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

September 2002

U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

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

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

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 of Booz Allen, for their research, technical support, and written contributions to this document. The authors would also like to express their thanks to all those who contributed input during the public comment period and who assisted with our internal review process.

Table of Contents

EXECUTIVE SUMMARY ES-1

1. INTRODUCTION 1
   - 1.1 AUTHORITY 1
   - 1.2 PURPOSE AND SCOPE 1
   - 1.3 AUDIENCE AND ASSUMPTIONS 2
   - 1.4 DOCUMENT STRUCTURE 2
2. WEB SERVER SECURITY PROBLEMS AND OVERVIEW 4
   - 2.1 GENERAL INFORMATION SYSTEM SECURITY PRINCIPLES 7
3. PLANNING AND MANAGEMENT OF WEB SERVERS 9
   - 3.1 PLANNING FOR A WEB SERVER DEPLOYMENT 9
   - 3.2 SECURITY MANAGEMENT STAFF 11
   - 3.3 MANAGEMENT PRACTICES 13
   - 3.4 SYSTEM SECURITY PLAN 14
   - 3.5 HUMAN RESOURCES FOR SECURING A WEB SERVER 16
   - 3.6 ALTERNATIVE WEB SERVER PLATFORMS 17
4. SECURING THE OPERATING SYSTEM 20
   - 4.1 SECURELY INSTALLING AND CONFIGURING AN OPERATING SYSTEM 20
   - 4.2 SECURITY TESTING THE OPERATING SYSTEM 24
   - 4.3 RESOURCES FOR OPERATING SYSTEM SPECIFIC SECURITY PROCEDURES 25
   - 4.4 SECURING THE WEB SERVER OPERATING SYSTEM CHECKLIST 25
5. SECURELY INSTALLING AND CONFIGURING THE WEB SERVER 27
   - 5.1 SECURELY INSTALLING THE WEB SERVER 27
   - 5.2 CONFIGURING ACCESS CONTROLS 28
   - 5.3 USING FILE INTEGRITY CHECKERS 34
   - 5.4 SECURELY INSTALLING AND CONFIGURING THE WEB SERVER CHECKLIST 35
6. SECURING WEB CONTENT 37
   - 6.1 PUBLISHING INFORMATION ON PUBLIC WEB SITES 37
   - 6.2 REGULATIONS REGARDING THE COLLECTION OF PERSONAL INFORMATION 39
   - 6.3 SECURING ACTIVE CONTENT AND CONTENT GENERATION TECHNOLOGIES 40
   - 6.4 SECURING WEB CONTENT CHECKLIST 49
7. AUTHENTICATION AND ENCRYPTION TECHNOLOGIES 52
   - 7.1 DETERMINING AUTHENTICATION AND ENCRYPTION REQUIREMENTS 52
   - 7.2 ADDRESS-BASED AUTHENTICATION 52
   - 7.3 BASIC AUTHENTICATION 52
   - 7.4 DIGEST AUTHENTICATION 53
   - 7.5 SSL/TLS 53

7.6 WEB AUTHENTICATION AND ENCRYPTION TECHNOLOGIES CHECKLIST 62

8. IMPLEMENTING A SECURE NETWORK FOR A WEB SERVER 64

8.1 NETWORK LOCATION 64
8.2 NETWORK ELEMENT CONFIGURATION 69
8.3 NETWORK INFRASTRUCTURE CHECKLIST 76

9. ADMINISTERING A WEB SERVER 78

9.1 LOGGING 78
9.2 WEB SERVER BACKUP PROCEDURES 82
9.3 RECOVERING FROM A SECURITY COMPROMISE 85
9.4 SECURITY TESTING WEB SERVERS 87
9.5 REMOTELY ADMINISTERING A WEB SERVER 90
9.6 SECURELY ADMINISTERING A WEB SERVER CHECKLIST 91

APPENDIX A. SECURING APACHE WEB SERVER A-1

A.1 INSTALLATION A-1
A.2 INITIAL CONFIGURATION A-4
A.3 PATCHING APACHE A-10
A.4 MAINTAINING A SECURE APACHE CONFIGURATION A-10

APPENDIX B. SECURING INTERNET INFORMATION SERVER B-1

B.1 IIS OVERVIEW B-2
B.2 IIS OVERVIEW B-2
B.3 WHERE TO INSTALL IIS B-4
B.4 CONFIGURING WINDOWS OPERATING SYSTEM B-4
B.5 CONFIGURING IIS B-9
B.6 CONFIGURING ACTIVE SERVER PAGES B-17
B.7 PATCHING IIS B-18
B.8 MAINTAINING SECURE IIS CONFIGURATION B-19
B.9 MICROSOFT PATCHING TOOLS B-20
B.10 MICROSOFT WINDOWS SECURITY CHECKLIST: B-36
B.11 MICROSOFT INTERNET INFORMATION SERVER SECURITY CHECKLIST B-37

APPENDIX C. ONLINE WEB SERVER SECURITY RESOURCES C-1

APPENDIX D. GLOSSARY D-1

APPENDIX E. WEB SECURITY TOOLS AND APPLICATIONS E-1

APPENDIX F. REFERENCES F-1

APPENDIX G. PRINT RESOURCES G-1

APPENDIX H. WEB SERVER SECURITY CHECKLIST H-1

INSTALLING AND CONFIGURING THE WEB SERVER SECURELY H-2
SECURING WEB CONTENT H-3
EMPLOYING WEB AUTHENTICATION AND ENCRYPTION TECHNOLOGIES H-5

EMPLOYING NETWORK INFRASTRUCTURE TO SECURE WEB SERVERS H-6
SECURELY ADMINISTERING A WEB SERVER H-7
SECURING APACHE H-9
SECURING MICROSOFT INTERNET INFORMATION SERVER.....H-10

APPENDIX I. ACRONYM LIST I-1

INDEX..... INDX-1

List of Figures

Figure 6.1: Relative Risk of Common Client Side Active Content.....43

Figure 7.1: SSL/TLS Location within the Internet Protocol Stack.....54

Figure 7.2: Sample CSR.....59

Figure 7.3: Sample Encoded SSL/TLS Certificate.....60

Figure 8.1: Basic DMZ 65

Figure 8.2: Two Firewall DMZ.....66

Figure 8.3: Three Interface Firewall DMZ.....66

Figure 8.4: Outsourced Web Server Hosting 68

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 focuses on the security issues of Web servers.[1]

Unfortunately, the Web server is the most targeted and attacked host on most organizations' network. As a result, it is essential 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 unauthorized access are gaining access to files or folders that were not meant to be publicly accessible or executing privileged commands and/or installing 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.
- Sensitive information on the Web server may be distributed to unauthorized individuals.
- Sensitive information that is not encrypted when transmitted between the Web server and the browser may be intercepted.
- Information on the Web server may be changed for malicious purposes. Web site defacement is a commonly reported example of this threat.
- 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 compromised Web server, concealing their actual identities, and perhaps making the organization from which the attack was launched liable for damages.
- The server may be used as a distribution point for illegally copied software, attack tools, or pornography, perhaps making the organization liable for damages.

This document is intended to assist organizations in installing, configuring, and maintaining secure public Web servers. More specifically, this document describes in detail the following practices to apply:

- Securing, installing, and configuring the underlying operating system

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

- Securing, installing, and configuring Web server software
- Deploying appropriate network protection mechanisms:
  - Firewalls
  - Routers
  - Switches
  - Intrusion detection systems (IDSs)
- 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 systematic manner.

The following key guidelines are recommended to Federal departments and agencies for maintaining a secure Web presence.

Organizations should carefully plan and address the security aspects of the deployment of any public Web server.

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 that addresses security. Establishing such a plan guides organizations in making the inevitable tradeoff decisions between usability, performance, and risk.

Organizations often fail to take into consideration the human resource requirements for both deployment and operational phases of the Web server and supporting infrastructure. Organizations should address the following points in a deployment plan:

- Types of personnel required (e.g., system and Web administrators, Webmaster, network administrators, information systems security officers [ISSO])
- Skills and training required by assigned personnel
- Individual (level of effort required of specific personnel types) and collective manpower (overall level of effort) requirements.

Organizations should implement appropriate security management practices and controls when maintaining and operating a secure Web presence.

Appropriate management practices are critical to operating and maintaining a secure Web server. Security practices entail the identification of an organization's information system assets and the development, documentation, and implementation of policies, standards, procedures, and guidelines that ensure confidentiality, integrity, and availability of information system resources.

To ensure the security of a Web server and the supporting network infrastructure, the following practices should be implemented:

- Organizational-wide information system security policy
- Configuration/change control and management
- Risk assessment and management
- Standardized software configurations that satisfy the information system security policy
- Security awareness and training
- Contingency planning, continuity of operations, and disaster recovery
- Certification and accreditation

Organizations should ensure that Web server operating systems are deployed, configured, and managed to meet the security requirements of the organization.

The first step in securing a Web server is securing the underlying operating system. Most 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. Securing the operating system at a minimum should include the following steps:

- Patch and upgrade the operating system
- Remove or disable unnecessary services and applications
- Configure operating system user authentication
- Configure resource controls
- Test the security of the operating system.

Organizations should ensure that the Web server application is deployed, configured, and managed to meet the security requirements of the organization.

In many respects, the secure installation and configuration of the Web server application mirrors the operating system process discussed above. 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. Securing a Web server application at a minimum should include the following steps:

- Patch and upgrade the Web server application
- Remove or disable unnecessary services, applications, and sample content
- Configure Web server user authentication
- Configure Web server resource controls
- Test the security of the Web server application and Web content.

Organizations should take steps to ensure that only appropriate content is published on Web site and that content is adequately protected from unauthorized alteration.

Many agencies lack 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 search for valuable information. Some generally accepted examples of what should not be published or at least should be carefully examined and reviewed before publication on a public Web site include:

- Classified or proprietary information
- Information on the composition or preparation of hazardous materials or toxins[2]
- Sensitive information relating to homeland security[2]
- An organization's detailed physical and information security safeguards
- Details about an organization's network and information system infrastructure (e.g., address ranges, naming conventions, access numbers)
- Information that specifies or implies physical security vulnerabilities
- Detailed plans, maps, diagrams, aerial photographs, and architectural drawings of organizational buildings, properties, or installations.

Organizations should ensure appropriate steps are taken to protect Web content from unauthorized access or modification.

While information on public web sites is content that is intended to be public, assuming a credible review process and policy is in place, it is still important to ensure that information cannot be modified without authorization. Users of this information rely upon the integrity of such information even if the information is not confidential. Because of the public accessibility, content on publicly accessible Web servers is inherently more vulnerable than information that is inaccessible from the Internet. This means that organizations need to protect public Web content through the appropriate configuration of Web server resource controls. Some examples of appropriate resource control practices include:

---

[2] For more guidance on protecting this type of information see the White Memorandum Dated March 19, 2000, *Action to Safeguard Information Regarding Weapons of Mass Destruction and Other Sensitive Documents Related to Homeland Security* (http://www.usdoj.gov/oip/foiapost/2002foiapost10.htm).

- Install or enable only necessary services.
- Install Web content on a dedicated hard drive or logical partition.
- Limit uploads to directories that are not readable by the Web server.
- Define a single directory for all external scripts or programs executed as part of Web content.
- Disable the use of hard or symbolic links.
- Define a complete Web content access matrix that identifies which folders and files within the Web server document directory are restricted and which are accessible (and by whom).
- Disable directory listings.
- Use user authentication, digital signatures, and other cryptographic mechanisms as appropriate.
- Use host-based intrusion detection systems and/or file integrity checkers to detect intrusions and verify Web content.

Organizations should use active content after carefully balancing the benefits gained against the associated risks.

In the beginning, most WWW sites presented static information residing on the server, typically in the form of text-based documents. Soon thereafter, interactive elements were introduced to offer users new ways to interact with a Web site. Unfortunately, these same interactive elements introduced new Web related vulnerabilities, since they involve moving code from a Web server to a client for execution. Different active content technologies have different associated vulnerabilities, which must be weighed against their benefits.[3]

Organizations must use authentication and cryptographic technologies as appropriate to protect certain types of sensitive data.

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. Web servers may be configured to use different cryptographic algorithms, providing varying levels of security and performance.

Without proper user authentication in place, organizations cannot selectively restrict access to specific information. All information that resides on a public Web server is then accessible by anyone with access to the server. In addition, without some process to authenticate the server, 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.

---

[3] See NIST Special Publication 800-28, *Guidelines for Active Content and Mobile Code* (http://csrc.nist.gov/publications/) for more extended discussion and advice on the policy and technical issues of active content.

Organizations should employ their network infrastructure to help protect their public Web server(s).

The network infrastructure that supports the Web server plays a significant role in the security of the Web server. In most configurations, the network infrastructure is the first line of defense between the Internet and a public Web server. 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 both layered and diverse defense mechanisms (i.e., defense in depth).

Organizations must commit to the ongoing process of maintaining the security of Public Web servers to ensure continued security.

Maintaining a secure Web server requires constant effort, resources, and vigilance from an organization. Thus, once a Web server has been initially deployed, administering it on a daily basis becomes a critical aspect of Web server security. Maintaining the security of a Web server usually involves the following steps:

- Configuring, protecting, and analyzing log files
- Backing up critical information frequently
- Maintaining a protected authoritative copy of the organization's Web content
- Establishing and following procedures for recovering from compromise
- Testing and applying patches in a timely manner
- Testing security periodically.

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.

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. While intended 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 to enhance security on Web server systems, and 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 8
- Security considerations related to Web client (browser) software[4]
- Special considerations for high traffic Web sites with multiple hosts[5].

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

- Securing backend servers that may support Web servers (e.g., database servers, file servers)
- Services other than Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS)
- Protection of intellectual property.

Information on security related topics is available on NIST's Computer Security Resource Center Web site (http://csrc.nist.gov).

1.3 Audience and Assumptions

The intended audience is varied. This document 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.

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

- System engineers and architects when designing and implementing Web servers
- Web and system administrators when administering, patching, securing, or upgrading Web servers
- Webmasters when creating and managing Web content
- 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 aspects of Web server operations.

This document assumes that readers have some minimal operating system, networking, and Web server 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.

---

[5] Although this document does not address the specific security concerns that arise from high traffic multiple server Web farms, much of what is covered will apply to these types of installations.

- Section 2 discusses Web server security problems and presents an overview.
- Section 3 provides information on planning and managing Web servers.
- Section 4 provides general information on choosing and securing the host operating system for a Web server.
- Section 5 discusses securely installing and configuring the Web server.
- Section 6 examines the security of Web content.
- Section 7 examines the popular Web authentication and encryption technologies.
- Section 8 discusses implementing a secure network for a Web server.
- Section 9 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.
- Appendix G provides a list of recommended printed Web Security resources.
- Appendix H provides a Web server security checklist.

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 it is not rigorous in configuring and operating its public Web site, an 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 efficiency, action at a distance, and rapid technique propagation [Sch00].

- **Increased Efficiency**. Automation makes attacks, even those with minimal opportunity for success, efficient and extremely profitable. For example, in the physical world an attack that would succeed once in 10,000 attempts would be ineffectual because of the time and effort required on the average for a single success. The time invested in getting a single success would be outweighed by the time invested in the 5,000 failures. On the Internet, automation enables the same attack to be a stunning success. Computing power and bandwidth are getting less expensive daily, while the number of hosts that can be targeted is growing rapidly. This synergy means that almost any attack, no matter how low its success rate, will likely find many systems to exploit.

- **Action at a Distance**. The Internet allows action at a distance. The Internet has no borders, and every point on the Internet is reachable from every other point. This means that an attacker in one country can target a remote web site in another country as easily as one close to 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 proprietary 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 host operating system, 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, thus concealing the intruders' identities, and perhaps making the organization liable for damages
  - Use the server as a distribution point for illegally copied software, attack 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.
- Poorly written software applications and scripts that allow attackers to compromise the security of the Web server.

A number of 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. 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 and maintenance processes (including server/application (updating and log reviews)

- Step 7. Conducting initial and periodic vulnerability scans of each public Web server and supporting network infrastructure (e.g., firewalls, routers).

The practices recommended in this document are designed to help mitigate the risks associated with public Web servers. 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 800-31, *Intrusion Detection Systems*
- NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*
- NIST Special Publication 800-34, *Contingency Planning Guide for Information Technology Systems*
- NIST Special Publication 800-37, *Federal Guidelines for the Security Certification and Accreditation of Information Technology Systems*
- NIST Special Publication 800-40, *Procedures for Handling Security Patches*
- NIST Special Publication 800-41, *Guide to Firewall Selection and Policy Recommendation*s
- NIST Special Publication 800-42, *Guideline on Network Security Testing*
- NIST Special Publication 800-43, *Guide to Securing Windows 2000 Professional*
- NIST Special Publication 800-46, *Security for Telecommuting and Broadband Communications*
- NIST Special Publication 800-52, *Guidelines for the Selection and Use of Transport Layer Security Implementations.*

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.

2.1 General Information System Security Principles

When addressing Web server security issues it is an excellent idea to keep some general information security principles in mind [Curt01 and Salt75]:

- **Simplicity** – Security mechanisms (and the information systems in general) should be as simple as possible. Complexity is at the root of many security issues.

- **Fail-Safe** – If a failure occurs, the system should fail in a secure manner. That is if a failure occurs, security should still be enforced. It is better to lose functionality rather than losing security.

- **Complete Mediation –** Rather than providing direct access to information, mediators that enforce access policy should be employed. Common examples include file system permissions, web proxies, firewalls, and mail gateways.

- **Open Design –** System security should not depend on the secrecy of the implementation or its components. "Security through obscurity" is not reliable.

- **Separation of Privilege –** Functions, to the degree possible, should be separate and provide as much granularity as possible. The concept can apply to both systems and operators/users. In the case of systems, such functions such as read, edit, write, and execute should be separate. In the case of system operators and users, roles should be as separate as possible. For example if resources allow, the role of system administrator should be separate from that of the security administrator.

- **Least Privilege** – This principle dictates that each task, process, or user is granted the minimum rights required to perform its job. By applying this principle consistently, should a task, process, or user be compromised, the scope of damage is constrained to the limited resources available to the compromised entity.

- **Psychological Acceptability –** Users should understand the necessity of security. This can be provided through training and education. In addition, the security mechanisms in place should present users with sensible options that give them the usability they require on a daily basis. If users find the security mechanisms too cumbersome, they may devise ways to work around or compromise them. For example, randomly generated passwords are strong but users weaken their security by writing down difficult to remember randomly generated passwords. The objective is not to weaken security so it is understandable and acceptable, but to train, educate, and design security mechanisms and polices that are usable and effective.

- **Least Common Mechanism –** When providing a feature to the system, it is best to have a process or service gain some function without granting the same function to other parts of the system. The ability for the Web server process to access a back-end database, for instance, should not also enable other applications on the system to access the back-end database.

- **Defense in Depth –** Organizations should understand that a single security mechanism would generally prove insufficient. Security mechanisms (defenses) need to be layered so that compromise of a single security mechanism is insufficient to

compromise a host or network. Effective security solutions recognize no "silver bullet" exists for information system security.

- **Work Factor –** Organizations should understand what it would take to break the system or network's security features. The amount of work necessary for an attacker to break the system or network should exceed the value that the attacker would gain from a successful compromise.

- **Compromise Recording –** Records and logs should be maintained so that if a compromise does occur, evidence of the attack is available to the organization. This information can assist in securing the network and host after the attack and assist in identifying the methods and exploits used by the attacker. This information can be used to better secure the host or network in the future. In addition, this can assist organizations in identifying and prosecuting attackers.

