# WEB SERVER
# SECURITY TECHNICAL IMPLEMENTATION GUIDE
Version 5, Release 1

29 October 2004

## Developed by DISA for the DOD

This page is intentionally left blank.

# TABLE OF CONTENTS

**Page**

This page is intentionally left blank.

## LIST OF FIGURES



## LIST OF TABLES

This page is intentionally left blank.

## SUMMARY OF CHANGES

Changes in this document since the previous release (Version 4, Release 1, dated 29 August 2003) are listed below.

### GENERAL:

- Changed the version to Version 5, Release 0. DRAFT
- Changed references to DOD Directive 5200.28 to DOD Directive 8500.1 and DOD Instruction 8500.2 Instruction when appropriate.
- Included references to the DOD 8520.2 Instruction where appropriate.
- Changed references from DISA Enclave Security STIG to Network Infrastructure STIG where appropriate.
- Removed references to DISA where inappropriate.
- Updated all Windows NT references to read "Windows"

### SECTION 1. INTRODUCTION

- **Section 1 Introduction** – Revised
- **Section 1.1 Background** - Revised
- **Section 1.2 Purpose** - Removed
- **Section 1.3 Scope** - Revised
- **Section 1.4 Authority –** Revised
- **Section 1.5 Intended Audience** - Removed
- **Section 1.6 DISA Gold Standard** - Replaced with **DISA Information Assurance Vulnerablity Management (IAVM)** section.
- **Section 1.7 STIG Distribution –** Revised

### SECTION 2. GENERAL INFORMATION

- **Section 2.2 Recommended Process for Content Approval and Posting** - Updated
- **Section 2.4 Network Configuration** - Moved Title of table

- **Section 2.5 Levels of Access Controls to Private Web Servers** - Updated Section and added PDI

- **Section 2.6 Passwords** - Corrected PDI information

**SECTION 3. WEB SERVER SOFTWARE SECURITY**

- **Section 3.1 Software Security** - Updated Web Server software versions and included Sun JAVA System Web Server

- **Section 3.3 Access Control –** Revised

- **Section 3.7 Application Administrative Web Servers** - Corrected and revised PDI information

- **Section 3.13 Secure Socket Layer (SSL) -** Removed reference to superceded SecDef Memo Dated 17 May 2001 Public Key Enabling of Applications

**SECTION 4. WEB SCRIPTS AND PROGRAMS SECURITY**

- **Section 4.2 CGI Programs** - Updated numbering and revised PDI's

- **Section 4.9 J3EE -** Created J2EE Specification requirements

**SECTION 5. SECURITY OF OTHER RELATED WEB SERVICES**

- **Section 5.4 Web Services** - Created Web Services Requirements

- **Section 5.5 Web Application Servers** - Created Web Applications Servers Descriptions

- **Section 5.5.1 BEA Weblogic** - Created BEA WebLogic Requirements

- **Section 5.5.2 Tomcat** - Created Tomcat Requirements

**APPENDIX A. RELATED PUBLICATIONS**

- Added new publications and deleted those that no longer apply.

**APPENDIX J. LIST OF ACRONYMS**

- Made minor changes and additions as appropriate.

# 1. INTRODUCTION

Web servers, by virture of their exposure to external forces, are vulnerable to an array of assaults. Web Servers need to be publicly available, but that very availability exposes them to malicious activity. Common security methods such as firewalls, intrusion detection systems (IDSs), and code integrity checkers do not fully address a web server's particular security needs. A thorough approach to web server security is best achieved by applying a defense-in-depth strategy, which also addresses specific web server configuration and operational issues. The purpose of this document the *Web Server Security Technical Implementation Guide (STIG)* is to assist Department of Defense (*DOD*) *sites* in meeting the minimum requirements, standards, controls, and options for securing web server operations.

