows Start menu. The execution of the Hotfix reporter is very similar to HfNetChk (see Figure B.20).



```
Microsoft Network Security Hotfix Checker, 3.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)


 ** Attempting to download the XML from http://download.microsoft.com/download/x
ml/security/1.0/NT5/EN-US/mssecure.cab. **
```

Figure B.20: Running HfNetChk With Hotfix Reporter

When HfNetChk attempts to install and run the latest XML file from Microsoft, the user will need to click "Yes" to continue the operation (see Figure B.21).



Do you want to install and run "MSSecure XML File" signed on 10/2/2001 1:02 AM and distributed by:

Microsoft Corporation

Publisher authenticity verified by Microsoft Code Signing PCA

Caution: Microsoft Corporation asserts that this content is safe. You should only install/view this content if you trust Microsoft Corporation to make that assertion.

☐ Always trust content from Microsoft Corporation

Figure B.21: Installing Microsoft XML Data File

Once HfNetChk has completed the scan, the Hotfix Reporter will then open a HTML file in the default Web browser. This action provides results in a more readable and usable format than that of HfNetChk. These results are also stored locally and can be reviewed as needed. Note that it is important to rerun the Hotfix Reporter again after updates are installed to ensure that the system is running appropriately.

B.11 Microsoft Windows Security Checklist:

Table B.7 is a checklist that lists steps to secure Microsoft Windows. This checklist should not be considered the only method of securing Windows, but it will work well in tandem with other operations. Administrators should follow the steps on this checklist before using the checklist for IIS.

Table B.7: Microsoft Windows Security Checklist

Completed	Action
☐	Protect server environment
☐	Remove unnecessary hardware from server
☐	Update operating system
☐	Apply security template (where applicable)
☐	Restrict anonymous registry access
☐	Remove/disable unnecessary services
☐	Format/convert all partitions to NTFS
☐	Remove Everyone Group from ACLs of protected resources; replace with Authenticated Users group
☐	Synchronize IIS and NTFS permissions
☐	Secure Default Windows Repair directory
☐	Run Windows Update Weekly
☐	Run HfNetChk Weekly

B.12 Microsoft Internet Information Server Security Checklist

Table B.8 is a checklist that presents steps to implement the security for IIS. This assumes that the checklist for Windows security has already been followed. Each step provided here is explained in detail in the sections above.

Table B.8: Microsoft IIS Security Checklist

Completed	Action
☐	Implement SSL for all Web sites (where applicable)
☐	Enable logging (W3C extended format)
☐	Set proper log file ACLs
☐	Backup logs offline weekly if not daily
☐	Remove all IIS sample applications
☐	Remove unused script mappings
☐	Disable RDS Support
☐	Disable IPP Support (IIS 5.x only)
☐	Disable parent paths

Completed	Action
☐	Enable strong authentication for protected resources
☐	Set NTFS directory permissions for Web folders
☐	Back up IIS Metabase frequently
☐	Run IIS Lockdown Tool

Appendix C. Online Web Server Security Resources

Active Content Security Resources

Resource/Title	URL
ASP Alliance	http://www.aspalliance.com/
Exploiting Common Vulnerabilities in PHP Applications	http://www.securereality.com.au/studyinscarlet.txt
Java Security	http://java.sun.com/security/
Java Security Frequently Asked Questions	http://www.cs.princeton.edu/sip/faq/java-faq.php3
PHP Manual, Chapter 4 Security	http://www.securereality.com.au/studyinscarlet.txt
www.asp.net	http://www.asp.net
www.cgisecurity.com	http://www.cgisecurity.com/lib/

Apache Web Server Security Resources

Resource/Title	URL
Apache-server.com	http://apache-server.com/
Apache Tutorials	http://httpd.apache.org/docs/misc/tutorials.html
Apache SSL	http://www.apache-ssl.org/
Introduction to Securing Apache	http://www.linux.com/newsitem.phtml?sid=12&aid=3549
Securing Apache	http://www.macromedia.com/v1/documentcenter/partners/asz_aswps_securing_apache.pdf
Securing Your Web Pages with Apache	http://apache-server.com/tutorials/LPauth1.html