3. Planning and Management of Web Servers

The most critical aspect of deploying a secure Web server is careful planning prior to installation, configuration, and deployment. Careful planning will ensure that the Web server is as secure as possible and in compliance with all relative organizational policies. Many Web server security and performance problems can be traced back to a lack of planning or management controls. The importance of management controls is difficult to overstate. In many organizations, the information technology support structure is highly fragmented. This fragmentation often leads to inconsistencies, which then lead to security vulnerabilities and other issues.

3.1 Planning for a Web Server Deployment

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 enables organizations to make informed tradeoff decisions between usability and performance, and risk. A deployment plan allows an organization to maintain a secure configuration and aids in identifying security vulnerabilities, which often manifest themselves as deviations from the plan.

Organizations planning for the development of a Web server should consider the following steps [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[6]
- Secure Hypertext Transfer Protocol (S-HTTP)[7]
- Remote Authentication Dial-In User Service (RADIUS) Protocol
- Open Database Connectivity (ODBC) Protocol
- Network File System (NFS) Protocol[8]
- Common Internet File 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.
- 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 (see NIST ICAT vulnerability database [http://icat.nist.gov] for the known vulnerabilities of specific Web server applications)
- Ability to restrict administrative or root level activities to authorized users only
- Ability to control access to data on the server
- 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

---

[6] HTTP transactions protected via the Secure Socket Layer (SSL)/Transport Layer Security (TLS) protocols (see Section 7.5).

[7] A seldom-used alternative to HTTPS.

[8] Not generally recommended unless used exclusively within a protected network environment such as a Demilitarized Zone (DMZ) (see Section 8.1.2) for data replication to multiple Web servers.

- Ability to log appropriate server activities to detect intrusions and attempted intrusions
- Provision of a host-based firewall capability

In addition, organizations should consider the availability of trained experienced staff to administer the server and server products. Many organizations have learned the difficult lesson that a capable and experienced administrator on one type of operating environment is not automatically as effective on another.

Given the sensitive nature of the Web server, it is critical that it be located in an area that provides a secure physical environment. When planning the location of the Web server the following items should be considered:

- Does the proposed location offer the appropriate physical security protection mechanisms? Examples include:
  - Locks
  - Card-reader access
  - Security guards
  - Physical intrusion detection systems (e.g., motion sensors, cameras)
- Does the proposed location offer the appropriate environmental controls so that the necessary humidity and temperature are maintained?
- Is there a backup power source?

3.2 Security Management Staff

Because Web server security is tightly intertwined with the organization's general information system security posture, a number of IT and system security staff may be interested in Web server planning, implementation and administration. This section provides a list of those roles and identifies their responsibilities as related to Web server security. These roles may vary with the organization, however, and not all organizations will have the identical roles described here.

3.2.1 Senior IT Management/Chief Information Officer (CIO)

The Senior IT Management/CIO ensures that the organization's security posture is adequate. The Senior IT Management provides direction and advisory services for the protection of information systems for the entire organization. The Senior IT Management/CIO is responsible for the following activities that are associated with Web servers:

- Coordinating the development and maintenance of the organization's information security policies, standards, and procedures
- Coordinating the development and maintenance of the organization's change control and management procedures

- Ensuring the establishment of, and compliance with, consistent IT security policies for departments throughout the organization
- Coordinating with upper management, public affairs, and other relevant personnel to produce a formal policy and process for publishing information to Web sites and ensuring this policy is enforced

3.2.2 Information Systems Security Program Managers

The Information Systems Security Program Managers oversee the implementation of, and compliance, with the standards, rules, and regulations specified in the organization's security policy. The ISSMs are responsible for the following activities associated with Web servers:

- Developing and implementing standard procedures (security policy)
- Complying with security policies, standards, and requirements
- Ensuring that all critical systems are identified and that contingency planning, disaster recovery plans, and continuity of operations plans exist for these critical systems
- Ensuring that critical systems are identified and scheduled for periodic security testing according to the security policy requirements of each respective system.

3.2.3 Information Systems Security Officers

Information Systems Security Officers (ISSOs) are responsible for overseeing all aspects of information security within a specific organizational entity. They ensure that the organization's information security practices comply with organizational and departmental policies, standards, and procedures. ISSOs are responsible for the following activities associated with Web servers:

- Developing internal security standards and procedures for the Web server(s) and supporting network infrastructure
- Cooperating in the development and implementation of security tools, mechanisms, and mitigation techniques
- Maintaining standard configuration profiles of the Web servers and supporting network infrastructure controlled by the organization, including but not limited to, operating systems, firewalls, routers, and Web server applications
- Maintaining operational integrity of systems by conducting security tests and ensuring that designated IT professionals are conducting scheduled testing on critical systems.

3.2.4 Web and Network Administrators

Web administrators are system architects responsible for the overall design, implementation, and maintenance of a Web server. Network administrators are responsible for the overall design, implementation and maintenance of a network. Daily, Web and network administrators must address the security requirements of the specific system(s) for which they are responsible. Security issues and solutions can originate from either outside (e.g., security

patches and fixes from the vendor or computer security incident response teams) or within the organization (e.g., the Security Office). The administrators are responsible for the following activities, associated with Web servers:

- Install and configure systems in compliance with the organizational security policy(ies) and standard system/network configurations
- Maintain systems in a secure manner, through frequent backups, timely application of patches
- Monitoring system integrity, protection levels, and security related events
- Following-up with detected security anomalies associated with their information system resources
- Conducting security tests as required.

3.3 Management Practices

Appropriate management practices are the most critical to operating and maintaining a secure Web server. Security practices entail the identification of an organization's information system assets and the development, documentation, and implementation of policies, standards, procedures and guidelines that ensure confidentiality, integrity, and availability of information system resources.

To ensure the security of a Web server and the support network infrastructure, the following practices should be implemented:

- **Organizational Information System Security Policy** – A security policy should outline who in the organization is responsible for particular areas of information security (e.g., implementation, enforcement, audit, review). The policy should also specify what the basic information system security policies are and their intended internal purpose. The policy must be enforced consistently throughout the organization in order to be effective. Generally the CIO and upper management are responsible for drafting the organization's security policy.
- **Configuration/Change Control and Management** – Is the process of controlling modification to a system's design, hardware, firmware and hardware which provides sufficient assurance the system is protected against the introduction of an improper modification prior to, during, and after system implementation. Configuration control leads to consistency with the organization information system security policy. Configuration control is traditionally overseen by a configuration control board that is the final authority on all proposed changes to an information system.
- **Risk Assessment and Management** – Risk assessment is the process of analyzing and interpreting risk. It involves determining the assessment's scope and methodology, collection and analyzing risk related data, and interpreting the risk analysis results. Collecting and analyzing risk data requires identifying assets, threats, vulnerabilities, safeguards, consequences, and the probability of a successful attack. Risk management is the process of selecting and implementing of controls to reduce risk to a level acceptable to the organization.

- **Standardized Configurations** – Organizations should develop standardized secure configurations for widely used operating systems and applications. This will provide guidance to Web and network administrators on how to securely configure their systems and ensure consistency and compliance with the organizational security policy. Because it only takes one insecurely configured host to compromise a network, organizations with a significant number of hosts are especially encouraged to apply this recommendation.

- **Security Awareness and Training** – A security training program is critical to the overall security posture of an organization. Making users and administrators aware of their security responsibilities and teaching the correct practices helps them change their behavior to conform to security best practices. Training also supports individual accountability, which is an important method for improving information system security.

- **Contingency Planning, Continuity of Operations and Disaster Recovery Planning** – Contingency planning, continuity of operations and disaster recovery planning are plans setup in advance to allow an organization or facility to maintain operations in the event of a disruption to their organization.[9]

- **Certification and Accreditation** – Certification in the context of information systems security means that a system has been analyzed as to how well it meets all of the security requirements of the organization. Accreditation occurs when the organization's management accepts that the system meets the organization's security requirements.[10]

3.4 System Security Plan

The objective of computer security planning is to protect information assets (i.e., information and information resources).[11] Plans that adequately protect information assets require managers and information owners – directly affected by and interested in the information and/or processing capabilities – to be convinced that their information assets are adequately protected from loss, misuse, unauthorized access or modification, unavailability, or undetected activities.

The system security plan provides a basic overview of the security and privacy requirements of the subject system and the organization's plan for meeting those requirements. The system security plan is also perceived as way of documenting the structured process of planning adequate, cost-effective security protection for a system. Consequently, the system security plan should reflect input from various managers with responsibilities concerning the system, including functional end-users or information owners, system operations, and system security manager.

---

[9] For more information see NIST Special Publication 800-34, *Contingency Planning Guide for Information Technology Systems* (http://csrc.nist.gov/publications/)

[10] For more information on certification and accreditation see NIST Special Publication 800-37, *Federal Guidelines for the Security Certification and Accreditation of Information Technology Systems* (http://csrc.nist.gov/publications/)

[11] For more information on system security plans, see NIST Special Publication 800-18, *Guide for Developing Security Plans for Information Technology Systems* (http://csrc.nist.gov/publications/) (http://csrc.nist.gov/publications/nistpubs/index.html).

All important information systems should be covered by a system security plan. For Federal agencies the completion of system security plans is a requirement of the Office of Management and Budget (OMB) Circular A-130, "Management of Federal Information Resources," Appendix III, "Security of Federal Automated Information Resources," updated in 1996, and of Public Law 100-235, "Computer Security Act of 1987." Other organizations should strongly consider the completion a system security plan for each of their systems as well [NIST98].

For Federal agencies, OMB Circular A-130, Appendix III, does not distinguish between sensitive and non-sensitive systems. Rather, consistent with the Computer Security Act of 1987, the Circular recognizes that federal automated information systems have varied sensitivity and criticality. All federal systems have some level of sensitivity and require protection as part of good management practice.

The purposes of system security plans are to [NIST98]:

- Provide an overview of the security requirements of the system and describe the controls in place or planned for meeting those requirements; and

- Delineate responsibilities and expected behavior of all individuals who access the system.

The System Owner[12] is generally responsible for ensuring that the security plan is prepared and for implementing the plan and monitoring its effectiveness. Security plans should reflect input from various individuals with responsibilities concerning the system, including functional "end users," Information Owners[13], the System Administrator, and the System Security Manager.

In general, an effective system security plan should include the following [NIST98]:

- **System Identification** – The first section of the system security plan provides basic identifying information about the system. It contains general descriptive information regarding who is responsible for the system, the purpose of the system, and the sensitivity level of the system.

- **Management Controls** – This section, describes the management control measures (in place or planned) that are intended to meet the protection requirements of the information system. Management controls focus on the management of the computer

[12] The System Owner is responsible for defining the system's operating parameters, authorized functions, and security requirements. The information owner for information stored within, processed by, or transmitted by a system may or may not be the same as the System Owner. Also, a single system may utilize information from multiple Information Owners.

[13] The Information Owner is responsible for establishing the rules for appropriate use and protection of the subject data/information (rules of behavior). The Information Owner retains that responsibility even when the data/information are shared with other organizations.

security system and the management of risk for a system. The types of control measures shall be consistent with the need for protection of the system.[14]

- **Operational Controls** – This section addresses security methods that focus on mechanisms that primarily are implemented and executed by people (as opposed to systems). These controls are put in place to improve the security of a particular system (or group of systems). They often require technical or specialized expertise – and often rely upon management activities as well as technical controls. This section describes the operational control measures (in place or planned) that are intended to meet the protection requirements of the information system.

- **Technical Controls** – Technical controls focus on security controls that the computer system executes. The controls can provide automated protection from unauthorized access or misuse, facilitate detection of security violations, and support security requirements for applications and data. The implementation of technical controls, however, always requires significant operational considerations and should be consistent with the management of security within the organization. This section describes the technical control measures (in place or planned) that are intended to meet the protection requirements of the major application.

3.5 Human Resources for Securing a Web Server

The greatest challenge and expense in developing and securely maintaining an public Web presence is providing the necessary human resources to adequately perform the required functions. Many organizations fail to fully realize the amount of expense and skills required to field a secure public Web server. This failure often results in overworked employees and insecure systems. From the initial planning stages, an organization needs to determine the necessary human resource requirements. Appropriate and sufficient human resources are the single most important aspect of Web server security. Organizations should also consider the fact that in general technical solutions do NOT substitute for skilled and experienced personnel.

When considering the human resource implications of developing and deploying a Web server, organizations should consider the following:

- **Required Personnel** – What types of personnel are going to be required? This would include such positions as system and Web administrators, Webmasters, network administrators, ISSOs, etc.

- **Required Skills** – What are the required skills to adequately plan, develop and maintain the Web server in a secure manner? Examples would include, operating system administration, network administration, active content expertise, programming, etc.

- **Available Personnel** – What are the available human resources within the organization? In addition, what are their current skill sets and are they sufficient for

[14] For more detail on management controls, see NIST Special Publication 800-12, *An Introduction to Computer Security: The NIST Handbook.*

supporting the Web sever? Often an organization will discover that their existing human resources are not sufficient and will have to consider the following options:

- Hire Additional Staff, – If there is not enough staff available or they do not have the requisite skills, it may be necessary to hire additional personnel.

- Train Current Staff, – If there are personnel available but they do not have the requisite skills, the organization may choose to train the existing staff in the skills required. While this is an excellent option, the organization should ensure that employees meet the necessary prerequisites for training.

Once the organization has staffed the project and the Web server is active it will be necessary to ensure the number and skills of the personnel are still adequate. The threat and vulnerability levels of IT systems including Web servers are constantly changing, as is the technology. This means that what is adequate today may not be tomorrow.

3.6 Alternative Web Server Platforms

Although many organizations manage Web servers that operate over general-purpose operating systems, there are instances when an organization may wish to use one of the alternative discussed below. While these technologies are relatively new to the area of Web servers, they are based on sound technologies and will probably see broader use in the Web server environment in the years to come.

3.6.1 Trusted Operating Systems

Trusted operating systems (TOSes) are security modified or enhanced operating systems that include additional security mechanisms not found in most general-purpose operating systems. They were originally created to meet the need of Federal government for high security mandatory access control systems. TOSes provide a very secure system-wide control policy, a finely defined set of access privileges, and extensive logging and auditing capabilities. Most TOSes are independently verified to ensure that meet the requirements set forth in their design, documentation.

TOSes are generally used in applications where security is paramount. TOSes are able to securely control all aspects of a computing environment including networking resources, users, processes, memory, etc. More specifically, TOSes are able to limit access to system resources, and in a manner that not likely to be interfered with or compromised.

Applying a TOS will generally lead to a very secure Web server, however, some difficulties exist in using TOSes. A major drawback is that administering a TOS requires knowledge of each protected subsystem and its access needs. It may also require significant planning and administrative overhead to design and support a complex Web site on a TOS. However even with these limitations, organizations that have very high security requirements should consider using a TOS to support their Web server(s).

Some items to consider when considering a Web platform:

- What is the underlying operating system and how has it fared in security testing?

- Does the organization have the expertise in administering a TOS?

- Are the additional costs of purchasing and supporting a TOS outweighed by the benefits?

- Is the TOS compatible with the organization's existing Web applications and scripts?

3.6.2 Web Server Appliances

A relatively recent development in the area of Web servers has been the development of Web appliances. A Web appliance is a software/hardware combination that is designed to be a "plug-and-play" Web server. These appliances employ the use of a simplified operating system that is optimized to support a Web server. The simplified operating system improves security by minimizing unnecessary features, services, and options. The Web server application on these systems is often pre-hardened and pre-configured for security.

These systems offer other benefits in addition to security. Performance is often enhanced since the system (operating system, Web server application, and hardware) are designed and built specifically to operate as a Web server. Cost is often reduced since hardware and software is not specifically required by a Web server is included. These systems can be an excellent option for smaller to mid-sized organizations that cannot afford a full-time Web administrator.

The greatest weakness in these systems is that they are not suitable for large complex and multi-layered Web sites. They may also be inappropriate for organizations that require more than one server, unless the organization is willing to purchase Web appliances from a single vendor, since their simplicity makes it difficult to configure Web appliances from different vendors to work together. Web appliances are available from most major hardware vendors and from a variety specialized vendors who concentrate solely on Web appliances.

Some items to consider when contemplating the purchase of a Web appliance:

- What is the underlying operating system and how has it fared in security testing?

- How has the Web appliance itself fared in security testing? (Note that the configuration options of Web appliances are necessarily limited so a Web appliance will generally only be as secure as its default install.)

- How heterogeneous is the organization's Web server infrastructure? (Different brand of Web appliances do not generally work well together.)

- Are the limited expansion options inherent in Web appliances acceptable to the organization? (Organizations who are anticipating or experiencing rapid growth in Web traffic may not wish to limit themselves to a Web appliance.)

3.6.3 Pre-hardened Operating Systems and Web Servers

There are a growing number of pre-hardened operating system and Web server packages being distributed today. These packages include an operating system and Web server application that are modified and pre-configured to provide high security. Some of these packages include the hardware platform while others are software distributions that include only the operating system and Web server application. These distributions are generally based on hardened and/or modified general-purpose operating systems (e.g., Linux, Unix, and, less often, Windows) that are specifically designed to support a secure Web server. The Web server

application is also often based on a hardened and/or modified generally available Web server application (e.g., Apache or IIS). These packages often include a greater number of security options and are designed to be easier to configure through the use of precompiled scripts and graphical user interfaces (GUIs). Although each of these packages is different, they usually rely on one or more of the following to provide a higher level of protection and security:

- Secure initial default configuration
- Hardened operating system and/or TOS
- Hardened Web server software
- Extensive auditing capabilities
- Application wrappers
- Network wrappers and/or host-based firewall capabilities
- Host-based intrusion detection systems
- Simplified security administration (e.g., menus, GUIs).

These types of systems should be considered by organizations that face a significant threat level and/or have high value Web sites (e.g., major Federal government organizations, banks, health insurance companies). These packages are available from some major hardware and software vendors in addition to a variety specialized vendors.

Some items to consider when contemplating the purchase of a hardened Web appliance:

- What is the underlying operating system and how has it fared in security testing?
- How has the Web server application itself fared in security testing?
- How difficult is it to administer?
- Is the hardened Web server application and operating system compatible with the organization's existing Web applications and scripts?

4. Securing the Operating System

The first step in securing a Web server is securing the underlying operating system. Most 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 Web administrators configure and deploy Web servers that satisfy their organization's security requirements. Web administrators with existing Web servers should confirm that their current configurations address the issues discussed here.

Given that the techniques for hardening different operating systems vary greatly, this section will include the generic procedures common in securing most operating systems. References for securing specific operating systems are provided in Section 4.3. 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 (see Section 3.1)
- 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.

4.1 Securely Installing and Configuring an Operating System

4.1.1 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:

- Create and implement a patching process
- Identify vulnerabilities and applicable patches[15]
- Mitigate vulnerabilities (until patches are available, tested, and installed)

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

- Install permanent fixes (often called patches, hotfixes, service packs, or updates).[16]

4.1.2 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 unneeded services. Some common examples of services that should usually be disabled would include:

- Windows Network Basic Input/Output System (NetBIOS), if not required
- NFS, if not required
- File Transfer Protocol (FTP)
- Berkeley "r" services (e.g., rlogin, rsh, rcp)
- 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]:

- Unnecessary 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, having a single administrator responsible for a host provides better accountability.

[16] For more information on vulnerabilities and patching, see NIST Special Publication 800-40, *Procedures for Handling Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

- 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, preferably, remove 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 when 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, comes with an increased risk to the server. Whether the risks outweigh the benefits is a decision each organization must make for itself.

4.1.3 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 prospective users by requiring proof that each person 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 7).

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. 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 to malicious entities.

- **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. This approach is preferable to assigning rights to individual users.