This *STIG* serves to assist in the securing of DOD information systems. This STIG along with its companion document the *Web Server Checklist* can be used to implement security best practice procedures on a variety of web server platforms. This STIG is also intended for use in conjunction with the appropriate Operating System (OS) STIG as well as other technology specific STIGs that may have influence on the web server or platform that it resides on. This STIG is a tool used in the design and installation phase, operation and production phase, and maintenance phases of a web server deployment. This STIG includes appendices that aid in the installation and configuration of Netscape/iPlanet, Apache, and Microsoft IIS.

## 1.1 Background

Due to the ease of access users have to web sites, web servers have become a focus for those individuals who wish to steal, damage, or deny access to an organization's information and information systems. This is consistent with a trend in malicious user behavior, which focuses on attacking applications accessible via the Internet, as opposed to attacking the operating system of the host platform. An improperly implemented web server can be attacked directly or be used as a launching point to attack an organization's internal networks or other services.

Major security forums (e.g., SANS and The Open Web Application Security Project (OWASP)) publish reports describing the most critical Internet security threats. From these reports, some threats unique to web server technology are as follows:

- Default OS and web server software installs and mis-configurations
- Broken access controls - accounts with no passwords, weak passwords, or default passwords
- Unvalidated input
- Cross site scripting
- Broken authentication and session management
- Non-existent or incomplete backups
- Non-existent or incomplete logging
- Vulnerable CGI programs and application extensions installed on web servers
- Remote data services in the Microsoft Internet Information Server (IIS)
- Global file sharing and inappropriate information sharing via NetBIOS and Windows ports 135 – 139 (port 445 in Windows 2000), UNIX NFS exports on port 2049, and Macintosh web sharing (AppleShare/IP) on ports 80, 427, and 548

It should be noted that FSO Support for the STIGs, Checklists, and Tools is only available to DOD Customers.

## 1.2 Authority

DOD Directive 8500.1 requires that "all IA and IA-enabled IT products incorporated into DOD information systems shall be configured in accordance with DOD-approved security configuration guidelines" and tasks DISA to "develop and provide security configuration guidance for IA and IA-enabled IT products in coordination with Director, NSA." This document is provided under the authority of DOD Directive 8500.1.

The use of the principles and guidelines in this STIG will provide an environment that meets or exceeds the security requirements of DOD systems operating at the MAC II Sensitive level, containing unclassified but sensitive information.

## 1.3 Scope

The requirements in this document will assist Information Assurance Managers (IAMs), Information Assurance Officers (IAOs), Security Managers (SMs), System Administrators (SAs) and Web Managers in securing web server technologies. This document will also assist in identifying external security exposures created when the site is connected to at least one Information System (IS) outside the site's control. This document does not cover issues related to style, performance, response time, or bandwidth.

This STIG addresses known security issues facing web server technologies. Although implemented differently most web server platforms provide a means to implement the security requirements in this STIG. These requirements have been written to apply to any web server platform while more specific platform guidance is documented in the Web Server Checklist Procedures Guide.

There are many functional areas of Internet and Intranet web technology that must be secured, including the following:

- Network access - Architecture
- The host operating system
- Web server software
- The application running via the Web server, to include associated scripts and data
- The database server and associated applications
- Information (e.g., account logon data) that is transmitted between client and server

This document provides the technical security policies, requirements, and implementation details for applying security concepts to web servers.

## 1.4 Writing Conventions

Throughout this document, statements are written using words such as "**will**" and "**should**." The following paragraphs are intended to clarify how these STIG statements are to be interpreted.

A reference that uses "**will**" implies mandatory compliance. All requirements of this kind will also be documented in the italicized policy statements in bullet format, which follow the topic paragraph. This will make all "**will**" statements easier to locate and interpret from the context of the topic. The IAO will adhere to the instruction as written. Only an extension issued by the Designated Approving Authority (DAA) will table this requirement. The extension will normally have an expiration date, and does not relieve the IAO from continuing their efforts to satisfy the requirement.

A reference to "**should**" is considered a recommendation that further enhances the security posture of the site. These recommended actions will be documented in the text paragraphs but not in the italicized policy bullets. Nevertheless, all reasonable attempts to meet this criterion will be made.