Computer Crime/Incident Handling

Resource/Title	URL
CERT/CC How the FBI Investigates Computer Crime	http://www.cert.org/tech_tips/FBI_investigates_crime.html
CERT/CC Responding to Intrusions	http://www.cert.org/security-improvement/modules/m06.html
CERT/CC Detecting Signs of Intrusion	http://www.cert.org/security-improvement/modules/m09.html
Computer Evidence Processing Steps	http://www.forensics-intl.com/evidguid.html
Federal Guidelines on Searching and Seizing Computers	http://www.usdoj.gov/criminal/cybercrime/searching.html
Federal Code Related to Cybercrime	http://www.usdoj.gov/criminal/cybercrime/fedcode.htm
NIST ITL Bulletin September 1999: Securing Web Servers	http://csrc.nist.gov/publications/nistbul/09-99.pdf
NIST SP 800-3 Establishing a Computer Security Incident Response Capability	http://csrc.nist.gov/publications/nistpubs/800-3/800-3.pdf

Digital Certificate Providers (Third-Party Certificate Authorities)

Resource/Title	URL
Thawte Consulting	http://www.thawte.com/certs/server/request.html
CertiSign Certificadora Digital Ltda	http://www.certisign.com.br/
IKS GmbH	http://www.iks-jena.de/produkte/ca/
BelSign NV/SA	http://www.belsign.be/
TC TrustCenter	http://www.trustcenter.de/html/Produkte/TC_Server/855.htm
NLsign B.V.	http://www.nlsign.nl/
Deutsches Forschungsnetz	http://www.pca.dfn.de/dfnpca/certify/ssl/
128i Ltd.	http://www.128i.com/
Entrust.net Ltd.	http://www.entrust.net/products/index.htm
GlobalSign NV/SA	http://www.globalsign.net/
Certplus SA	http://www.certplus.com/
GeoTrust Inc.	http://www.freessl.com/
Register.com	http://commercelock.register.com/
Lanechange.net	http://www.lanechange.net/

Federal Government Web Server Security Resources

Resource/Title	URL
Defense Information System Agency (DISA) Security Checklist	http://iase.disa.mil/techguid/checklist.html
DoD Web Site Administration Policies and Procedures	http://www.defenselink.mil/webmasters/policy/dod_web_policy_12071998.html
Federal Computer Incident Response Center (FedCIRC)	http://www.fedcirc.gov/
Guidelines for Establishing and Maintaining Publicly Accessible DoD Web Information Service	http://www.defenselink.mil/webmasters/policy/policy97.html
National Infrastructure Protection Center	http://www.nipc.gov/
National Information Assurance Partnership	http://www.niap.nist.gov/
National Security Agency Rainbow Series	http://www.radium.ncsc.mil/tpep/library/rainbow/index.html
National Security Agency Security Recommendation Guides	http://nsa1.www.conxion.com/
NIST Computer Security Resource Center	http://csrc.nist.gov/
NIST ICAT Vulnerability Metabase	http://icat.nist.gov/
Office of Management and Budget Circular No. A-130	http://www.whitehouse.gov/omb/circulars/a130/
U.S. Department of Energy Computer Incident Advisory Capability (CIAC)	http://www.ciac.org/ciac/

General Web Server Security Resources

Resource/Title	URL
CERIAS	http://www.cerias.purdue.edu/
Computer Emergency Response Team (CERT)	http://www.cert.org/
CERT/CC Securing Public Web Servers	http://www.cert.org/security-improvement/modules/m11.html
A Look Into Web Server and Web Application Attack Signatures	http://www.cgisecurity.com/papers/fingerprint-port80.txt
NIST ICAT Vulnerability Metabase	http://icat.nist.gov/
RISKS Forum	http://catless.ncl.ac.uk/Risks/
Security Administration, Networking, and Security (SANS) Institute	http://www.sans.org/
SANS Twenty Most Critical Internet Security Vulnerabilities	http://www.sans.org/top20.htm
Shockwave Security Issues	http://www.webcomics.com/shockwave/
Trust Management on the World Wide Web	http://www.firstmonday.dk/issues/issue3_6/khare/
World Wide Web Security Frequently Asked Questions	http://www.w3.org/Security/Faq/www-security-faq.html

IIS Web Server Security Resources

Resource/Title	URL
eEye Advisories and Alerts	http://www.eeye.com/html/Research/Advisories/index.html
IIS Lockdown Tool	http://www.microsoft.com/Downloads/Release.asp?ReleaseID=33961
IIS 4.0 Security Checklist	http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/iischk.asp
IIS 5.0 Baseline Security Checklist	http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/iis5cl.asp
IIS 5.0 Security	http://www.microsoft.com/windows2000/en/server/iis/default.asp?url=/WINDOWS2000/en/server/iis/htm/core/iiabtsc.htm
NSA Guide to the Secure Configuration and Administration of Microsoft IIS 5.0	http://nsa1.www.conxion.com/win2k/guides/w2k-14.pdf
Secure IIS 5.0 Checklist	http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/tools/iis5chk.asp
Securing Your Site Against Intruders	http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/iis/deploy/projplan/iischp5.asp