- **Create the user accounts**. Identify 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**. Set account passwords appropriately. This policy should address the following:

  - **Length** – a minimum length for passwords.

  - **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 to 120 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 9.5), 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 to reflect in the way in which they are applied.

- **Generate and distribute user account reports.** In order to ensure that all unnecessary accounts are removed in a timely matter, it is critical for the organization to setup a system that generates reports of the user accounts that include information necessary to determine whether or not he account should remain active. These reports should be disseminated to appropriate supervisors and management personnel to identify individuals who no longer require accounts.

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)/Transport Layer Security (TLS) (see Section 7.5).

4.1.4 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 5.2.

4.2 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. Of the several methods for testing operating systems, the most popular 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 an "attacker." It is an iterative testing process that identifies the weakest areas of the network and exploits them to expand access to the remainder of the network. The process eventually results in compromising the overall security of the network. Vulnerability scanning should be conducted periodically, at least weekly to monthly, and penetration testing

should be conducted at least annually. Since both of these testing techniques are applicable to testing the Web server application as well, they are discussed in greater detail in Section 9.4.[17]

4.3 Resources for Operating System Specific Security Procedures

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

- **Solaris** – Paul D. J. Vandenberg and Susan D. Wyess, *Securing Solaris Servers* (http://www.usenix.org/sage/sysadmins/solaris/)
- **Unix** – CERT, *Unix Security Checklist Version 2.0* (http://www.cert.org/tech_tips/usc20_full.html)
- **Windows NT** – National Security Agency (NSA), *Guide to Securing Microsoft Windows NT Networks* (http://nsa1.www.conxion.com/winnt/guides/wnt-1.pdf)
- **Windows 2000** – NIST 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* (including 2000 Server) (http://nsa1.www.conxion.com/win2k/index.html).

4.4 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

[17] 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).

Completed	Action
☐	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
☐	Availability of 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 frequently to determine new vulnerabilities

5. 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.

5.1 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. From the Web server application root directory, remove all files that are not part of the Web site
7. Remove all sample documents, scripts, and executable code
8. Remove all vendor documentation from server
9. Apply appropriate security template or hardening script to server (see Appendix E)
10. 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.)

5.2 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 resource permissions can be set at both the operating system and Web server application, it is important that they are identical otherwise 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.

5.2.1 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 service process(es) is (are) configured to run as a user with a strictly limited set of privileges (i.e., not running as root, Administrator, or equivalent).
- Web content files can be read but not written by Web service process(es).
- Web service process(es) cannot write the directories where public Web content is stored.
- Only process(es) authorized for Web server administration can write Web content files.
- The Web server 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 the 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 Web server application 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 or the 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.
- If uploads are allowed to the Web server, these files should not be readable by the Web server. They should only be readable by the Web server after some automated or manual review process. This prevents the Web server from being used to traffic pirated software, attack tools, pornography, etc.
- 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 9.1).

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 illegitimate 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.

5.2.2 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 content (e.g., CGI, Active Server Page [ASP], Hypertext Preprocessor [PHP]).

- 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.

- 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 7).

5.2.3 Uniform Resource Identifiers and Cookies

Uniform Resource Identifiers (URIs) are the address technology from which URLs are created. Technically URLs (e.g., http://www.mywww.gov) are a subset of URIs. There are a number of security issues that arise from URIs. Since URIs are sent in the clear, any data that is stored within them can be easily compromised. For example, URIs are recorded in numerous location including Web browser logs (i.e., browser history), proxy server logs, and third-party HTTP referrer logs. Thus hiding sensitive data such as usernames and passwords, or hidden server resources in URI is not recommended. Security through obscurity is not secure.

URIs are often included with public web content. Although those these URIs may not display as Web content in a user's Web browser, they can be easily discovered in the source code. Therefore no publicly served Web content should include sensitive URI hidden in the source code. Many attackers and malicious bots (see Section 5.2.4) will search the source code for sensitive URI information including:

- E-mail addresses
- Images on other servers
- Links to other servers
- Particular text expressions (e.g., userid, password, root, administrator)
- Hidden form values
- Hyperlinks

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. The intent of cookies is to allow servers to recognize a specific browser (user). In essence, they add state to the stateless HTTP protocol. Unfortunately cookies are usually sent in the clear and are stored in the clear on the user's host and so are vulnerable to compromise. There are known vulnerabilities in certain versions of Internet Explorer for example that allow a malicious Web site to remotely collect all a visitor's cookies without the visitor's knowledge. Therefore, cookies should never contain data that can be used directly by an attacker (e.g., username, password).

5.2.4 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 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 AltaVista 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.[18]

The 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/
```

[18] 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: /_borders/
Disallow: /_fpclass/
Disallow: /_overlay/
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 7 for more information on Web based authentication methods.

5.3 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 8.2.2) 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.

5.4 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

Completed	Action
	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)
☐	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 files 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

6. 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 7). The less obvious component of content security is compromised caused by the way particular types of content are processed on a server can lead to a compromise.

6.1 Publishing Information on Public Web Sites

Little thought is usually 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, attackers often read the contents of a target organization's Web site to gather intelligence before any attacks [Sca01].

Absent compelling reasons, a public Web site should generally not contain the following information:

- Classified records
- Internal personnel rules and procedures
- Sensitive or proprietary information
- Personal information about an organization's personnel[19]
    - Home addresses and phone numbers
    - Social Security Numbers (SSNs)
    - Detailed biographical material (could be employed for social engineering)
    - Staff family members
- Telephone numbers, e-mail addresses[20], or general listings of staff unless necessary to fulfill organizational requirements

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

- Schedules of organizational principals or their exact location (whether on or off the premises)
- Information on the composition or preparation of hazardous materials or toxins[21]
- Sensitive information relating to homeland security[13]
- Investigative records
- Financial records (beyond those already publicly available)
- Medical records
- Organization's physical and information security procedures
- Information about organization's network and information system infrastructure (e.g., address ranges, naming conventions, access numbers)
- Information that specifies or implies physical security vulnerabilities
- Plans, maps, diagrams, aerial photographs, and architectural plans of organizational building, properties, or installations
- Information on disaster recovery or continuity of operations plans except as absolutely required
- Details on emergency response procedures, evacuation routes, or organizational personnel responsible for these issues
- Copyrighted material without the written permission of the owner
- Privacy or security policies that indicate the types of security measures in place to the degree that they may be useful to an attacker

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:

---

[20] When an e-mail address must be published on a Web site, consider the use of generic email addresses or aliases (e.g., webmaster@mydomain.gov as opposed to jane_doe@mydomain.gov). There are two reasons to do this. One, published email addresses are much more likely to receive unsolicited bulk e-mal (i.e., "spam"). Two, personally identifying email addresses can provide useful information to an attacker (e.g., possible usernames or as a information to be used for social engineering attempts).

[21] For more guidance on protecting this type of information see the White Memorandum Dated March 19, 2000, *Action to Safeguard Information Regarding Weapons of Mass Destruction and Other Sensitive Documents Related to Homeland Security* (http://www.usdoj.gov/oip/foiapost/2002foiapost10.htm).

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, publishing, and maintaining 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.
10. Periodically review published information to confirm continued compliance with organizational guidelines.

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.

6.2 Regulations Regarding the Collection of Personal Information

Federal and state laws and regulations apply to 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. Governmental organizations with Web sites should be aware of the appropriate and applicable laws, regulations, and agency guidelines. Private organizations may wish to use these 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 do apply to commercial organizations that operate Web sites on behalf of federal agencies. Organizations should be aware of changes to legal, regulatory, and contractual issues and seek 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 personal information includes the following:

- Name

- 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 a user's hard drive when a Web site is visited. 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 are 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.

6.3 Securing Active Content and Content Generation Technologies

In the early days of the WWW, most sites presented textual static information based on the American Standard Code of Information Interchange (ASCII). No interactivity existed between the user and Web site beyond the user clicking on hyperlinks. Soon thereafter, interactive elements were introduced that offered users new ways to interact with the Web site. Unfortunately, these interactive elements introduced a raft of new Web related vulnerabilities.[22]

Active content refers to interactive elements processed at the client (Web browser). If not implemented correctly, active content 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

[22] 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).

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 malicious entity 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.

6.3.1 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 aware that even if their content does not present a threat to the user, active content from other sites may present a threat, and that the user may not 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 type 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. This situation 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].

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].

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.

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, ActiveX controls are digitally signed by their author under a technology scheme called Authenticode. The digital signatures are verified using identity certificates issued by a trusted certificate authority to an ActiveX software publisher. For an ActiveX publisher's certificate to be granted, the software publisher must pledge that no harmful code will be knowingly distributed under this scheme. 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 [NIST01a].

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. Most users, are probably 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 6.1 shows the relative risk of ActiveX compared with other popular client side active content technologies [NIST01a].

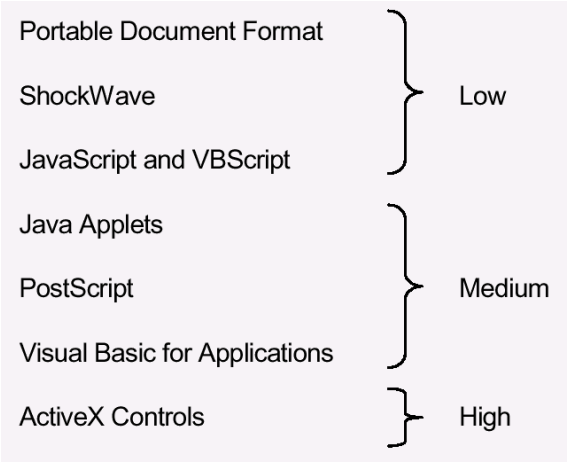


Figure 6.1: Relative Risk of Common Client Side Active Content

6.3.2 Server Side Content Generation Technologies and Related Vulnerabilities

Unlike the above technologies, CGI, ASP, and other similar server interfaces fall on the (Web) server side of the client-server model. Common uses of server side execution include [Zir02]:

- Database access
- E-commerce/E-government applications
- Chart rooms
- Threaded discussion groups.

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].

Server side content generators can create security vulnerabilities at the server:

- They may intentionally or unintentionally leak information about the Web server application and host operating 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.
- Allow attackers to deface or modify site content.

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. A script 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 so this option should only be used as a last resort. 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 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]. Some issues to consider when contemplating the deployment of CGI [Zir02]:

- Host file system (see Section 4.1) provides security for CGI.

- Most servers allow per-directory CGI restrictions.
- CGI itself provides little security enforcement.
- Perl facilitates secure programming most other languages (e.g., C, C++, sh) do not.
- CGI wrappers available from third parties offer additional protection for CGI.

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]. Some issues to consider when contemplating the deployment of SSIs:

- The security of SSIs is extremely weak if the exec command is enabled on the Web server.
- The impact of SSIs can hurt the performance of heavily loaded Web servers.
- The Security of SSIs relies heavily on host operating system and the Web server application for security.

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 6.3.1) is an optional technology and not required by ASPs [NIST01a].

Some issues to consider when contemplating the deployment of ASP [Zir02]:

- ASP relies heavily on host operating system and the Web server application for security.

- Client security is well integrated with Web server and host operating system authentication services.
- ASP does not support security policy enforcement so there is no method for the content developer or administrator to constrain privileges.
- Relatively immune to buffer overflows.
- Often uses Component Object Model (COM) objects which may have weak security
- Well-documented and mature technology.

Java Servlets – Servlets are based on Java technology (see Section 6.3.1) and are a type 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].

Some issues to consider when contemplating the deployment of Java Servlets [Zir02]:

- Tightly integrated with host operating system security and Web server authentication for strong security.
- Facilitates secure programming:
  - Leverages security of Java language
  - Strong security model supporting constraints by developers and server administrators
  - Secure error handling
- Well-documented and mature technology.

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]. Some issues to consider when contemplating the deployment of PHP:

- Old versions of PHP have numerous security vulnerabilities, upgrade to latest version.

- Excellent performance.
- Highly configurable which may make it difficult for novices to secure.
- Much of the freely available third-party code for PHP is poorly written from a security perspective.[23]

6.3.3 Server Side Content Generator Security Considerations

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

- Ensure that the executable code is as simple as possible. The longer or more complex it is, the more likely it is to have problems.
- Limit the executable code's ability to read and write programs. Code that reads files may inadvertently violate access restrictions or pass sensitive system information. Code that writes files may modify or damage documents or introduce Trojan horses.
- Analyze the code's interaction with other programs or applications. For example, many CGI scripts send e-mail in response to form input by opening up a connection with the sendmail program. Ensure this interaction is performed in a secure manner.
- On Linux/Unix hosts, the code should not run with suid (set-user-id).
- The code should use explicit path names when invoking external programs. Relying on the PATH environment variable to resolve partial path names is not recommended.
- Web servers (whether or not they employ active content) should be scanned periodically for vulnerabilities (see Section 9.4.1). There are several scanners that specifically scan for content generator vulnerabilities (e.g., Whisker and Retina, see Appendix E).
- 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

[23] There are a number of reasons for the poor security of many PHP scripts. The most obviously being that many scripts are written without security being a concern. In addition due to the relative ease of coding PHP scripts many novices who have little knowledge of secure programming create (and often freely distribute) poorly written (from a security prospective) scripts. PHP can be very secure as long as an experienced programmer codes it with security as a primary consideration.

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 7.5).

- 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.

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.

6.3.4 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:

- Writable files should be identified and placed in separate folders. No script files should exist in writable 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 writable 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. Note much of the risk with include files is in their execute capability. If the execute capability is disabled this risk is drastically reduced.

6.4 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 records
☐	Internal personnel rules and procedures
☐	Sensitive or proprietary information
☐	Personal information about an organization's personnel
☐	Telephone numbers, e-mail addresses, or general listings of staff unless necessary to fulfill organizational requirements
☐	Schedules of organizational principals or their exact location (whether on or off the premises)
☐	Information on the composition, preparation, or optimal use of hazardous materials or toxins
☐	Sensitive information relating to homeland security
☐	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
☐	Information that specifies or implies physical security vulnerabilities
☐	Plans, maps, diagrams, aerial photographs, and architectural plans of organizational building, properties, or installations

Completed	Action
☐	Copyrighted material without the written permission of the owner
☐	Privacy or security policies that indicate the types of security measures in place to the degree that they may be useful to an attacker
	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, publishing, and maintaining 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
☐	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

Completed	Action
☐	SSIs are disabled or execute function is 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

7. 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, organizations 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, without some process to authenticate the server, 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.

7.1 Determining Authentication and Encryption Requirements

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

For information that requires some level of user authentication, the organization should determine which of the following technologies or methods would 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.

7.2 Address-Based Authentication

The simplest authentication mechanism that is supported by most Web servers is address-based authentication. Access control is based on an Internet Protocol (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 susceptible to several types of attacks, including IP spoofing and Domain Name Service (DNS) poisoning. This type of authentication should be use only where minimal security is required, unless it is used in conjunction with stronger authentication methods.

7.3 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, if 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. These limitations can be overcome using basic authentication in conjunction with SSL/TLS (see Section 7.5). Basic authentication is supported by standard-compliant Web browsers [CERT01]. Basic authentication is useful for protecting information from malicious bots (see Section 5.2.4).

7.4 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. Thus, digest authentication is more secure than basic authentication. Unfortunately, all other data is sent in the clear (i.e., unencrypted), and this is vulnerable to interception and alteration. These limitations can be overcome using digest authentication in conjunction with SSL/TLS (see Section 7.5). Like basic authentication, digest authentication is useful for protecting information from malicious bots (see Section 5.2.4).

7.5 SSL/TLS

The SSL and Transport Layer Security (TLS) protocols provide server and client authentication and encryption of communications.[24] SSL was first introduced by Netscape Communications in 1994 and was revised twice (SSL version 3 is the latest version)[25]. 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

---

[24] 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*. For more information on transport layer security, see NIST Special Publication 800-52, *Guidelines for the Selection and Use of Transport Layer Security Implementations*. Both of these documents can be found at http://csrc.nist.gov/publications/nistpubs/index.html.

[25] SSL Version 2.0 is insecure and should not be used.

version 1.0 is formally specified in the IETF RFC 2246,[26] 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.

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.

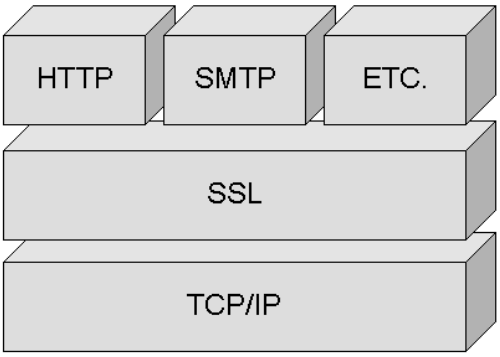


Figure 7.1: SSL/TLS Location within the Internet Protocol Stack

7.5.1 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 name and public key are contained in a valid certificate 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 using the same techniques as those used for server authentication by reversing the roles. SSL/TLS-enabled Web server software can confirm that a client's certificate is valid and was 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.

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

- **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. With an appropriate encryption algorithm SSL/TLS provides a high degree of confidentiality. In addition, all data sent over an encrypted SSL/TLS connection is protected with a mechanism for detecting tampering – that is, for automatically determining whether the data has been altered in transit.

7.5.2 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 7.5.3), substitutes the attacker's keys making it appear to the Web client that the attacker is the Web server and to the Web server that the attacker 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 Web users to be educated in the dangers of this type of attack and that they be taught to confirm a certificate before relying on the security of an SSL/TLS session.[27] This threat can be mitigated if clients rely upon server certificates issued by trusted CAs, or self-signed certificates obtained by secure mechanisms. Presentation of a self-signed certificate may be an indication that a man-in-the-middle attack is underway. Recent browsers perform some checks automatically but cannot be relied upon in all instances.

7.5.3 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, while public-key encryption is better suited to provide authentication and establish symmetric keys. 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

[27] To check a certificate when using Microsoft Explorer, click on the yellow padlock icon in the lower right-hand corner (this icon will only appear when accessing a SSL/TLS protected resource). To check a certificate when using Netscape, click on the padlock icon in the lower right-hand corner (this icon will only appear when accessing a SSL/TLS protected resource).

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."

7.5.4 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. Table 7.1 provides a list of Federal cipher suites, their recommended usage, and their relative strength [SSL98 and Cho02].

Table 7.1: SSL/TLS Encryption Schemes with Diffie-Hellman or RSA Key Exchange

Recommended Use	Cipher Suites
Highest Security:	Encryption: Advanced Encryption Standard (AES) 256-bit encryption Authentication & Digest: Digital Signature Standard (DSS) or RSA with 2048 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: AES 128-bit encryption with fallback to Triple Data Encryption Standard (3DES) 168/112-bit encryption (note: 3DES is considerably slower than AES) Authentication & Digest: DSS or RSA with 1024-bit keys, and SHA-1
Authentication and Tamper Detection	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.[28] Furthermore, a number of countries still maintain restrictions on the export, import, and/or use

[28] AES 128 is the exception to this rule, since it provides higher performance and security than Triple-DES.

of encryption. Patents and licensing issues may affect which commercial encryption schemes can be used. 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 that are in place and that may reduce the need for stronger encryption. For example, using protected methods of communications such as dedicated circuits as opposed to the public Internet.
- Required performance (higher performance requirements may require procurement of additional system resources such as a hardware cryptographic accelerator or 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.

7.5.5 Implementing SSL/TLS

A digital signature is needed to implement SSL/TLS on a Web server. 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." Organizational requirements determine which approach is used.

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, 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 7.2: Sample CSR

Web servers that are SSL/TLS enabled provide specific instructions for the generation of a CSR.[29] 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].

Spelling and punctuation should be checked when information is provided during the CSR generation process. 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 7.3 shows an X.509

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

certificate encoded in PEM format. 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.