For each italicized policy bullet, the text will be preceded by parentheses containing the italicized Short Description Identifier (SDID), which corresponds to an item on the checklist and the severity code of the bulleted item. An example of this will be as follows "(*G111: CAT II*). "If the item presently has no Potential Discrepancy Item (PDI), or the PDI is being developed, it will contain a preliminary severity code and "N/A" for the SDID (i.e., "*[N/A: CAT III]*").

## 1.5 Vulnerability Severity Code Definitions

| | |
|---|---|
| **Category I** | Vulnerabilities that allow an attacker immediate access into a machine, allow superuser access, or bypass a firewall. |
| **Category II** | Vulnerabilities that provide information that have a high potential of giving access to an intruder. |
| **Category III** | Vulnerabilities that provide information that potentially could lead to compromise. |
| **Category IV** | Vulnerabilities, when resolved, will prevent the possibility of degraded security. |

## 1.6 DISA Information Assurance Vulnerability Management (IAVM)

The DOD has mandated that all IAVMs are received and acted on by all commands, agencies, and organizations within the DOD. The IAVM process provides notification of these vulnerability alerts and requires that each of these organizations take appropriate actions in accordance with the issued alert. IAVM notifications can be accessed at the Joint Task Force - Global Network Operations (JTF-GNO) web site, http://www.cert.mil.

## 1.7 STIG Distribution

Parties within the DOD and Federal Government's computing environments can obtain the applicable STIG from the Information Assurance Support Environment (IASE) web site. This site contains the latest copies of any STIG, as well as checklists, scripts, and other related security information.

The NIPRNet URL for the IASE site is http://iase.disa.mil/. The National Institute of Standards and Technology (NIST) site is http://csrc.nist.gov/pcig/cig.html. Access to the STIGs on the

IASE web server requires a network connection that originates from a **.mil** or **.gov** address. The STIGs are available to users that do not originate from a **.mil** or **.gov** address by contacting the FSO Support Desk at DSN 570-9264, commercial 717-267-9264, or e-mail to **fso_spt@ritchie.disa.mil**.

### 1.8 Document Revisions

Comments or proposed revisions to this document should be sent via e-mail to **fso_spt@ritchie.disa.mil**. DISA FSO will coordinate all change requests with the relevant DOD organizations before inclusion in this document.

# 2. GENERAL INFORMATION

## 2.1 Web Server Security Administration

The SA is responsible for the host operating system. Under the normal web administrative responsibilities, the Web Manager, or equivalent position, may have the following security responsibilities:

- Configure and manage the web server in accordance with this STIG and IAO guidance.
- Coordinate placement of information and scripts on the web server with appropriate authorities.
- Provide security guidance and training to personnel regarding the capabilities and features of the web server.
- Advise the IAO of any technical, operational, or security problems along with possible solutions.

The organization or activity that sponsors the web site will have web content responsibility. These persons will ensure that all information is kept current and that information and scripting placed on the web server is reviewed and approved by a configuration management authority.

- *(WA030: CAT III) The IAO or IAM will verify that local policies are developed to ensure that all information posted is reviewed and approved by appropriate authorities and as needed by the Public Affairs Officer (PAO) prior to release.*

- *(WA035: CAT III) The IAO or IAM will verify that local policies are developed to ensure that all information that is hosted on a DOD site, which originated from a DOD or other Federal organization, has been reviewed and approved for posting by the originating organization according to the Web Site Administration Policies and Procedures, dated 25 November 1998.*

- *(WA040: CAT III) The IAO or IAM will verify the approval of the Designated Approving Authority (DAA) is obtained prior to entering into any agreements with commercial Internet service providers or non-DOD web hosting service.*

- *(WA050: CAT III) The IAO will ensure that trained staff (i.e. a Web Manager and a System Administrator) are appointed for each web server.*

***NOTE:*** The Web Manager may be an additional duty or a separate role.

- *(WA140: CAT III) The IAO will ensure that web server content and configuration files are part of a routine backup program to protect the web server from damage and system failure.*

## 2.2 Recommended Process for Content Approval and Posting

Much coordination and cooperation is involved in crafting and obtaining approval for the placement of information on a DOD web site. A communications method such as email should be used as the means of notification that the files are ready for placement on the web server. The

aim is verification that the organization or program's web page guidelines for posting new content have been followed.