Miscellaneous Web Security Resources

Resource/Title	URL
Microsoft Internet Explorer Security Page	http://www.microsoft.com/windows/ie/security/default.asp
dominosecurity.org	http://www.dominosecurity.org/
Lotus Domino Security Page	http://www.lotus.com/home.nsf/welcome/securityzone
Netcraft	http://www.netcraft.com/
Netscape Security Page	http://home.netscape.com/security/

Web Bot Information

Resource/Title	URL
BotSpot	http://bots.internet.com/
Configuring the robots.txt Files	http://www.internet-tips.net/Advertising/robots.htm
NIST Mobile Agent Security	http://csrc.nist.gov/mobileagents/web-overview/index.htm
Showing Robots the Door	http://www.ariadne.ac.uk/issue15/robots/
University of Maryland Baltimore County (UMBC) AgentWeb	http://agents.umbc.edu/

Appendix D. Glossary

Host—A computer that acts as a source of information or signals. The term can refer to almost any kind of computer, from a centralized mainframe that is a host to its terminals, to a server that is host to its clients, to a desktop personal computer (PC) that is host to its peripherals. In network architectures, a client station (user's machine) is also considered a host because it is a source of information to the network in contrast to a device such as a router or switch that directs traffic.

Hoftfix—Microsoft's term for a bug fix, which is accomplished by replacing one or more existing files in the operating system or application with revised versions.

Network Administrator—A person who manages a local area network (LAN) within an organization. Responsibilities include network security, installing new applications, distributing software upgrades, monitoring daily activity, enforcing licensing agreements, developing a storage management program, and providing for routine backups.

Operating System—The master control program that runs the computer. The first program loaded when the computer is turned on, its main part, the "kernel," resides in memory at all times. The operating system sets the standards for all application programs that run in the computer. The applications "talk to" the operating system for all user interface and file management operations.

Patch—A patch (sometimes called a "fix") is a "repair job" for a piece of programming. A patch is the immediate solution that is provided to users; it can sometimes be downloaded from the software maker's Web site. The patch is not necessarily the best solution for the problem, and the product developers often find a better solution to provide when they package the product for its next release. A patch is usually developed and distributed as a replacement for or an insertion in compiled code (that is, in a binary file or object module). In larger operating systems, a special program is provided to manage and track the installation of patches.

Service Pack—A software patch that is applied to an installed application. It is either downloaded from the vendor's Web site or distributed via Compact Disc Read Only Memory (CD-ROM). When executed, it modifies the application in place.

System—See host.

System Administrator—A person who manages a multiuser computer system and whose responsibilities are similar to that of a network administrator. A system administrator would perform systems programmer activities with regard to the operating system and other network control programs.

Vulnerability—A security exposure in an operating system or other system software or application software component. A variety of organizations maintain publicly accessible databases of vulnerabilities based on the version number of the software. Each vulnerability can potentially compromise the system or network if exploited.

Web Administrator—The Web equivalent of a system administrator. Web administrators are system architects responsible for the overall design and implementation of an Internet Web site or intranet. They may or may not be responsible for Web content, which is traditionally the purview of the Webmaster (see below).

Webmaster—A person responsible for the implementation of a Web site. Webmasters must be proficient in hypertext markup language (HTML) and one or more scripting and interface languages, such as JavaScript and Perl. They may or may not be responsible for the underlying server, which is traditionally the responsibility of the Web administrator (see above).

Web Server—A computer that provides World Wide Web (WWW) services on the Internet. It includes the hardware, operating system, Web server software, Transport Control Protocol (TCP)/Internet Protocol (IP), and the Web site content (Web pages). If the Web server is used internally and not by the public, it may be known as an “intranet server.”

Appendix E. Web Security Tools and Applications

File Integrity Checkers

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
AIDE	Unix and Linux	http://www.cs.tut.fi/~rammer/aide.html	✓		Free
Description	*Advanced Intrusion Detection Environment (AIDE) is a free replacement for Tripwire. It performs file integrity checking and supports a number a large number of Unix and Linux platforms.*				
LANGuard	Windows 2000/NT	http://www.gfi.com/languard/		✓	Free
Description	*LANguard File Integrity Checker is a utility that provides intrusion detection by checking whether files have been changed, added, or deleted on a Windows 2000/NT system.*				
Tripwire	Windows, Unix, Linux, and Routers	http://www.tripwiresecurity.com/	✓	✓	Free to $$$
Description	*Tripwire monitors file changes, verifies integrity, and notifies the administrator of any violations of data on network hosts.*				

$$$=This product involves a fee.