```
-----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 7.3: 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 is lost and cannot be recovered from backup media, a new certificate must be created.

- Create and store a backup copy of the encryption keys on read-only media in case the keys are deleted accidentally. If the keys lost and cannot be recovered from backup media, a new key pair and certificate must be created. Note that the backup copy of the keys must be physically secured and should be encrypted as well.

- 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 8.2.2) 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 some CAs do not issue

revocation information and many client implementations do not obtain or process revocation information.

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 is signed by the web server itself rather than a third-party CA. Three limitations of 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. With a self-signed certificate, the web server is "guaranteeing" its own identity. 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.

- The security services provided using this certificate depend entirely upon the security of the mechanism used for distribution. Where organizations install the certificate as part of browser configuration, acceptable levels of security may be achieved. Public users accepting the certificate upon access to the site are provided little real security.

Although the sequence of steps is not identical for all Web servers, the implementation of a self-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:

- Turn off SSL 2.0

- Configure SSL/TLS to restrict cryptographic algorithms to the selected cipher suite(s) (see Section 7.5.4).

- 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 8.2).

- 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 RFCs 2246, 2626, and 2817.[30]

7.5.6 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:

- SSLava is an implementation of SSL/TLS written in Java from Phaos Technology (http://www.phaos.com/index.html).

- OpenSSL is an open source implementation of SSL/TLS for Unix and Linux platforms (http://www.openssl.org).

7.6 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 5.2.4), 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

[30] http://www.ietf.org/rfc/rfc2246.txt, http://www.ietf.org/rfc/rfc2626.txt, and http://www.ietf.org/rfc/rfc2817txt

Completed	Action
	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
☐	Configure file integrity checker to monitor Web server certificate
☐	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)

8. 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.

8.1 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 form the internal network and 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.

8.1.1 Unadvisable Network Locations

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 attackers. If they manage to compromise the Web server, they will be on the internal network and can more easily compromise internal hosts.

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.

8.1.2 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. There are a wide variety of DMZ configurations, each with its own strengths and weaknesses.

In creating a DMZ, an organization will place a firewall between its border router and its internal network (in some configurations the border router itself may act as a basic firewall). The new segment of network that is created by this action is where a Web server is placed along with other network infrastructure components and servers that need to be externally accessible. Figure 8.1 illustrates an example of a simple DMZ using a router with access controls lists (ACLs) to restrict certain types of network traffic to and from the DMZ.

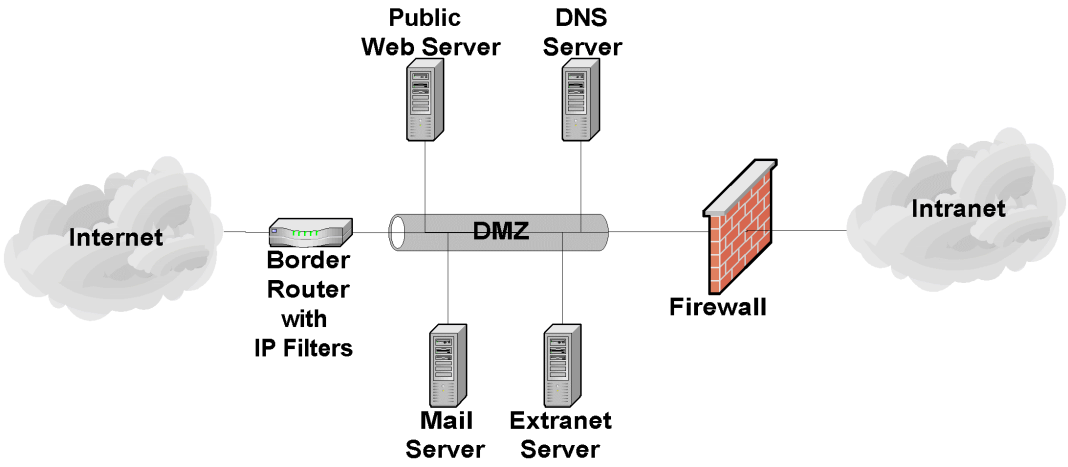


Figure 8.1: Basic DMZ

This type of DMZ is a lower cost approach. That is generally only appropriate for small organizations that face a minimal threat. The basic weakness in the approach is that while the router is able to protect against most network attacks it is not "aware" of the HTTP and thus cannot protect against application layer attacks aimed at the Web server. A superior approach is to add a second firewall in between the Internet and the DMZ. This offers better protection to the DMZ. An example of this type of implementation is shown in Figure 8.2.

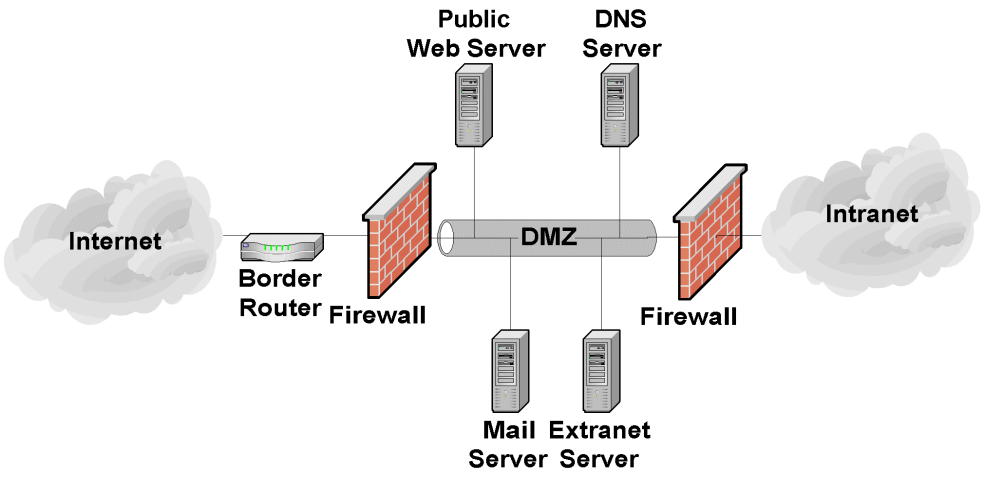


Figure 8.2: Two Firewall DMZ

This two firewall DMZ offers superior protection over a router based DMZ since the dedicated firewalls can have a more complex and powerful security rule set. In addition, the dedicated firewall is often able to analyze incoming and outgoing HTTP traffic, it can detect and protect against application layer attacks aimed at the Web server. Depending on the configuration of the firewalls and the level of traffic the DMZ receives; this type of DMZ may result in some performance issues.

For organizations who desire the security of the two firewall DMZ but who do not have the resources to purchase two firewalls, there exists another option called the "service leg" DMZ. In this configuration, a firewall is constructed with three (or more) 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 8.3).

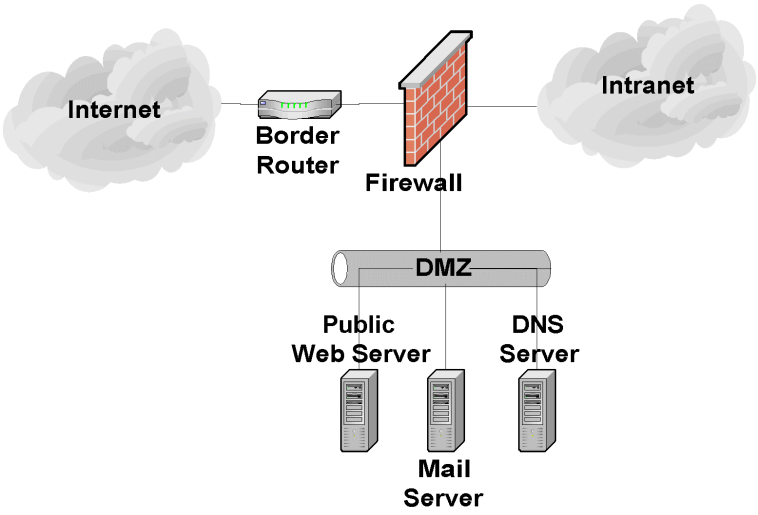


Figure 8.3: Three Interface Firewall DMZ

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 [NIST02a].

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

- Web server may be better protected and network traffic to and from the Web server can be monitored.
- Compromise of the Web server does not directly threaten the internal production network.
- Greater control can be provided over the security of the Web server since traffic two and from the Web server can be controlled.
- 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. However it should only be considered secure when employed in conjunction with the other steps discussed in this document.

8.1.3 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 8.4).

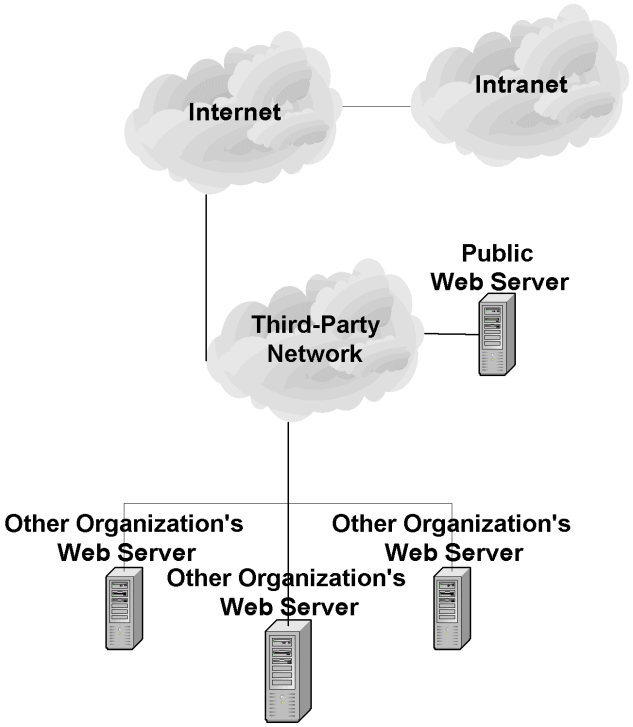


Figure 8.4: Outsourced Web Server Hosting

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 the internal production network.
- Outsourcer may have greater knowledge in securing and protecting Web servers.
- The network can be optimized solely for the support and protection of Web servers.

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

- Requires trusting a third-party with Web server content.
- It is difficult to remotely administer the Web server or remotely update Web server content.
- Less control can be provided 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. It usually does not make sense for organizations that wish to maintain tight control over the Web server.

8.2 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.

8.2.1 Router/Firewall Configuration

Firewalls (or routers acting as 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 routers, which can provide access control on IP packets; 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.[31]

A common misperception is that firewalls (or routers acting as firewalls) 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 to software vulnerabilities. Web servers in particular are vulnerable to many attacks, even when located behind a secure, well-configured firewall. For example, a firewall that is protecting a Web server should 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.

A router is able to function as network layer (also called packet filter) firewall. A router functioning as a network layer firewall can provide filtering based on several pieces of information [NIST02a]:

- Source IP address
- Destination IP address
- Traffic type

[31] 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).

- TCP/UDP port number (sometimes).

The strengths of network layer firewalls are as follows:

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

The weaknesses of network layer firewalls are as follows:

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

In addition to packet filtering routers, the only pure network layer firewalls that are available today are small office home office (SOHO) firewall appliances and personal firewalls [NIST02a] that may only perform packet-level filtering.

Stateful inspection firewalls are transport layer devices that incorporate "awareness" of the state of a TCP connection. Stateful inspection firewalls maintain internal information such as the state of connections passing through them, and the contents of some of the data streams. This allows for better and more accurate rules sets and filtering. Stateful inspection firewalls add the capability to enforce rules based upon connection state to the capabilities of a filtering router.

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:

- Provide logging capabilities
- Provide 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 be secured, in addition to the firewall software itself. Typically, routers and stateful inspection firewalls run on specialized operating systems, thus reducing this risk.

To more 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 all inbound traffic with an internal IP address (to prevent IP spoofing attacks)
- Block client connections from the Web server to the Internet and the organization's internal network (this will reduce the impact of certain worms such as Code Red)
- Block (in conjunction with the intrusion detection system [see Section 8.2.2]) IP addresses or subnets that the IDS reports are attacking the organizational network
- Notify the network administrator or appropriate security personnel of suspicious activity through an appropriate means (e.g., page, e-mail and network trap)
- Provide content filtering
- Protect against denial of service attacks
- Detect malformed or known 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.

- Be 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 9.1.

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.

8.2.2 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 and/or appropriate security personnel when it identifies a possible intrusion or penetration attempt.[32]

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 7.5]) 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-

[32] For more information about IDSs see NIST Special Publication 800-31, *Intrusion Detection Systems* (http://csrc.nist.gov/publications/nistpubs/index.html).

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 of 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 and are not immediately compromised by a successful attack on the Web server.

Network-based IDSs 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 the attack, it may not be detected by the IDS. In addition, the attacker can format the method of the attack (e.g., fragment packets, alter attack pattern 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 8.2.3), 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 weakness is no IDS can detect all, or, often, 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.

The following applications have some IDS capabilities and are a useful complement to an IDS although they are not considered to be IDSs.

- **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 may divert an attacker's attention from the "real" information system resources and allow an organization to monitor the attacker's actions without risking "real" organizational information and resources. The real benefit of a honey pot is that since, by definition, it is not used for anything other than detecting attackers, there is a high probability that any network traffic or log entries are an indicator of malicious activity. The danger of a honeypot is that if it is

compromised, it may be used to attack other networks potentially making the honeypot host organization liable for damages.[33]

- **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 5.3). 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-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

---

[33] Honey pots and related technologies should be used conservatively and only by organizations with a highly skilled technical staff that are willing to experiment with leading-edge technology. Furthermore, such techniques should be used only after seeking guidance from legal counsel given the possible liability issues. For more information see NIST Special Publication 800-31 *Intrusion Detection Systems* (http://csrc.nist.gov/publications/).

- Network protocol used by attack.

- Be updated with new attack signatures frequently (at least on a weekly basis).

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

8.2.3 Network Switches and Hubs

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. Switches isolate the communications of hosts on a network segment from each other. This isolation may be beneficial in reducing the impact of a DoS attack on other hosts on the network.

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. E-mail servers, 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 is 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 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 Address Resolution Protocol (ARP) spoofing and ARP poisoning attacks.[34] A switch that has these security capabilities, should be enabled (see appropriate vendor documentation).

[34] ARP poisoning occurs when an attacker successful updates the ARP cache on a target host with a forged ARP entry. This is generally used to redirect network traffic for malicious purposes.

8.3 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 using SSL/TLS)
☐	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 to 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
☐	IDS updated with new attack signatures frequently (weekly basis)

Completed	Action
☐	IDS configured to monitor the system resources available on the Web server (host-based)
	Network switches and hubs
☐	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)

9. Administering a Web Server

9.1 Logging

Logging is the principal component of secure administration of a Web server. Logging the appropriate data and then monitoring and analyzing those logs are critical activities. Review of Web server logs is effective, particularly for encrypted traffic, where network monitoring is far less effective. Review of logs is a mundane activity that many Web administrators have a difficult time fitting 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 and assess intrusion attempts. Similar problems can result if necessary procedures and tools are not 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.

9.1.1 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 Web 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 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[35]
  - Authenticated user in accordance with the basic authentication scheme (see Section 7.3)
  - Date
  - URL requested
  - Status of the request
  - Number of bytes actually transferred.
- **Combined Log Format** – This 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** – This 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 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).

---

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

9.1.2 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 special 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 host operating system. Log information associated with programs, scripts, and plug-ins can add significantly to the typical information logged by the Web server.

9.1.3 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.
- Ensure procedures or mechanisms are in place so that log files do not fill up the hard drive.

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 in avoiding inadvertent alteration of log files as a result of errors in file access administration. Web administrators should determine the circumstances under which they may want to enable such checks (assuming the Web server software supports this feature).

9.1.4 Reviewing and Retaining Log Files

Reviewing log files can be time-consuming and laborious. 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 (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 the 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 9.1.5).

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 attackers who compromise a Web server cannot alter the log files to cover their 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. Although storing the log files on a separate server is imperative for security, a local copy (on the originating server) should also be maintained to protect against the possibility of data being lost in transmission.

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.

9.1.5 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.

9.2 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 content.

9.2.1 Web Server Backup Policies and Strategies

The Web administrator should 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 should 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 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
- Definition of key terms, especially legal and technical
- Description of the requirements in detail from the legal, business, and organization's perspective
- Frequency of backups
- Procedures for ensuring data is properly retained and protected
  - Encrypted transmission
  - Encrypted storage
  - Physical protection mechanisms
- Procedures for ensuring data is properly destroyed or archived when no longer required
- Procedures for responding to discovery requests
- Responsibilities of those involved in data retention, protection and destruction activities
- A table showing the information type and it corresponding retention period
- Specific duties of a central/organizational data backup team if one exists.

Two primary types of backups exist. A full backup includes 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.

9.2.2 Maintain a Test Web Server

Most organizations will probably wish to maintain a test or development Web server. Ideally this server should have identical hardware and software as the production or live Web server and be located on an internal network segment (intranet) where it can be fully protected by the organization's perimeter network defenses. Although the cost of maintaining an additional Web server is not inconsequential, having a test Web server offers numerous advantages:

- Provides a platform to test new patches and service packs prior to application on the production Web server
- Provides a development platform for the Web master and Web administrator to develop and test new content and applications
- Software that is critical for development and testing that might represent an unacceptable security risk on the production server can be installed on the development server (e.g., software compliers, administrative tool kits, remote access software)

Note: The test Web server should be separate from the server that maintains an authoritative copy of the content on the production Web server (see Section 9.2.3).

9.2.3 Maintain an Authoritative Copy of Organizational Web Content

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 9.2.1). 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 8.1.2). 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 as 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 a production Web server
  - Data can be transferred using a secure physical media (e.g., encrypted and/or write once media such as a CD-R)
  - Use a secure protocol (e.g., SSH) for network transfers.
- Include the procedures restoring from the authoritative copy in the organizational incident response procedures (see Section 9.3)
- Consider automatic updates from authoritative copy to Web server periodically (quarter hourly, hourly, daily, etc.) because this will overwrite a Web site defacement automatically.

9.3 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[36]. 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.

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

- Report incident to organization's computer incident response capability
- Consult the organization's security policy
- Isolate compromised system(s) or take steps to contain attack so additional evidences can be collected[37]
- Investigate other "similar"[38] hosts to determine if the attacker also has compromised other systems
- Consult, as appropriate, with management, legal counsel, and law enforcement expeditiously
- 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
    - Data from system logs, intrusion detection and firewall log files.
- Restore the system
    - Two options exist:
        - Install clean version of operating system, applications, necessary patches and Web content
        - Restore from backups (this option can be more risky, as the backups may have been made after the compromise and restoring from a comprised back may still allow the attacker access to the system).

---

[36] 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).

[37] Isolating the system must be accomplished with great care if the organization wishes to collect evidence. Many attackers now configure compromised systems to erase evidence if a compromised system is disconnected from the network or rebooted. One method to isolate a system would be to reconfigure the nearest upstream switch or router.

[38] "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.

- 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 attacker 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 attacker's actions, the less risk there is in restoring from a backup and patching the vulnerability. If less is known about the intruder's actions, it is essential 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).

9.4 Security Testing Web Servers

Periodic security testing of public Web servers is critical. Without periodic testing, there is no assurance that the current protective measures are working or that the security patch just

applied by the Web administrator is functioning as advertised. Although a variety of security testing techniques exists, vulnerability scanning is the most common. Vulnerability scanning assists a Web administrator in identifying vulnerabilities and verifying whether the existing security measures are effective. Penetration testing is also used, less frequently and usually only as part of an overall penetration test of the organization's network.[39]

9.4.1 Vulnerability Scanning

Vulnerability scanners are automated tools that are 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, missing patches or system upgrades, and validate compliance with, or deviations from, the organization's security policy. To accomplish this effort, vulnerability scanners identify operating systems and major software applications running on hosts and match them with known vulnerabilities. Vulnerability scanners employ large databases of vulnerabilities to identify vulnerabilities associated with commonly used operating systems and applications.