In cases where content is generated from a database or approved web application via an interactive user session, the system by which the database is populated or by which the application generates content operates under an approved process inherent to the system in question.

The simplest method for obtaining content approval is to view pending content via a web browser. The author should notify all necessary managers of the file location via e-mail. Several areas of content evaluation need to be addressed. The following are a sample of these issues:

Testing by organizational webmaster and/or alternate
Validation of author's functional testing
Validation of file names and locations
Review of content for potential issues
Review and approval by the organizational content manager

Once organizationally and technically approved, the organizational Web Content Manager forwards the files to be reviewed to the appropriate command representative or Office of Public Affairs as appropriate.

When submitting materials for review/approval, the following information should be submitted:

Web Page Posting Request
Name of Organizational Content Manager: _____
Department/Organization: _____
Telephone Number (direct line): _____
Name of Page Author: (If applicable)
To be Posted on: Intranet (Server_name) [ ] Internet (server_name.mil) [ ]
Password protected: YES [ ] NO [ ]
PKI-enabled: YES [ ] NO [ ]
FOUO document: YES [ ] NO [ ]
Content: _____
Web Site URL: _____
New Page[ ] Revision[ ] Major Change[ ] (List in the "Comments" section)
Specific files/documents to be reviewed: (List separately)
Comments: _____

## 2.3 Private and Public Web Servers

A DOD private web server as defined by the *Department of Defense Instruction 8520.2 states:*

"E2.1.12. DoD Private Web Server. For unclassified networks, a DoD private web server is any DoD-owned, operated, or controlled web server providing access to sensitive information that has not been reviewed and approved for release in accordance with DoD Directive 5230.9 (reference (q)) and DoD Instruction 5230.29 (referencer)). For Secret Internet Protocol Router

Network and other classified networks that are not accessible to the public, a DOD private web server is any server that provides access to information that requires need-to-know control or compartmentation."

A DOD public web server is any DOD-owned, operated, or controlled web server providing access to information that has been reviewed and approved for release in accordance with DOD Directive 5230.9 (reference (q)) and DOD Instruction 5230.29 (referencer)).

Given the DOD definition of a private web server, a public web server may reside on the SIPRNet provided that the information published does not require need-to-know control or compartmentation.

While encouraged to post appropriate materials and documents to a properly protected public or private web server all personnel and other content providers must exercise extreme caution to ensure that they do not post inappropriate material. Current examples of such material include classified data, data covered by the Privacy Act, unclassified but sensitive data (such as Human Resources or Health Care related information), contract (procurement) sensitive information, proprietary data, or FOUO information.

## 2.4 Network Configuration

Within DOD, web servers can be designated either as public or private. Web servers that support public information are highly visible targets for malicious users. From a network perspective, the following controls are relevant:

- Premise Router Filtering
- Intrusion Detection
- Firewall protection
- Connections to internal hosts and support servers

Public web servers must be isolated from internal systems. Public web servers must not have trust relationships with assets outside the confines of the DMZ or isolated separate public enclave (subnet).