Log File Analysis Tools

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
Analog	Most common operating systems	http://www.analog.cx/intro.html	✓	✓	Free
Description	*Analog is an automated Web server log file analysis tool that will compile on nearly an platform that supports the C programming language.*				
Fwgstat	Most Linux/Unix based operating systems	http://www.ibiblio.org/jem/fwgstat.html	✓		Free
Description	*Fwgstat is an automated Web server, FTP and, gopher log file analysis tool.*				
LiveStats6	Most Web servers and operating systems	http://www.deepmetrix.com/livestats/	✓	✓	$$$
Description	*Livestat6 is an automated Web server log file analysis tool.*				
NetTracker	Most Web servers and operating systems	http://www.sane.com/products/NetTracker/	✓	✓	$$$
Description	*NetTracker is automated Web server log file analysis tool.*				
Wwwstat and splitlog	Linux and Unix with Perl installed	http://www-old.ics.uci.edu/pub/websoft/wwwstat/	✓		Free
Description	*Wwwstat and splitlog is an automated Web server log file analysis tool for common log file format access_log files.*				

$$$=This product involves a fee.

Network Sniffers

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
Dsniff	Unix sniffer	http://www.monkey.org/~dugsong/dsniff/	✓	✓	Free
Description	*Dsniff is a collection of tools for network auditing and penetration testing. Dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, and webspy passively monitor a network for interesting data (passwords, e-mail, files, etc.). Arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching). Sshmitm and webmitm implement active monkey-in-the-middle attacks against redirected SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKIs.*				
Ethereal	Unix/Windows sniffer with GUI	http://www.ethereal.com/	✓	✓	Free
Description	*Ethereal is a free network protocol analyzer for Unix and Windows. It allows users to examine data from a live network or from a capture file on disk. It can interactively browse the capture data, viewing summary and detail information for each packet. Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session.*				
Sniffit	Unix sniffer	http://reptile.rug.ac.be/~coder/sniffit/sniffit.html http://www.symbolic.it/Prodotti/sniffit.html (Windows)	✓	✓	Free
Description	*Sniffit is a freeware general-purpose sniffer for various versions of Linux, Unix, and Windows.*				
Snort	Unix sniffer and IDS	http://www.snort.org	✓	✓	Free
Description	*Snort is a freeware lightweight IDS and general-purpose sniffer for various versions of Linux, Unix and Windows.*				
TCPDump	Unix sniffer	http://www-nrg.ee.lbl.gov/	✓		Free
Description	*TCPdump is a freeware general-purpose sniffer for various versions of Linux and Unix.*				

$$$=This product involves a fee.

Scanning and Enumeration Tools

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
DUMPSec	Windows enumeration tool	http://www.systemtools.com		✓	Free
Description	*DumpSec is a security auditing program for Microsoft Windows. It dumps the permissions (DACLs) and audit settings (SACLs) for the file system, registry, printers, and shares in a concise, readable listbox format so that holes in system security are readily apparent. DumpSec also dumps user, group, and replication information.*				
Firewalk	Firewall filter rule mapper	http://www.packetfactory.net/firewalk/	✓		Free
Description	*Firewalk is an application that employs traceroute-like techniques to analyze IP packet responses to determine gateway ACL filters and map networks. This allows Firewalk to determine the filter rules in place on packet-forwarding devices.*				
Nmap	Port scanner OS detection	http://www.insecure.org/nmap/	✓	✓	Free
Description	*Nmap ("Network Mapper") is an open source utility for network exploration or security auditing. It was designed to rapidly scan large networks, although it also works against single hosts. Nmap uses raw IP packets to determine what hosts are available on the network, what services (ports) they are offering, what operating system (and version) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.*				
Solarwinds	Network enumeration	http://www.solarwinds.net/		✓	$$$
Description	*Solarwinds is a collection of network and management and discovery tools.*				
SuperScan	Port scanner, OS detection, and banner enumeration	http://www.foundstone.com/		✓	Free
Description	*SuperScan is a GUI network mapper. It will rapidly scan large networks to determine what hosts are available on the network, what services they are offering, the version of these services, and the type and version of the operating system. It will also perform reverse DNS lookup.*				

$$$=This product involves a fee.