However, vulnerability scanners have some significant weaknesses. Generally, they identify only surface vulnerabilities and are unable to address the overall risk level of a scanned Web server. 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. Furthermore, vulnerability scanners cannot 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 rather than more esoteric ones because it is impossible for any one product to incorporate all known vulnerabilities in a timely manner. Also manufacturers want to keep the speed of their scanners high (more vulnerabilities detected requires 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

---

[39] 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).

- Identifying applications and banner grabbing
- Identifying operating systems
- Identifying vulnerabilities associated with discovered operating systems and applications
- Testing host compliance with organizational security policies.

Organizations should conduct vulnerability scanning to validate that operating systems and Web server applications are up to date on security patches and software versions. Vulnerability scanning is a labor-intensive activity that requires a high degree of human involvement to interpret 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 on a weekly to monthly basis. Many organizations also run a vulnerability scan whenever a new vulnerability database is released for the organization's scanner application.

Organizations should also considering running more than one vulnerability scanner. As previously discussed, no scanner is able to detect all know vulnerabilities, however using two scanners will generally increase the number of vulnerabilities detected. A common practice is to use one commercial and one freeware scanner.

Vulnerability scanning results should be documented and discovered deficiencies corrected. The following corrective actions may be necessary as a result of vulnerability scanning: 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.

9.4.2 Penetration Testing

"Penetration testing is security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation" [NISS99]. The purpose of penetration testing is to identify methods of gaining access to a system by using common tools and techniques developed by hackers. This testing is highly recommended for complex or critical systems.

Penetration testing can be an invaluable technique to any organization's information security program. However, it is a very labor-intensive activity and requires great expertise to minimize the risk to targeted systems. At a minimum, it may slow the organization's networks response time resulting from network mapping and vulnerability scanning. Furthermore, the possibility exists that systems may be damaged in the course of penetration testing and may be rendered inoperable. Although this risk is mitigated by the use of experienced penetration testers, it can never be fully eliminated.

Penetration testing does offer the following benefits [NIST02b]:

- Tests the network using the same methodologies and tools employed by hackers
- Verifies whether vulnerabilities exist

- Goes beyond surface vulnerabilities and demonstrates how these vulnerabilities can be exploited iteratively to gain greater access
- Demonstrates that vulnerabilities are not purely theoretical
- Provides the "realism" necessary to address security issues
- Allows for testing of procedures and the susceptibility of the human element to social engineering.

9.5 Remotely Administering a Web Server

Web server administrators should carefully consider 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 8.1). For example, if the Web server is located external to the organization's firewall or IP filtering router, then no remote administration or updating should be implemented. Remote administration or content updating can be implemented relatively securely from the internal network when a Web server is located behind a firewall. Remote administration or content updating should not be allowed from a 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, the following steps should ensure secure implementation:

- 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 for the remote administration utility or application.
- Do not mount any file shares on the internal network from the Web server or vice versa.

9.6 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
☐	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 log file 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
☐	Report incident to organization's computer incident response capability
☐	Consult the organization's security policy
☐	Isolate compromised system(s) 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, as appropriate, with management, legal counsel, and law enforcement officials expeditiously
☐	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

Completed	Action
☐	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)
☐	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

DISCLAIMER

This Appendix is only a guide on recommended product security settings and is meant to complement not replace well-structured organizational policy or sound judgment. Moreover, its contents do not address site-specific issues, which typically must be taken into account when applying this material to actual system environments involving local operational and policy concerns. This appendix is intended to serve as an example of how the generic guidelines given in the body of the main document can be interpreted and applied to specific classes of products. NIST recommends that experienced technical personnel, familiar with the organizational operational and policy aims, review these settings, experiment with them on a suitably configured test system, and judge the suitability of their effect on an operational system before attempting deployment.

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. The section covers Apache on Linux/Unix and does not address the Windows version.

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 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 TCP wrappers, ipchains, or iptables, provided they are provided by the host operating system, 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

The Apache server software is available in two forms, source code and precompiled/prepackaged binaries. For expert administrators, the best and most secure option for installing the Apache is to download the source code and compile the application. This allows expert administrators to include only the functionality their organization requires.

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 6.2).

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 6.2). 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 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. Disable automatic directory listings with the "-Indexes" options directive. To prevent the use of symbolic links that could allow access to files or directories that are outside the Web server's DocumentRoot, with options directive "-FollowSymLinks."

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 6.2.

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 A.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 hosts 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 attacker. 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 hostname 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 hostnames, 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.[40]

---

[40] 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

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
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 replacement 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 7.5, 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

---

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.

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, and replacement for 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 those provided in Appendix E, 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.[41]

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 7.

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.cronolog.org. 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.

---

[41] For more information about applying security patches, see NIST Special Publication 800-40, *Procedures for Handling Security Patches* (http://csrc.nist.gov/publications/nistpubs/index.html).

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.[42]

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.[43]

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

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

[43] 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
☐	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
☐	Remove all unnecessary scripts and executables (e.g., CGI, PHP, etc.)
☐	Delete all unnecessary files from the HTML document tree
☐	Prevent users from creating .htaccess 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

DISCLAIMER

This Appendix is only a guide on recommended product security settings and is meant to complement not replace well-structured organizational policy or sound judgment. Moreover, its contents do not address site-specific issues, which typically must be taken into account when applying this material to actual system environments involving local operational and policy concerns. This appendix is intended to serve as an example of how the generic guidelines given in the body of the main document can be interpreted and applied to specific classes of products. NIST recommends that experienced technical personnel, familiar with the organizational operational and policy aims, review these settings, experiment with them on a suitably configured test system, and judge the suitability of their effect on an operational system before attempting deployment.

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 4.3). 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.[44]

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

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.

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,[45] 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

[45] 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.

completely clear.[46] 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:[47]

- 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).

The minimum installation for IIS 4.0 and the NTOP 4.0 includes the following components:[48]

- 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

---

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

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

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

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.

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 8).

In addition, IIS should be installed on a separate physical or logical (partition) hard drive from that of the host operating system. This provides additional protection against certain types of attacks. It also provides some performance benefits as well.

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 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 NIST Special Publication 800-40, *Procedures for Handling Security Patches*.[49]

Microsoft Windows administrators should consult several sources to obtain updates for their operating system. Service packs contain orderable or downloadable updates for Microsoft operating systems 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/

[49] http://csrc.nist.gov/publications/nistpubs/index.html.

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 be ineffective or cause instability.

B.4.3 Apply Security Templates

Windows NT provided the Security Configuration Editor (secedit) with Service Pack 4.0 (and later) and Windows 2000 provides secedit 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.4.4 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.4.5 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
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.4.6 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.4.7 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.4.8 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) and 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.4.9 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 6.2.

B.4.10 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 mediate the access users have to particular resources on the hard disk.[50]

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 5.2 for more information).

B.4.11 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.

---

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

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 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 7.5). 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

Field Name
URI Stem
HTTP Status
Win32 Error
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 7.

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).

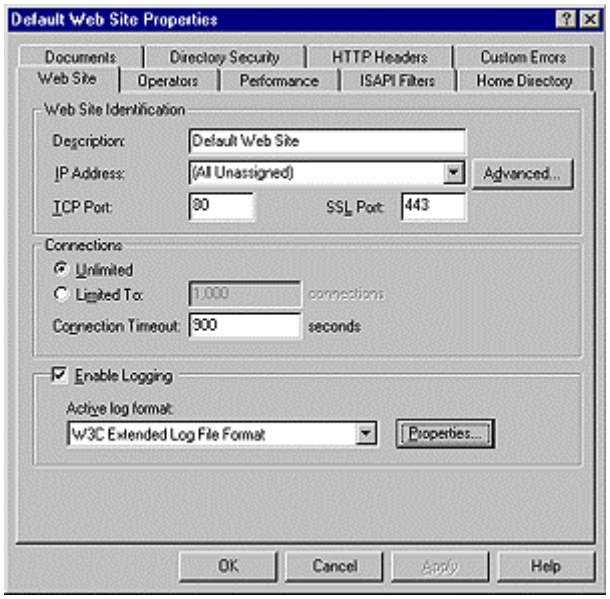


Figure B.1: Default Web Site Properties

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.

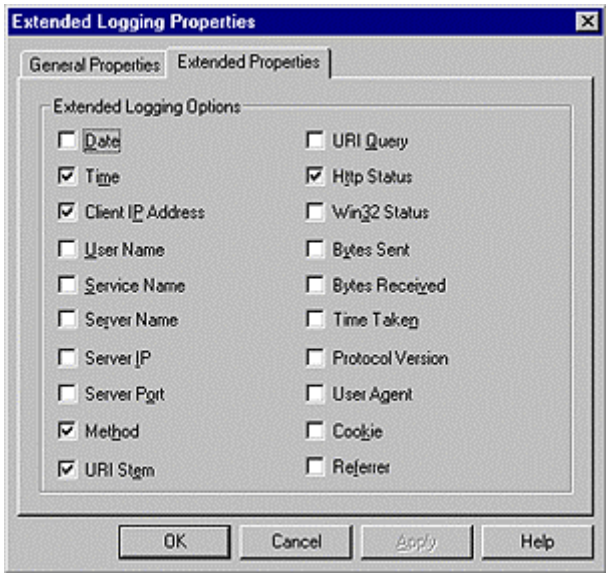


Figure B.2: Default Web Site Properties

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	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

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

B.5.6 Consider Moving or Removing the Command Shell

Many of the most commonly exploited IIS vulnerabilities allow a Uniform Resource Locator (URL) to be constructed that ultimately causes cmd.exe to be invoked on the server. The attacker then appends commands, such as ECHO, to the command-line parameter for cmd.exe. This results in the ability to execute commands on the targeted IIS server. On Windows NT, the best way to mitigate this risk is to remove cmd.exe or if its functionality is required move it to a different location. Windows 2000 with its Windows File Protection (WFP) mechanism will automatically replace cmd.exe if it is move, rename or removed. The best way to protect cmd.exe under Windows 2000 is to assign explicit permissions to member of the Administrators group and deny all access to member of the SYSTEM and IUSR/IWAM accounts [Coop01].

B.5.7 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.[51]

B.5.8 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, .idw, .htw
Internet Database Connector	.idc
Internet printing	.printer
Server-side includes	.shtm, .stm, .shtml

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

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.5.9 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.[52]

B.5.10 Disable Internet Printing Protocol Support

Cited by SANS as one of the five most widely exploited holes in unpatched versions of IIS in 2001,[53] 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.[54]

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.5.11 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

---

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

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

[54] 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.

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.5.12 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:

- Anonymous. This authentication method was discussed in Section 4.1.3. 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 7.3 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.5.13 Set Appropriate Directory Permissions

Segregating files of a similar type or function into their own directories allows for more granular control over a Web site or Web application. Tables B.6 and B.7 list recommended directory permission for IIS 4.0 and 5.0 respectively.

Table B.6: Recommended Directory Permissions for IIS 4.0

Type of Data	Example Directories	Data Examples	NTFS File Permissions	IIS4.0 Permissions
Static Content	\wwwroot\images \wwwroot\home \ftproot\ftpfiles	HTML, images, FTP downloads, etc.	Administrators (Full Control) System (Full Control) WebAdmins (Read,Write,Delete) Authenticated Users (Read) Anonymous (Read)	Read
FTP Uploads (if required)	/ftproot/dropbox	Directory used as a place for users to store documents or review prior to the Admin making them available to everyone	Administrators (Full Control) WebAdmins or FTPAdmins (Read,Write,Delete) Specified Users (Write)	Write
Script Files	\wwwroot\scripts	.ASP	Administrators (Full Control) System (Full Control) WebAdmins (Read,Write,Execute,Delete) Anonymous (Execute)	Script
Other Executable and Include Files	\wwwroot\executables \wwwroot\include	.exe, .dll, .cmd, .pl .inc, .shtml, .shtm	Administrators (Full Control) System (Full Control) WebAdmins (Read,Write,Execute,Delete) Authenticated Users (Read) Anonymous (Execute)	Execute
Metabase	\WINNT\system32\inetsrv	MetaBase.bin	Administrators (Full Control) System (Full Control)	N/A

Table B.7: Recommended Directory Permissions for IIS 5.0

Type of Data	Example Directories	Data Examples	NTFS File Permissions	IIS5.0 Permissions
Static Content	\wwwroot\images \wwwroot\home \ftproot\ftpfiles	HTML, Images, FTP downloads, etc.	Administrators (Full Control) System (Full Control) WebAdmins (Read & Execute, Write, Modify) Authenticated Users (Read) Anonymous (Read)	Read
FTP Uploads (if required)	/ftproot/dropbox	Directory used as a place for users to store documents for review prior to the Admin making them available to everyone	Administrators (Full Control) WebAdmins or FTPAdmins (Read & Execute, Write, Modify) Specified Users (Write)	Write
Script Files	\wwwroot\scripts	.ASP	Administrators (Full Control) System (Full Control) WebAdmins (Read & Execute, Write, Modify) Authenticated Users: special access (Execute) Anonymous: special access (Execute)	Scripts only
Other Executable and Include Files	\wwwroot\executables \wwwroot\include	.exe, .dll, .cmd, .pl .inc, .shtml, .shtm	Administrators (Full Control) System (Full Control) WebAdmins (Read & Execute, Write, Modify) Authenticated Users: special access (Execute) Anonymous: special access (Execute)	Scripts only or Scripts and Executable (depending on necessity)
Metabase	\WINNT\system32\ inetsrv	MetaBase.bin	Administrators (Full Control) System (Full Control)	N/A

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.[55]

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:

---

[55] 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).