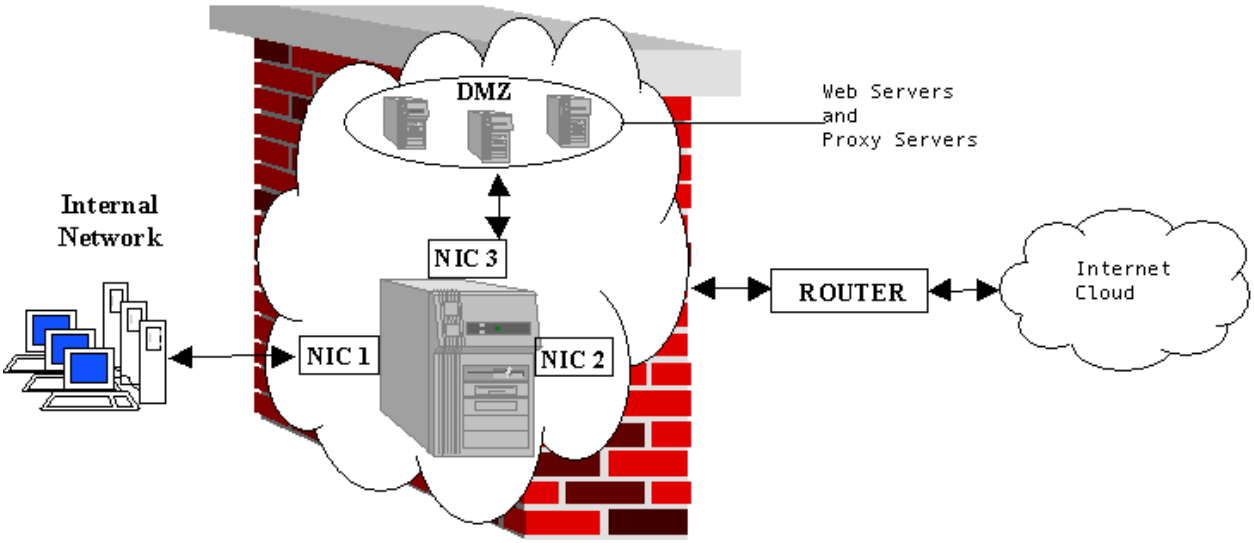


**Figure 1. Typical Web Server DMZ**

- *(WA060: CAT II) The IAO will ensure that a public web server is isolated in accordance with the Network Infrastructure STIG. A public web server is on a separate subnet isolated from the internal systems.*

- *(WG070: CAT II) The IAO will ensure that a private web server is located inside the premise router or site firewall.*

- *(WG040: CAT II) The SA will ensure that a public web server does not have a trust relationship with any asset that is not also in a separate public enclave.*

| *CONTROLS* | *SECURITY* | *DATA SENSITIVITY* |
|---|---|---|
| Public -<br>Access can be controlled by IP address or some other means where the restriction is not due to data sensitivity. | Unencrypted | Non-sensitive, of general interest to the public, cleared and authorized for public release for which worldwide dissemination poses limited risk for DOD or for DOD personnel, even if aggregated with other information reasonably expected to be in the public domain |
| Private –<br>User PKI certificate<br>Server PKI certificate | Encrypted SSL | Sensitive information that has not been reviewed and approved for release in accordance with DOD Directive 5230.9 (reference (q)) and DOD Instruction 5230.29 (referencer)) |

**Table 1. Minimum Web Server Access Control Requirements**

## 2.5 Levels of Access Controls to Private Web Servers

The *DOD Web Site Administration Policies and Procedures*, dated 25 November 1998 and the *DOD Instruction 8520.2* define access controls for private web servers based on the sensitivity of the information that will be accessed and the target audience for that information. They also provide guidance on the technology that will be used to obtain the appropriate level of protection. These technologies include, but are not limited to, IP address/domain name restriction, userid/password requirements, SSL and PKI. The Program Manager, in coordination with the IAO, will determine which technology or combination of technologies will be applied to a particular web server.

Private web servers will be protected from unauthorized remote access at the enclave perimeter and host levels. All sensitive WWW applications will use at a minimum128-bit SSL encryption and will migrate to Public Key Infrastructure (PKI). In accordance with the *DOD 8520.2 Instruction*, all private web servers will have Class 3 PKI server certificates.