Vulnerability Scanning Tools

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
AppScan	Vulnerability scanner	http://www.pgp.com/products/	✓	✓ (client only)	$$$
Description	*AppScan is a Web application vulnerability scanner.*				
CyberCop Scanner	Vulnerability scanner	http://www.pgp.com/products/		✓	$$$
Description	*CyberCop Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
Domilock	Vulnerability scanner	http://domilockbeta.2y.net/		✓	Free
Description	*Domilock is a Web based Lotus Domino Web Server vulnerability scanner.*				
ISS Internet Scanner	Vulnerability scanner	http://www.iss.net/		✓	$$$
Description	*ISS Internet Scanner is a network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
Nessus	Vulnerability scanner	http://www.nessus.org/	✓	✓	Free
Description	*Nessus is a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
Retina	Vulnerability scanner	http://www.eeye.com		✓	$$$
Description	*Retina is a general-purpose network security scanner that identifies a large number of Web server vulnerabilities.*				
SARA	Vulnerability scanner	http://www-arc.com/sara/	✓		Free
Description	*SARA is a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
WebInspect	Web Vulnerability scanner	http://www.spidynamics.com/	✓	✓	$$$
Description	*WebInspect is Web application vulnerability scanner.*				
Whisker	CGI Vulnerability scanner	http://www.wiretrip.net/rfp/2	✓	✓	Free
Description	*Whisker is scanner that identifies vulnerabilities in CGI scripts.*				

$$$=This product involves a fee.

Web Server Hardening Tools

Tool	Capability	Web site	Linux/ Unix	Win32	Cost
Bastille Hardening System	Hardens Linux	http://www.bastille-linux.org/	✓		Free
Description	*Bastille Hardening System attempts to "harden" or "tighten" the Linux operating system. It supports Red Hat and Mandrake system and it attempts to provide the most secure, yet usable, system possible.*				
IIS Lockdown Tool	Hardens IIS	http://www.microsoft.com/downloads/		✓	Free
Description	*IIS lockdown tool assists Web administrators in locking down IIS versions 4.0 and 5.0.*				
Microsoft Network Security Hotfix Checker	Windows 2000/NT	http://www.microsoft.com/downloads/		✓	Free
Description	*This checker allows Web administrators to assess the patch status for Windows NT 4.0 and Windows 2000 operating systems, as well as the status of hotfixes for IIS 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later.*				
Windows Update	Assists in the update of most versions of Windows	http://www.microsoft.com/downloads/		✓	Free
Description	*Windows Update allows Web administrators to scan their server to find any updates that are available at that time from Microsoft and other participating vendors.*				

Appendix F. References

[Cho02]	Pete Chow, *AES Ciphersuites for TLS*, January 2002, http://www.ietf.org/internet-drafts/draft-ietf-tls-ciphersuite-06.txt
[MASS99]	Commonwealth of Massachusetts, Executive Order 412, 1999, http://www.state.ma.us/consumer/New/privexeco.htm
[NIST01a]	Wayne A. Jansen, NIST Special Publication 800-28, *Guidelines on Active Content and Mobile Code*, October 2001, http://csrc.nist.gov/publications/nistpubs/index.html
[NIST01b]	Rebecca Bace and Peter Mell, NIST Special Publication 800-31, *Intrusion Detection Systems*, August 2001, http://csrc.nist.gov/publications/nistpubs/index.html
[NIST02]	John Wack, et al, NIST Special Publication 800-41, *Guidelines on Firewalls and Firewall Policy*, January 2002, http://csrc.nist.gov/publications/nistpubs/index.html
[OMB00a]	Office of Management and Budget Memorandum 2000-13, 2000, http://www.whitehouse.gov/omb/memoranda/m00-13.html
[OMB00b]	Office of Management and Budget Cookie Clarification Letter 1, 2000, http://www.whitehouse.gov/omb/inforeg/cookies_letter72800.html
[OMB00c]	Office of Management and Budget Cookie Clarification Letter 2, 2000, http://www.whitehouse.gov/omb/inforeg/cookies_letter90500.html
[RSA00]	*PKCS #10 Version 1.7, Certification Request Syntax Standard*, May 26, 2000, http://www.rsasecurity.com/rsalabs/pkcs/pkcs-10/index.html
[CERT00]	*Securing Network Servers*, 2000, http://www.cert.org/security-improvement/modules/m10.html
[CERT01]	*Securing Public Web Servers*, 2001, http://www.cert.org/security-improvement/modules/m11.html
[Sca01]	Joel Scambray, et al, *Hacking Exposed Second Edition*, McGraw-Hll, 2001.
[Sch00]	Bruce Schneier, *Secrets & Lies: Digital Security in a Networked World*, John Wiley &Sons Inc., 2000
[SSL98]	*Introduction to SSL, Netscape Communication*, Corporation, 1998, http://developer.netscape.com/docs/manuals/security/sslin/contents.htm
[WWW01]	*WWW Security FAQ*, September 2001, http://www.w3.org/Security/Faq/.