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

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 _): [56]

```
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

[56] 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.[57]

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 7. 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.

---

[57] The importance of patching for software is elaborated in NIST Special Publication 800-40, *Procedures for Handling 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 addresses the Windows operating system, it does not address the Microsoft IIS Web server application. The Microsoft Base Line Security Analyzer corrects that oversight. It runs on Windows 2000 and Windows XP systems and uses Microsoft Network Security Hotfix Checker (HFNetChk) to scan for vulnerabilities as well as missing hotfixes and service packs in Windows NT 4.0, Windows 2000, Windows XP, Internet Information Server (IIS) 4.0 and 5.0, SQL Server 7.0 and 2000, Internet Explorer 5.01 and later, and Office 2000 and 2002.

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.

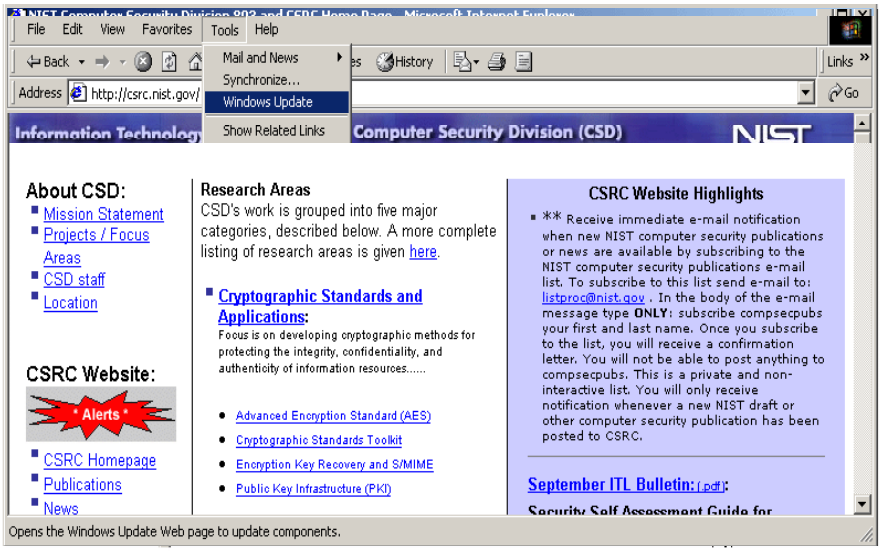


Figure B.3: Accessing Windows Update Through Internet Explorer

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.

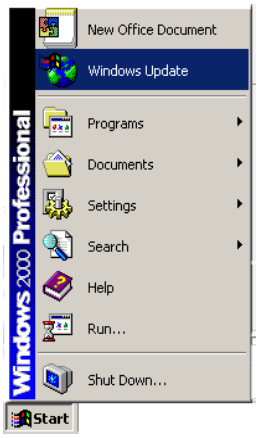


Figure B.4: Accessing Windows Update Though the Start Menu

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.

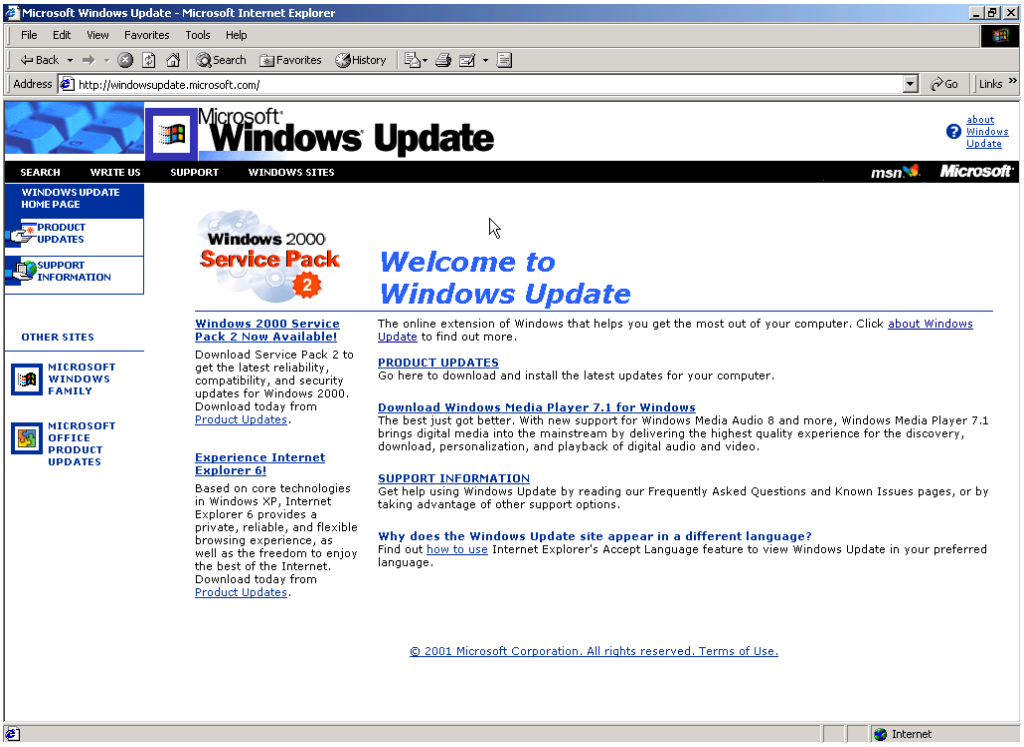


Figure B.5: Windows Update Homepage

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.

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.

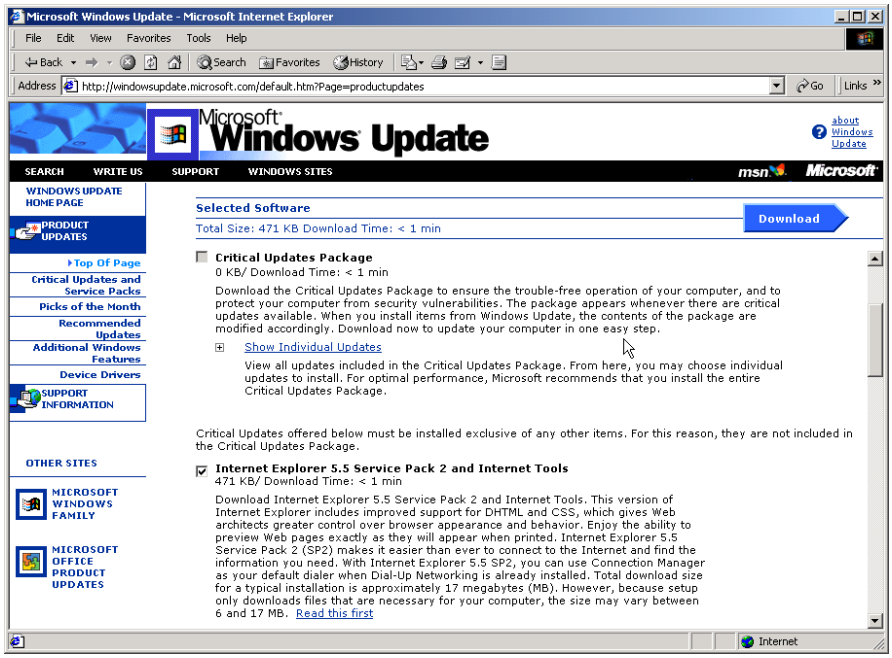


Figure B.7: Windows Update Recommend Updates

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.

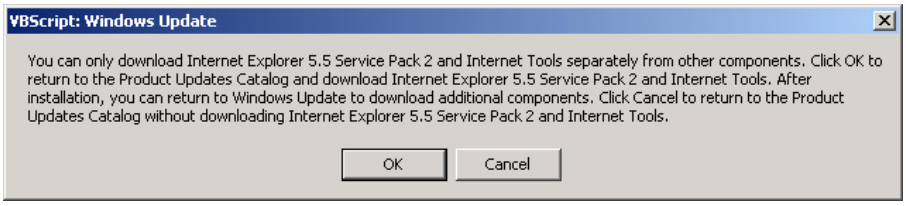


Figure B.8: Windows Update Warning

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.

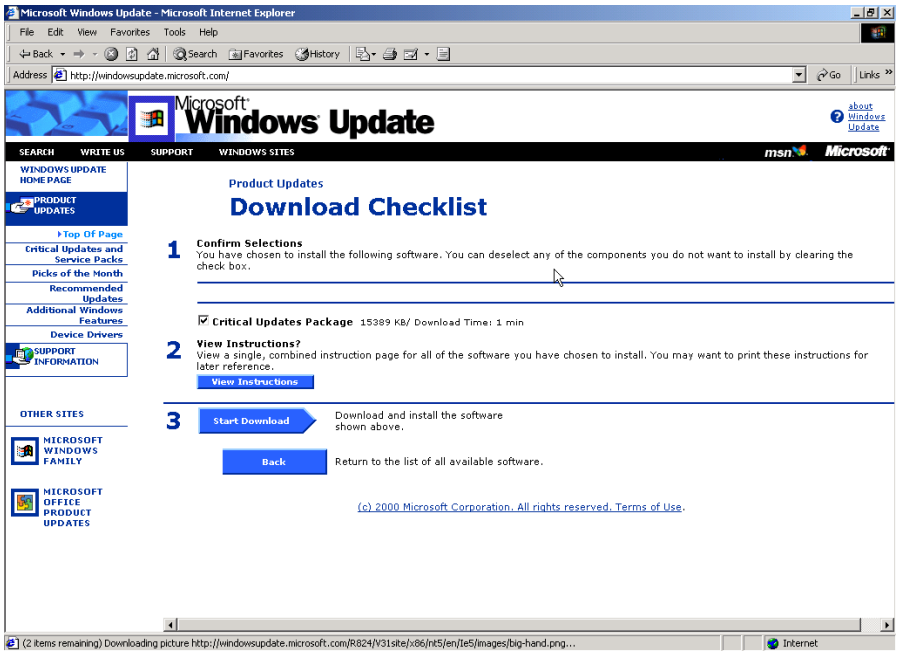


Figure B.9: Windows Update Download Checklist

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).

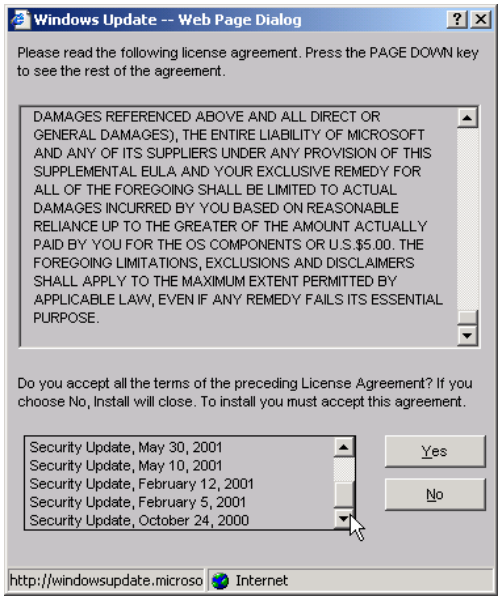


Figure B.10: Windows Update Confirmation and License Agreement

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.

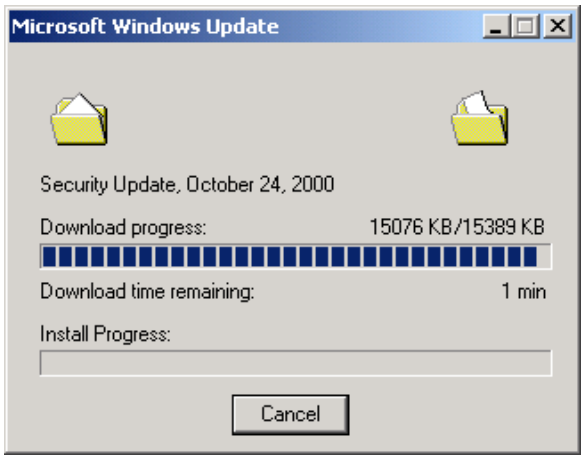


Figure B.11: Windows Update Download Status Window

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).

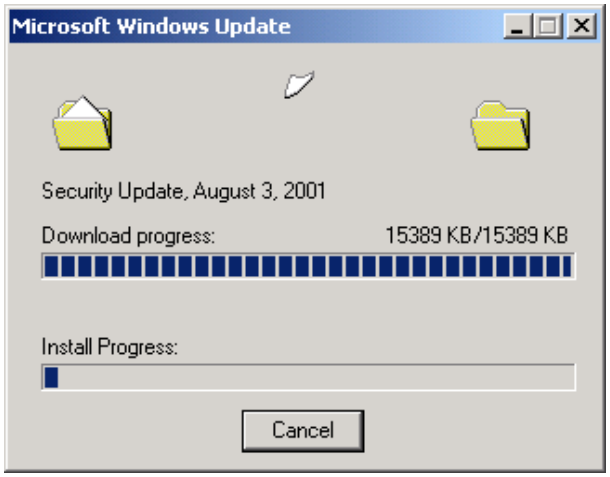


Figure B.12: Windows Update Install Status Window

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

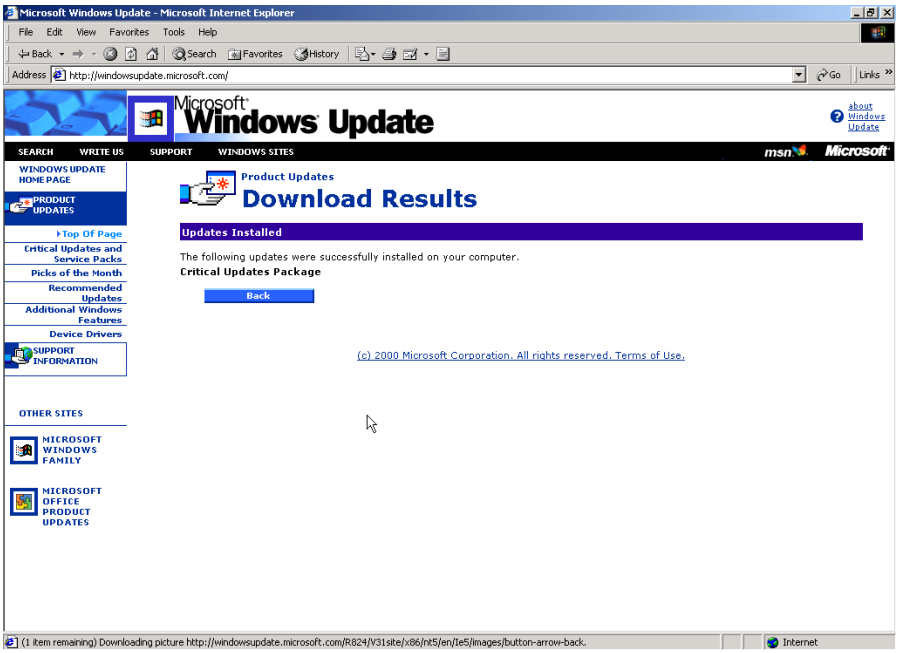


Figure B.13: Windows Update Download Results Window.

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.

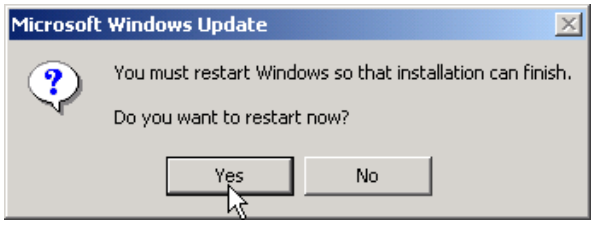


Figure B.14: Windows Update Reboot Dialog Box

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

A.9.2 Using Microsoft Baseline Security Advisor

The Microsoft Baseline Security Analyzer (MBSA) is a tool that identifies common security misconfigurations and missing hotfixes via local or remote scans of Windows systems. MBSA, designed and developed to replace the Microsoft Personal Security Advisor (MPSA), runs on Windows 2000 and Windows XP systems and uses Microsoft Network Security Hotfix Checker (HFNetChk) to scan for vulnerabilities as well as missing hotfixes and service packs in Windows NT 4.0, Windows 2000, Windows XP, Internet Information Server (IIS) 4.0 and 5.0, SQL Server 7.0 and 2000, Internet Explorer 5.01 and later, and Office 2000 and 2002.

MBSA provides users with the ability to scan a single Windows system and obtain a security assessment as well as a list of recommended corrective actions. Furthermore, administrators may use the MBSA tool to scan multiple Windows systems on their network for vulnerabilities to help ensure systems are up-to-date with the latest security-related patches.

MBSA provides the same functionality as HFNetChk in an easy-to-use interface with some additional capabilities, including the ability to examine Windows desktops and servers for common security vulnerabilities and best practices such as:

- Examining Windows desktops and servers for common best practices such as strong password parameters;
- Scanning servers running IIS and SQL server for common security misconfigurations; and
- Checking for misconfigured security zone settings in Microsoft Office, Outlook, and Internet Explorer.

MBSA is available for free download at http://www.microsoft.com/technet/security/tools/Tools/mbsahome.asp.

The Welcome screen appears upon launching the application (see Figure B.15).

Figure B.15: MBSA Welcome Screen

The navigation menu runs vertically along the left side of the MBSA window (see Figure B.16). To navigate within the application, click on the appropriate button in the menu. The upper half of the menu contains options to conduct scans and view security reports of previously scanned computers. The lower half of the menu contains links to helpful resources for additional information and troubleshooting.

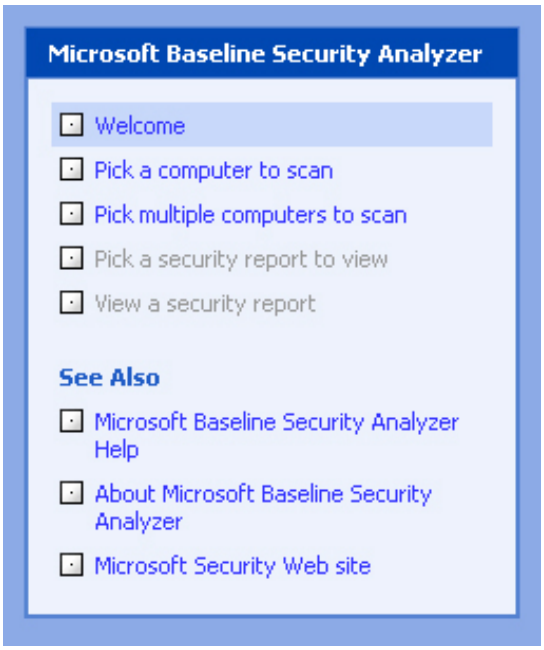


Figure B.16: MBSA Navigation Menu

The Welcome screen includes a brief description of the MBSA utility's purpose and capabilities. The introduction notes that a user must have administrative privileges on each computer to be scanned. When scanning a single system, the account with which a user runs MBSA must either be the Administrator or a member of the local Administrator's group. When scanning multiple systems users must be an administrator of each computer or a domain administrator. If the account with which a user runs MBSA is not an Administrator or a member of the local or domain Administrator's group (for single and multiple scans, respectively), the Unable to scan all computers screen will be appear noting for which computers the scan could not be conducted (see Figure B.17). This screen will appear after a scan has been attempted on the computer name or IP address, and no security report will be produced for the identified computer(s).

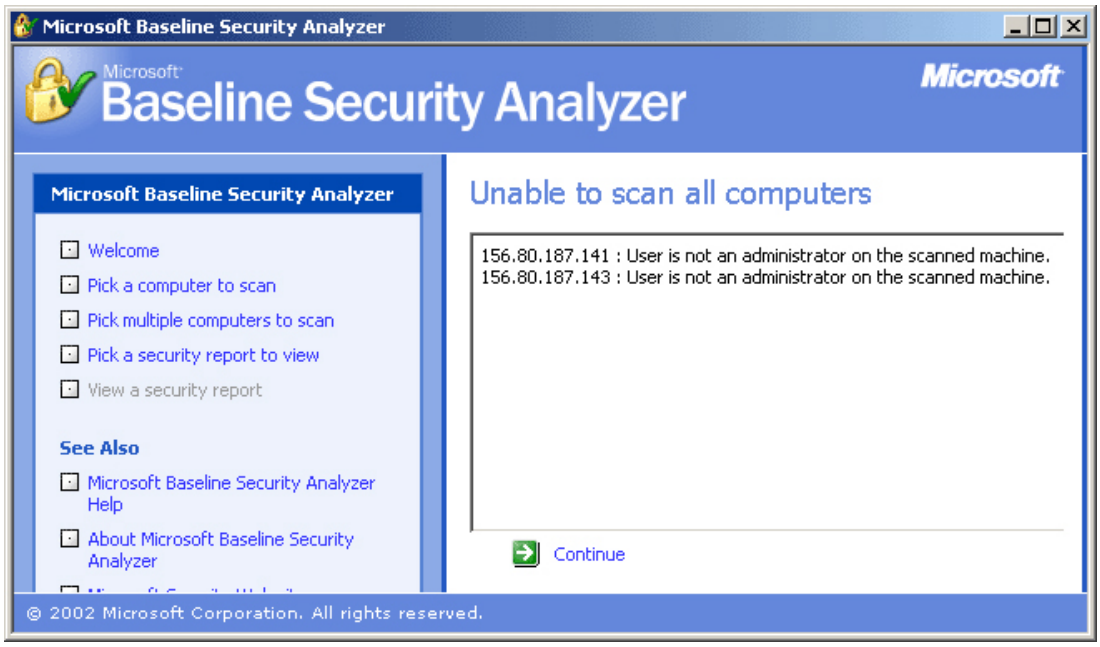


Figure B.17: Unable to Scan All Computers Screen

Three options are located in the Welcome screen (see Figure B.18).

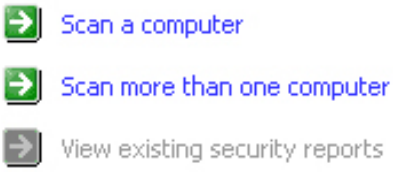


Figure B.18: Welcome Screen Options

These options are identical to those in the navigation menu along the left side of the MBSA window.

To scan a single computer, click on the **Scan a computer** option from the *Welcome* screen or on the **Pick a computer to scan** option from the navigation menu.

The *Pick a computer to scan* screen will appear (see Figure B.19). Here, the computer to be scanned is specified and the scope of the scan is defined.

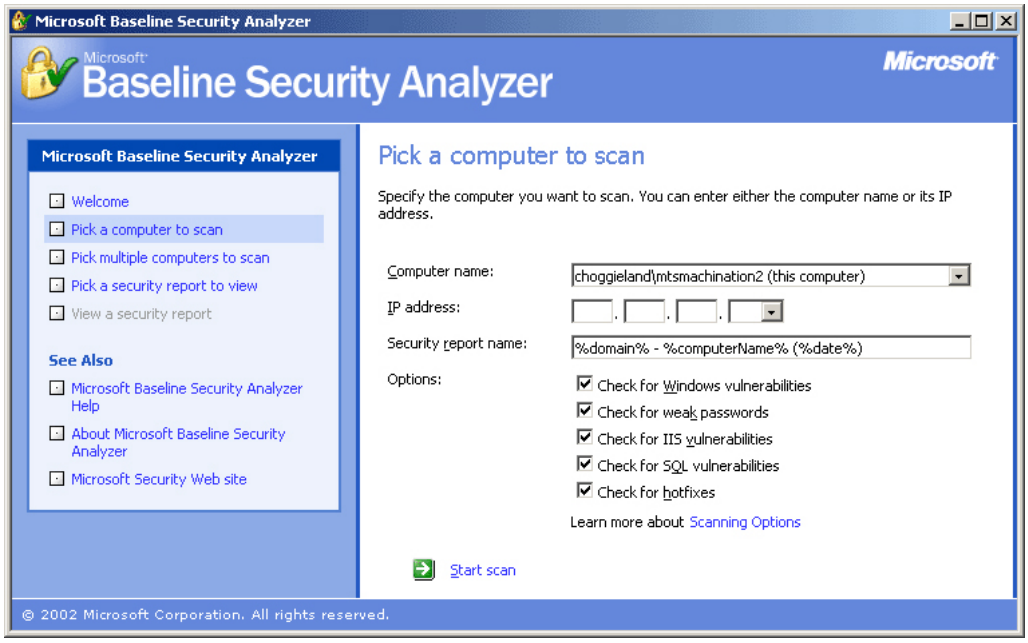


Figure B.19: Pick a Computer to Scan Screen

- **Computer name** – Identifies the computer to be scanned. By default, the field is populated with the name of the local machine running the MBSA utility. Therefore, to conduct a "self-scan" do not alter this field. To scan a computer other than the local machine, enter the appropriate computer name in this field.

- **IP address** – To specify the IP address of the computer to be scanned, instead of a computer name, enter the IP address in this field.

- **Security report name** – By default MBSA labels the security report with the domain name followed by the name of the computer scanned and the date of the scan. To rename the security report, specify the name in this field.

- **Options** – Specifies the scope of the scan. Select or deselect the areas MBSA will check for vulnerabilities as appropriate.

For more information on the benefits and/or purpose of the different types of checks MBSA can conduct, select the **Scanning Options** link highlighted in blue. Also, to learn about what each of the various scans searches for, use the **Microsoft Baseline Security Analyzer Help** link in the Navigation menu.

To begin the scan, click on the green arrow next to the **Start scan** option at the bottom of the window. The *Scanning* screen will appear (see Figure B.20) with an illustrative bar to track the progress of the scan.

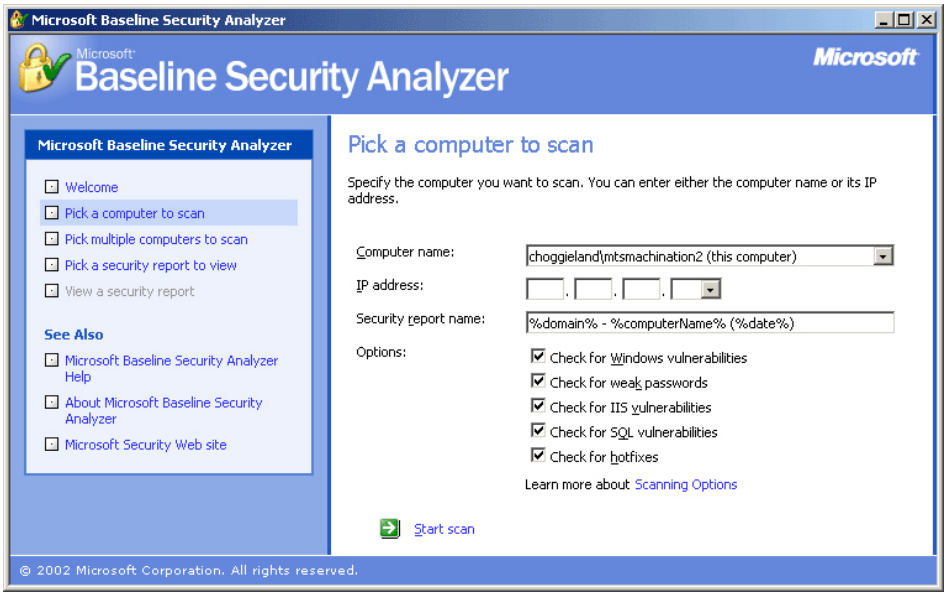


Figure B.20: MBSA Scanning Screen

When the scan completes the security report will show on the screen. For more information on reading the security report, see the Security Report section later in this appendix.

To scan more than one computer, click on the **Scan more than one computer** option from the *Welcome* screen or on the **Pick multiple computers to scan** option from the navigation menu.

The *Pick multiple computers to scan* screen will appear (see Figure B.21). Here, the computers to be scanned are specified and the scope of the scan is defined.

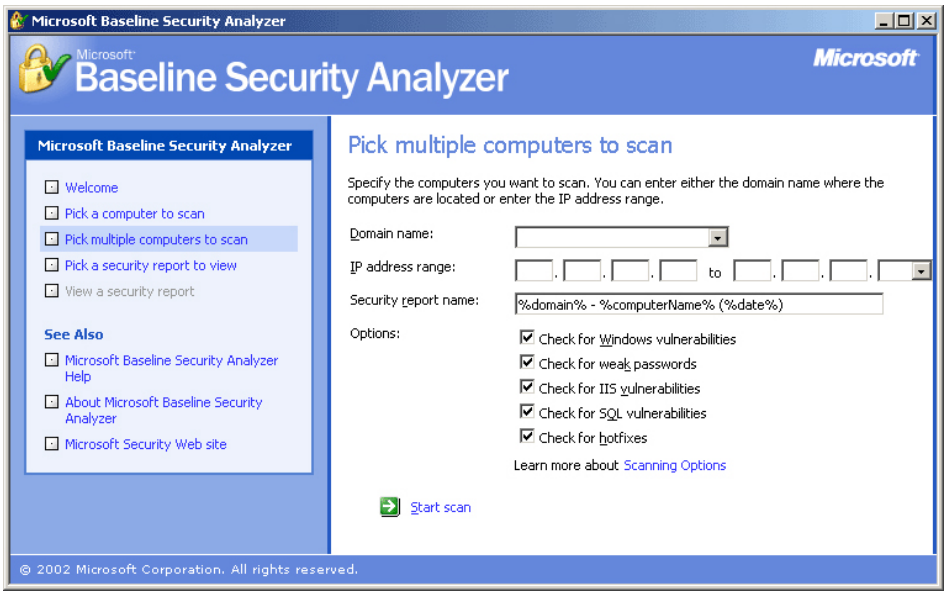


Figure B.21: Pick Multiple Computers to Scan Screen

- **Domain name** – Specifies the domain to be scanned. Enter a domain to be scanned. MBSA will discover and scan all Windows-based machines in the specified domain.

- **IP address range** – Enter the IP addresses of the first and last machines in the IP range to be scanned to specify an IP address range instead of an entire domain. All Windows-based machines found within the range will be scanned.

- **Security report name** – By default MBSA labels the security report with the domain name followed by the name of the computer scanned and the date of the scan. To rename the security report, specify the new name in this field.

- **Options** – Specify the scope of the scan. Select or deselect the areas MBSA will check for vulnerabilities as appropriate.

For more information on the benefits and/or purpose of the different types of checks MBSA can conduct, select the **Scanning Options** link highlighted in blue. Also, to learn about what each of the various scans searches for, use the **Microsoft Baseline Security Analyzer Help** link in the Navigation menu.

To begin the scan click on the green arrow next to the **Start scan** option at the bottom of the window. The *Scanning* screen will appear (see Figure B.22) with an illustrative bar to track the progress of the scan.

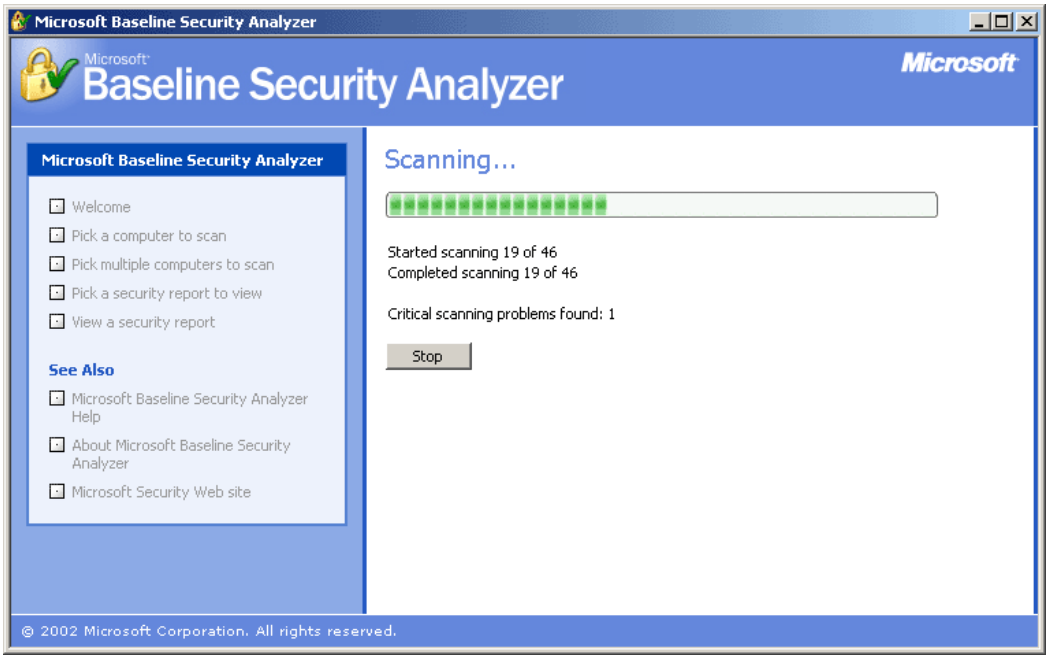


Figure B.22: MBSA Scanning Screen

When the scan completes the *Pick a security report to* view screen will show on the screen. For more information on this screen, see the Viewing a Security Report section later in this appendix.

The top portion of the security report contains summary information regarding the scan (see Figure B.23).

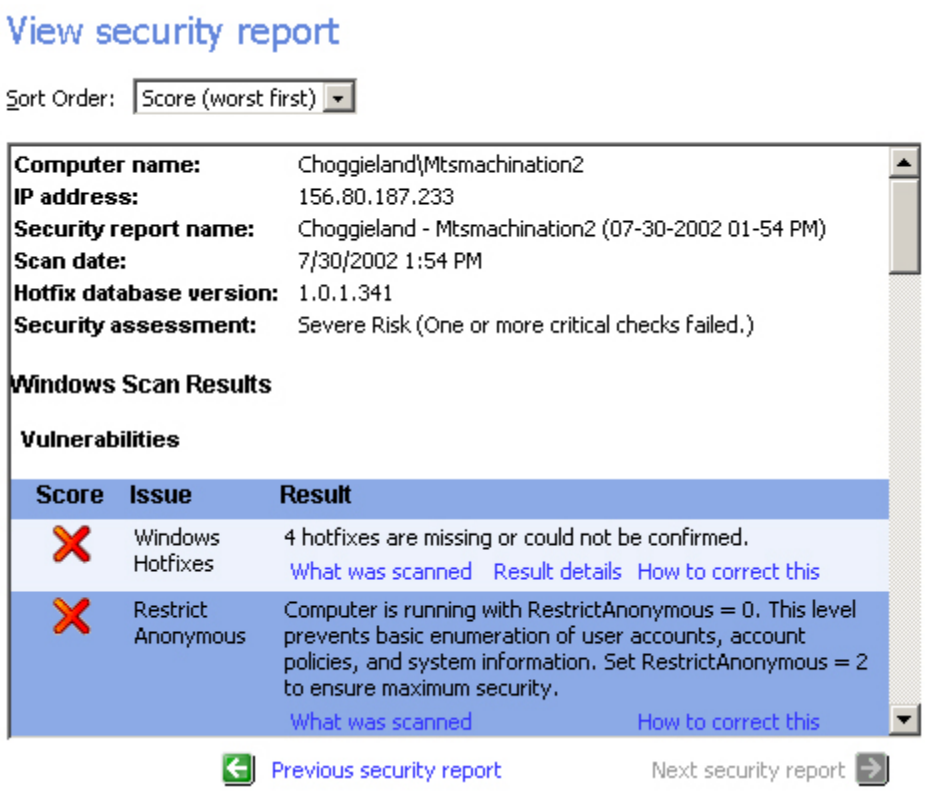


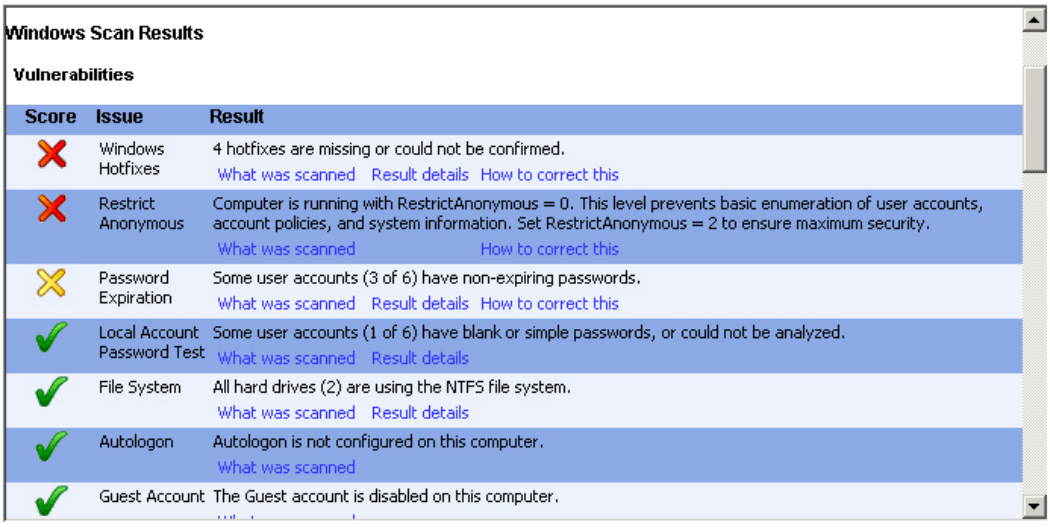
Figure B.23: MBSA Scan Summary Information

The vulnerability assessment follows below and is divided into sections. Depending on the options selected in either the *Pick a computer to scan* screen or the *Pick multiple computers to scan* screen, the report is divided into as many as four sections:

- **Windows Scan Results** – Scan results for Windows operating system vulnerabilities.
- **Internet Information Services (IIS) Scan Results** – Scan results for IIS vulnerabilities.
- **SQL Server Scan Results** – Scan results for SQL Server vulnerabilities.
- **Desktop Application Scan Results** – Scan results for desktop application vulnerabilities.

Each section contains vulnerabilities discovered by MBSA as well as any pertinent additional system information. Vulnerabilities include security vulnerabilities discovered during the scan. Additional system information includes best practice suggestions and resource information gathered by MBSA, such as operating system type and version number.

The security report is populated with issues found by MBSA during the scan. Each issue has a score and result associated with it. The score is depicted in graphical form (see Figure B.24).



Windows Scan Results

Vulnerabilities

Score	Issue	Result
✗	Windows Hotfixes	4 hotfixes are missing or could not be confirmed. What was scanned Result details How to correct this
✗	Restrict Anonymous	Computer is running with RestrictAnonymous = 0. This level prevents basic enumeration of user accounts, account policies, and system information. Set RestrictAnonymous = 2 to ensure maximum security. What was scanned How to correct this
✗	Password Expiration	Some user accounts (3 of 6) have non-expiring passwords. What was scanned Result details How to correct this
✓	Local Account Password Test	Some user accounts (1 of 6) have blank or simple passwords, or could not be analyzed. What was scanned Result details
✓	File System	All hard drives (2) are using the NTFS file system. What was scanned Result details
✓	Autologon	Autologon is not configured on this computer. What was scanned
✓	Guest Account	The Guest account is disabled on this computer.

Figure B.24: MBSA Vulnerability Assessment

To view the meaning of each score, scroll the mouse over the icon. The issues in the security report may be sorted by score (most critical vulnerability to least critical, or vice versa) or alphabetically by using the drop-down box at the top of the S*ecurity report* screen.

MBSA provides detailed information for each issue discovered during the scan, including:

- **What is scanned** – Describes what MBSA is checking for (check description) and additional resources for information regarding that particular issue.
- **Result details** – Where appropriate, MBSA offers additional information on what it discovered during the scan.
- **How to correct this** – This option describes the vulnerability issue and offers a possible solution(s) to eliminate or mitigate the risk presented by the vulnerability.

To view a security report, click on the **View existing security reports** option from the *Welcome* screen or on the **Pick a security report to view** option from the navigation menu.

The *Pick a security report to view* screen will appear (see Figure B.25) with a list of previously scanned computers.

Figure B.25: Pick a Security Report to View Screen

To change the sort order of the reports, select the appropriate option from the drop down box at the top of the window. To open any of the security reports click on the report link in blue.

To toggle between viewing all security reports and just those security reports from the most recent scan, click on the appropriate blue link to the right of the sort order drop down box.

When viewing a security report, two new options appear in the navigation menu (see Figure B.26).

Figure B.26: Print and Copy Options

To print a report click on the **Print** option and, when prompted, specify a printer to print a copy of the report. To create a copy of the report, click on the **Copy** option. This action will save a copy of the security report to the local machine's clipboard.

The links under **See Also** in the navigation menu are comprehensive resources for understanding how the tool functions.

- **Microsoft Baseline Security Analyzer Help** – Contains helpful information about the utility including:

- *System requirements* – Defines the system requirements for a computer running the MBSA utility and the system requirements for a computer to be scanned by MBSA.
- *Tool security checks* – Lists the checks MBSA conducts for Windows, IIS, SQL, and desktop applications. Click on any one of the checks for a detailed explanation of the check and a list of additional resources for further information.
- *Tool scanning options* – Describes parts of a scan that are optional and may be turned off prior to scanning a computer.
- *Command Line Options* – Describes options that can be run by running the MBSA tool from a command line instead of a graphical user interface.
- *Notes on Scanning* – Offers helpful information regarding the scanning properties of the MBSA tool.
- *Reporting Bugs or Requesting Support* – Offers instructions for reporting bugs with the product or requesting technical support for using the tool.

- **About Microsoft Baseline Security Analyzer** – Contains information about the utility including the MBSA version number, engine version number, and hotfix version number.
- **Microsoft security website** – Connects to Microsoft's security site on the Internet.

B.10 Microsoft Windows Security Checklist:

Table B.8 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.8: 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 MBSA Weekly

B.11 Microsoft Internet Information Server Security Checklist

Table B.9 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.9: 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
☐	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.secureareality.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
Open Web Application Security Project	http://www.owasp.org/
PHP Manual, Chapter 4 Security	http://www.secureareality.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

Resource/Title	URL
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 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/

Resource/Title	URL
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 4.0	http://nsa1.www.conxion.com/support/guides/sd-3.pdf
NSA Guide to the Secure Configuration and Administration of Microsoft IIS 5.0	http://nsa1.www.conxion.com/win2k/guides/w2k-14.pdf

Resource/Title	URL
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
dominosecurity.org	http://www.dominosecurity.org/
Honeynet Project	http://project.honeynet.org/
Lotus Domino Security Page	http://www.lotus.com/home.nsf/welcome/securityzone
Microsoft Internet Explorer Security Page	http://www.microsoft.com/windows/ie/security/default.asp
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

Address Resolution Protocol – A protocol used to obtain a node's physical address. A client station broadcasts an Address Resolution Protocol (ARP) request onto the network with the Internet Protocol (IP) address of the target node it wishes to communicate with, and the node with that address responds by sending back its physical address so that packets can be transmitted.

Host – 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.

Mandatory Access Control – A means of restricting access to system resources based on the sensitivity (as represented by a label) of the information contained in the system resource and the formal authorization (i.e., clearance) of users to access information of such sensitivity.

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.

Nonce – A Nonce is a randomly generated value used to defeat "playback" attacks in communication Protocols. One party randomly generates a nonce and sends it to the other party. The receiver encrypts it using the agreed upon secret key and returns it to the sender. Since the sender randomly generated the nonce, this defeats playback attacks because the replayer cannot know in advance the nonce the sender will generate. The receiver denies connections that do not have the correctly encrypted nonce.

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.

Proxy – A proxy is an application that "breaks" the connection between client and server. The proxy accepts certain types of traffic entering or leaving a network and processes it and forwards it. This effectively closes the straight path between the internal and external

networks. Making it more difficult for an attacker to obtain internal addresses and other details of the organization's internal network. Proxy servers are available for common Internet services; for example, an Hyper Text Transfer Protocol (HTTP) proxy used for Web access, and an Simple Mail Transfer Protocol (SMTP) proxy used for e-mail.

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 Administrator – A person who manages a multi-user 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, implementation, and maintenance of a Web server. 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.*				
Cim Track WSE	Linux and Windows	http://www.cimcor.com/ct_wse.htm	✓ (Linux only)	✓	$$$
Description	*File integrity checker with automatic restoration of corrupted files.*				
LANGuard File Integrity Checker	Windows 2000/NT	http://www.gfi.com/languard/		✓	Free
Description	*LANguard File Integrity Checker is a utility that provides intrusion detection by checking whether files have been changed, added, or deleted on a Windows 2000/NT system.*				
Tripwire	Windows, Unix, Linux, and Routers	http://www.tripwiresecurity.com/	✓	✓	Free to $$$
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.*				
Cronolog	Linux/Unix	http://www.cronolog.org	✓		Free
Description	*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.*				
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.*				
Swatch	Linux/Unix	http://www.oit.ucsb.edu/~eta/swatch/	✓		Free
Description	*Linux/Unix syslog analysis utility.*				
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 and 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.*				
IRIS	Windows sniffer and network analyzer	http://www.eeye.com/		✓	$$$
Description	*Iris is a data and network sniffer and traffic analyzer that collects, stores, organizes and reports on all or selected data traffic on a network.*				
Sniffit	Unix and Windows 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 and Windows 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.*				
WinDump	Windows sniffer	http://windump.polito.it/install/default.htm		✓	Free
Description	*Windows version of TCPdump.*				

$$$=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 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.*				
SAINT	Vulnerability scanner	http://www.wwdsi.com/saint/	✓		$$$
Description	*SAINT is a network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
SARA	Vulnerability scanner	http://www-arc.com/sara/	✓		Free
Description	*SARA is a freeware network-based vulnerability-scanning tool that identifies security holes on network hosts.*				
WebInspect	Web Vulnerability scanner	http://www.spidynamics.com/	✓	✓	$$$
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.*				
Secure IIS	Windows 2000/NT	http://www.eeye.com		✓	$$$
Description	*SecureIIS protects Microsoft IIS web servers from known and unknown attacks. SecureIIS wraps around IIS and works within it, verifying and analyzing incoming and outgoing web server data for any possible security breaches.*				
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

[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

[Cho02] Pete Chow, *AES Ciphersuites for TLS*, January 2002, http://www.ietf.org/internet-drafts/draft-ietf-tls-ciphersuite-06.txt

[Coop01] Russ Cooper, *10 Steps to Better IIS Security*, Information Security Magazine, August 2001, http://www.infosecuritymag.com/articles/september01/features_IIS_security.shtml

[Curt01] Matt Curtin, *Developing Trust: Online Privacy and Security*, November 2001

[MASS99] Commonwealth of Massachusetts, Executive Order 412, 1999, http://www.state.ma.us/consumer/New/privexeco.htm

[NIST98] Marianne Swanson, NIST Special Publication 800-18, *Guide for Developing Security Plans for Information Technology Systems*, http://csrc.nist.gov/publications/nistpubs/index.html

[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

[NIST02a] 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

[NIST02b] John Wack, et al., NIST Special Publication 800-42, *Guideline on Network Security Testing*, February 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

[Salt75] Jerome H. Saltzer and Michael Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE*, Volume 63, pages 1278-1308

[Sca01] Joel Scambray, et al, *Hacking Exposed Second Edition*, McGraw-Hill, 2001

[Sch00] Bruce Schneier, *Secrets & Lies: Digital Security in a Networked World*, John Wiley &Sons Inc., 2000

[SSL98] *Introduction to SSL, Netscape Communication*, Netscape Corporation, 1998, http://developer.netscape.com/docs/manuals/security/sslin/contents.htm

[WWW01] *WWW Security FAQ*, September 2001, http://www.w3.org/Security/Faq/

[Zir02] Neal Ziring, *Web Server Execution: System and Security Issues*, presented to Information Assurance Technical Framework Forum, March 1, 2002

Appendix G. Print Resources

Aviel D. Rubin, *Web Security Sourcebook*, John Wiley & Sons, 1997.

Aviel D. Rubin, *White-Hat Security Arsenal: Tackling the Threats*, Addison-Wesley Publication Corporation, 2001.

Daniel S. Janal, *Risky Business: Protect Your Business from Being Stalked, Conned, or Blackmailed on the Web*, John Wiley & Sons, 1998.

Joel Sambray, David Wond, and Mike Shema, *Hacking Exposed Web Applications*, Osborne McGraw-Hill, 2002.

L. Brent Huston (Editor), *Hack Proofing Your Ecommerce Site*, Syngress Media Inc., 2001.

Lincoln D. Stein, *How to Set Up and Maintain a Web Site*, Addison-Wesley Publications Co., 1997.

Lincoln D. Stein, *Web Security: A Step-by Step Reference Guide*, Addison-Wesley Publications Co., 1998.

Marcus Goncalves and Vinicius A. Goncalves, *Protecting Your Web Site with Firewalls*, Prentice Hall, 1997.

Michael Howard, *Designing Secure Web-Based Application for Microsoft Windows 2000*, Microsoft Press, 2000.

Simson Garfinkel and Gene Spafford, *Web Security, Privacy and Commerce* (Second Edition), O'Reilly and Associates, 2002.

Appendix H. Web Server Security Checklist

This section provides a combined version of the individual security checklists provided at the end of many sections in this document.

Web Server Security 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
☐	Availability of 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

Completed	Action
☐	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 frequently to determine new vulnerabilities

Installing and Configuring the Web Server Securely

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

Completed	Action
☐	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)
☐	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 files 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

Securing Web Content

Completed	Action
	Ensure that none of the following types of information are available on or via a public Web server
☐	Classified records
☐	Internal personnel rules and procedures

Completed	Action
☐	Sensitive or proprietary information
☐	Personal information about an organization's personnel
☐	Telephone numbers, e-mail addresses, or general listings of staff unless necessary to fulfill organizational requirements
☐	Schedules of organizational principals or their exact location (whether on or off the premises)
☐	Information on the composition, preparation, or optimal use of hazardous materials or toxins
☐	Sensitive information relating to homeland security
☐	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
☐	Information that specifies or implies physical security vulnerabilities
☐	Plans, maps, diagrams, aerial photographs, and architectural plans of organizational building, properties, or installations
☐	Copyrighted material without the written permission of the owner
☐	Privacy or security policies that indicate the types of security measures in place to the degree that they may be useful to an attacker
	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, publishing, and maintaining 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
☐	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

Completed	Action
	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 or execute function is 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

Employing Web Authentication and Encryption Technologies

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

Completed	Action
☐	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 5.2.4), 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
☐	Configure file integrity checker to monitor Web server certificate
☐	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)

Employing Network Infrastructure to Secure Web Servers

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 using SSL/TLS)
☐	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

Completed	Action
☐	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 to 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
☐	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 and hubs
☐	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)

Securely Administering a Web Server

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

Completed	Action
☐	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
☐	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 log file 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
☐	Report incident to organization's computer incident response capability
☐	Consult the organization's security policy
☐	Isolate compromised system(s) 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, as appropriate, with management, legal counsel, and law enforcement officials expeditiously
☐	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

Completed	Action
☐	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)
☐	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

Securing Apache

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

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 .htaccess 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

Securing Microsoft Internet Information Server

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

Appendix I. Acronym List

ACL	Access Control List
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ASCII	American Standard Code of Information Interchange
ASP	Active Server Page
CA	Certificate Authority
CD-R	Compact Disc Recordable
CIDR	Classless InterDomain Routing
CIFS	Common Internet File System
CIO	Chief Information Officer
CGI	Computer Gateway Interface
CN	Common Name
COM	Component Object Model
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSR	Certificate-Signing Request
DES	Data Encryption Standard
DMZ	Demilitarized Zone
DNS	Domain Name Service
DoS	Denial of Service Attack
DSS	Digital Signature Standard
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTML	Hypertext Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

ICP Internet Caching Protocol

IDS Intrusion Detection System

IETF Internet Engineering Task Force

IIS Internet Information Server

IMAP Internet Message Access Protocol

IP Internet Protocol

ISAPI Internet Server Application Program Interface

ISP Internet Service Provider

ISSO Information Systems Security Officers

ITL Information Technology Laboratory

JVM Java Virtual Machine

LDAP Lightweight Directory Access Protocol

MAC Message Authentication Code

MD5 Message Digest 5

NetBIOS Network Basic Input/Output System

NFS Network File System

NIPC National Infrastructure Protection Center

NIS Network Information System

NIST National Institute of Standards and Technology

NSA National Security Agency

NTP Network Time Protocol

ODBC Open Database Connectivity Protocol

OMB Office of Management and Budget

PDF Portable Document Format

PEM Privacy Enhanced Mail

PHP Hypertext Preprocessor

PKCS Public Key Cryptography Standard

PWS Peer Web Services

RADIUS Remote Authentication Dial-In User Service

RAM Random Access Memory

REP Robots Exclusion Standard

RFC Request for Comments

S-HTTP Secure Hypertext Transfer Protocol

SHA-1 Secure Hash Algorithim-1

SMTP Simple Mail Transfer Protocol

SOHO Small Office Home Office

SSH Secure Shell

SSI Server Side Includes

SSL Secure Socket Layer

SSN Social Security Numbers

TCP Transport Control Protocol

TLS Transport Layer

TOS Trusted Operating Systems

UDP User Datagram Protocol

URI Uniform Resource Identifier

URL Uniform Resource Locator

WWW World Wide Web

Index

3DES, 57
access controls, 28
access controls lists. See ACLs
accountability, 21, 72
ACLs, 65
action at a distance, 4
Active Server Page. See ASP
ActiveX, 40, 42, 43, 45
Address Resolution Protocol. See ARP
address-based authentication, 52
administrative accounts, 30, 36
Adobe Acrobat, 41
Advanced Encryption Standard. See AES
aerial photographs, 4, 38, 49
aerver authentication, 54
AES, 57
agent log, 78, 79, 80, 91
aliases, 30
American Standard Code of Information Interchange. See ASCII
Apache, 1, 3, 27, 31, 44
application layer firewalls, 70, 71
architectural plans, 4, 38, 49
ARP, 75, 77
ASCII, 40, 79
ASP, 30, 36, 43, 45, 46, 49
attack technique propagation, 4
attackers, 5, 8, 31, 34, 37, 43, 44, 69, 73, 86, 87
audits, 2
authentication, 3, 10, 22, 24, 25, 26, 28, 34, 37, 46, 52, 53, 56, 60, 62, 63, 71, 79, 85, 90, 92
authentication mechanisms, 22, 24
authoritative copy of the Web site, 84
automated log file analysis, 81
automation of attacks, 4
backup procedures, 82
backups, 2, 5, 82, 83, 84, 86, 91, 8
basic authentication, 52, 53
buffer overflow, 41
C++, 45
CA, 58, 59, 61, 62
Certificate Authority. See CA
certificates, 24, 57, 59, 62, 63
certificate-signing request. See CSR
CGI, 10, 26, 30, 36, 43, 44, 45, 47, 49
checksums, 34
Cherry Picker, 32, 33
chief information officer. See CIO
CIFS, 10
CIO, 11, 38
classified, 37
classified records, 37
client authentication, 54
client side active content, 40, 43, 50
COBOL, 45
COM, 46
combined log format, 79
Common Gateway Interface. See CGI
Common Internet File System. See CIFS
common log format, 79
complete mediation, 7
Component Object Model. See COM
compromise recording, 8
computer security, 13
Computer Security Act of 1987, 1
configuration control, 13
content generators, 40, 41, 44
continuity of operations, 14
continuity of operations plans, 38
cookies, 32, 40, 50, 4
CPU, 80
CRC, 34
CSR, 58, 59
cyclic redundancy check. See CRC
default accounts, 22, 23, 26, 90, 92
default configuration, 3, 20
defense in depth, 7
Demilitarized Zone. See DMZ
Denial of service. See DoS
deployment plan, 2, 9, 23
digest authentication, 53
Digital Signature Standard. See DSS
disaster recover planning, 14
disaster recovery plans, 38
DMZ, 10, 65, 66, 67, 76, 84
DoS, 1, 4, 5, 23, 28, 29, 30, 32, 42, 66, 67, 68, 73, 74, 75, 76
DSS, 57
dynamic Web pages, 29, 46
e-commerce, 4, 44, 61, 84
E-government, 44, 84
EmailSiphon, 32, 33
emergency response procedures, 38

encryption, 3, 24, 37, 48, 52, 53, 55, 57, 58, 60, 62, 63, 81, 90
error log, 78, 82
evacuation routes, 38
exploits, 8, 42
extended log format, 79
fail-safe, 7
file integrity checkers, 34, 74
File Transfer Protocol. See FTP
financial records, 38, 49
FIPS 112, 57
firewalls, 2, 1, 69
Firewalls, 69
firmware, 22, 72
FTP, 21, 27, 35, 80, 90
general-purpose operating system, 3, 17, 20, 71
Googlebot, 32, 34
hardened web server appliances, 18
honey pot, 73
host-based firewall, 11
hotfixes, 21
HTML, 42, 43, 45, 46, 62
HTTP, 2, 9, 10, 27, 28, 31, 32, 35, 53, 54, 62, 65, 66, 69, 70, 71, 76, 79, 90
HTTPS, 2, 10, 28, 69, 71, 76, 90, 92
Hypertext Preprocessor. See PHP.
Hypertext Transfer Protocol. See HTTP
Hypertext Transfer Protocol Secure. See HTTPS
ICAT metabase, 27
ICP, 10
IDS, 2, 72, 73, 74, 75
IETF, 53
IIS, 1, 3, 27, 44
IIS Lockdown Tool, 27
IMAP, 54
information system security principles, 6
Information Systems Security Officers. See ISSO
Information Systems Security Program Managers. See ISSM
Information Technology Laboratory. See ITL
Information Technology Management Reform Act of 1996, 1
intellectual property, 2
interactive elements, 40
Internet, iii, 1, 6, 4, 22, 27, 32, 35, 41, 52, 54, 58, 64, 65, 67, 69, 70, 71, 76, 79, 84, 90, 92
Internet Caching Protocol. See ICP
Internet Engineering Task Force. See IETF
Internet Explorer, 32, 42
Internet Message Access Protocol. See IMAP
Intrusion detection system. See IDS
investigative records, 38, 49
IP, 54
IP filtering, 64, 90
ISSM, 12
ISSO, 2, 12
ITL, iii
Java, 40, 41, 42, 45, 46, 62
Java servlets, 44, 46
Java Virtual Machine. See JVM
JavaScript, 40, 42
JVM, 41
LDAP, 54
least common mechanism, 7
least privilege, 7
legal counsel, 39, 87
Lightweight Directory Access Protocol. See LDAP
Limit directive, 31
links, 30, 36
Linux, 47, 62, 71
log entries, 22, 73
log files, 29, 30, 31, 35, 72, 78, 80, 81, 86, 87
logging, 17, 24, 63, 70, 72, 78, 80, 85
logging requirements, 80
malicious bots, 31, 33, 53, 62
man in the middle, 55
medical records, 38
metacharacters, 47, 51, 5
Microsoft, 1, 3, 25, 27, 42, 43, 44, 45
Microsoft Internet Information Server. *See* IIS
Misconfiguration, 4
National Institute of Standards and Technology. See NIST
National Security Agency. See NSA
NetBIOS, 21
Netscape Navigator, 42
Network Administrators, 12
Network Basic Input/Output System. See NetBIOS
Network File System. See NFS
Network Information System. See NIS
network infrastructure, 1, 6, 62, 64, 65, 69
network layer firewalls, 70
network location, 64
network sniffers, 24
network switches, 75, 77
NFS, 10, 21, 90
NIS, 21

NIST, iii, iv, 1, 6, 10, 14, 20, 21, 25, 40, 53, 69, 72, 86, 88
nonce, 53
noninteractive accounts, 23, 26
NSA, 25
OBJECT tag, 43
ODBC, 10
Office of Management and Budget. See OMB
OMB, 1
OMB Circular A-130, 1, 15
one-time password, 22, 24
Open Database Connectivity. See ODBC
open design, 7
outsourced hosting, 67
password aging, 23
password complexity, 23
password reuse, 23
patches, 2, 3, 5, 20, 21, 26, 27, 35, 64, 86, 87, 88, 89
patching, 2, 14, 20, 21, 87
PDF, 41, 50
PEM, 59
penetration testing, 24, 88, 89
Perl, 45, 46
PHP, 30, 44, 46, 47, 49
PKCS, 59
port scanning, 74, 76
Portable Document Format. See PDF
PostScript, 41
premaster secret, 56
Privacy Enhanced Mail. See PEM
proprietary information, 4, 37, 49
psychological acceptability, 7
Public Key Cryptography Standard. See PKCS
RADIUS, 10
RAM, 30
random access memory. See RAM
rcp, 21
read access, 24
referrer log, 79
referrer logs, 31
remote administration, 22, 24, 27, 35, 64, 90, 92
Remote Authentication Dial-In User Service. See RADIUS
remotely administering a Web server, 90
REP, 32
Require directive, 31
reviewing log files, 80
RFC, 54, 79, 80, 91
risk assessment, 13
risk management, 13
rlogin, 21
Robots Exclusion Standard. See REP
robots.txt, 33, 34, 36
root, 6, 10, 22, 23, 24, 25, 27, 28, 29, 31, 87
routers, 2
RSA, 57
rsh, 21
sandbox security model, 42
scripts, 3, 10, 22, 26, 27, 30, 35, 36, 41, 42, 44, 45, 47, 48, 51, 80
Secure Hash Algorithim-1. See SHA-1
Secure Shell. See SSH
Secure Socket Layer, 10, 24
securing the operating system, 20
securing Web content, 37, 49
security policy, 5, 12, 13, 14, 46, 86, 88, 91
security requirements, 3, 9, 12, 17, 20, 25, 52
security testing, 2, 5, 17, 18, 19, 24, 87, 89
security through obscurity, 7, 31
security training, 14
Senior IT Management, 11
sensitive information, 1, 4, 22, 37, 38, 39, 52
sensitivity, 15
separation of privilege, 7
service packs, 21
SHA-1, 57
shared accounts, 23
shortcuts, 30, 36
Simple Mail Transfer Protocol. See SMTP
simplicity, 6
SMTP, 21
Social Security Numbers. See SSN
software bugs, 1
Solaris, 25
spiders. See Web bots. See Web bots
SSH, 24, 85, 90
SSI, 45, 49
SSL, 10, 24, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 72, 76
SSN, 37, 40
switches, 2
symbolic links, 30
SYN flood, 30
TCP, 28, 30, 54, 55, 61, 63, 69, 70, 71, 76
telnet, 21, 90
temporary files, 29, 35
TLS, 10, 24, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 72, 76
tokens, 22, 24
TOS, 17

transfer log, 78, 79, 80, 91
Transmission Control Protocol. See TCP
Transport Layer Security, 10, 24, 53
Triple Data Encryption Standard. See 3DES
Tripwire, 60
Trusted Operating System. See TOS
UDP, 69, 70
unauthorized access, 1, 5, 41, 85
Uniform Resource Identifier. See URI
Unix, 23, 25, 28, 46, 47, 62, 71
unnecessary services, 21, 22, 26, 87
upgrades, 2, 3, 5, 20, 26, 27, 35, 88
URI, 31
URL, 29, 31, 42, 44, 53, 59, 62, 71, 74, 76, 79, 82
user accounts, 23, 24, 26
user authentication, 22
User Datagram Protocol. See UDP
user groups, 23, 26, 36, 52
user-agent, 33, 34
VBScript, 40, 42, 45
vendors, 3, 18, 20, 31, 78
Visual Basic programming language, 42
Visual Basic Script. See VBScript
vulnerabilities, 5, 41, 43
vulnerability scanner, 24, 88, 89, 92
vulnerability scanners, 88
vulnerability scanning, 24, 88, 89
Web administrator, 3, 20, 21, 22, 23, 24, 31, 33, 34, 41, 64, 74, 75, 76, 78, 80, 81, 82, 86, 87, 88
Web appliances, 18
Web application, 29, 35, 2, 3
Web bots, 32
Web browser, 1
Web browsers, 1, 53, 54, 58, 61
Web content, 2, 3, 27, 29, 30, 31, 32, 35, 36, 39, 50, 53, 69, 70, 78, 82, 84, 86
Web server, 1, 2, 3, 6, 1, 2, 3, 4, 5, 6, 9, 10, 17, 18, 20, 21, 22, 24, 25, 27, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38, 39, 40, 41, 44, 45, 46, 48, 49, 52, 55, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 88, 89, 90, 91, 92
Web server software, 27, 28, 31, 53, 78, 80
Web service, 29, 30, 35, 48
Webmasters, 2, 22, 32, 39, 41
Windows 2000, 6, 25
Windows NT, 25, 28
work factor, 7
World Wide Web. See WWW
WWW, 1, 4, 40, 43
X.509 certificate, 59