- *(WA025: CAT II) The IAO will document the sensitivity level of all data for publication on a production web server.*

- *(WA080: CAT II) The IAO or Program Manager or Content Manager will determine an appropriate method to limit access to a Private web server based on the sensitivity of the data and the DOD 8520.2 Instruction.*

- *(WA140: CAT II) The IAO will ensure that a private web server use DOD PKI user certificates as an access control mechanism.*

## 2.6 Passwords

Many web-based applications involve the use of userids and passwords. In some cases these passwords are assigned to operating system accounts. In this situation, the password policy specified in the STIG for that operating system applies. In other instances, the userid and password scheme is determined by the application. In these latter cases, the application's documentation should detail the policy to be followed to add users and select or change passwords. Files containing sensitive password information should be owned by the SA or Web Manager account and reflect the minimum permissions necessary to allow the application to function as documented. The number of accounts on a web server should be limited to the least number possible to accomplish the task in question.

In cases where a Lightweight Directory Access Protocol (LDAP) server is used for authentication, the procedures for the web server suite should detail the web site's password policy.

A Private web server controls access by allowing only authorized users. Private web servers will have a connection restriction policy established to deny access to all except specifically authorized hosts or subnets. Connections from an unauthorized user will not be allowed.

The user may change passwords for web applications but the logon and password management screens must be encrypted with 128-bit secure sockets layer. Forgotten passwords should not be sent via e-mail to the user.

- *(WA150: CAT II) The IAO will ensure that in the case of web applications or servers that require restriction by userid and password, web users have a userid and password that provide access only to the web content.*

***NOTE:*** Shared user accounts are not authorized.

- *(WG050: CAT II) The IAO will ensure the web server password is entrusted to the SA or Web Manager.*

- *(WG060: CAT II) The SA or Web Manager will ensure the web server password is changed at least annually.*

***NOTE:*** WG060 Does not apply to Microsoft IIS versions 4.x and 5.x.

# 3. WEB SERVER SOFTWARE SECURITY

This section identifies the policies and requirements that must be followed when implementing a web server. A good plan will address items such as proper hardware selection, software configuration on the server, where the server will exist on the network, definition of server based on the sensitivity of the data (public vs. private), components to be installed, and security controls to be used. Prior to the installation of any web server software, the host operating system will be configured in accordance with the appropriate STIG.

- *(WA160: CAT II) The Web Manager or SA will ensure that before installing the web server application, the server host platform operating system is configured in accordance with the Enclave STIG.*

- *(WG190: CAT II) The Web Manager or SA will ensure the web server application software be a version supported by the vendor and appropriate to the host operating system of the server.*

## 3.1 Installation

To ensure a secure and functional web server, a detailed installation and configuration plan should be developed and followed. This will eliminate mistakes that arise as a result of *ad hoc* decisions made during the default installation of a server. Planners should carefully consider not attempting to support multiple services such as e-mail, databases, search engines, and indexing or streaming media on the same server that is providing the web publishing service. In order to take full advantage of these services, it is normally necessary, and often recommended, to install them on separate servers.

The Domain Name Service (DNS) is a network service to be isolated to a separate dedicated server. In the case of File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), and Network News Transport Protocol (NNTP), a well-defined need for these services should be documented by the IAO prior to their installation on the same platform as a web server. Primary and secondary Domain Controllers, in the Windows environment, will not share a common platform with a web server.

Finally, compilers and utility programs, such as office suites, graphic editors, third-party text editors, middleware development tools, and script editing GUIs, will not be permitted on production web servers. Software that supports development work that is not to be accomplished on a production server. However, vendor software may require JAVA Runtime Environment (JRE), JAVA Development Kit (JDK), or Software Development Kit (SDK) for Java support. JDK will be allowed on production servers to meet this requirement.

- *(WG080: CAT II) The Web Manager or SA will ensure that compilers are not installed on a production web server.*

- *(WG204: CAT II) The IAO will ensure that, if approved for installation, each installed Internet service (e.g., WWW, FTP, SMTP, NNTP) is located on a separate partition or drive.*

- *(WG090: CAT II) The Web Manager or SA will ensure that a web server is not installed on the same platform as a Microsoft domain controller.*

- *(WG130: CAT III) The Web Manager or SA will ensure that utility programs, traditional workstation applications, or development tools are not installed on the same platform as the production web server.*

## 3.2 Configuration

This section provides the policies and requirements regarding the secure configuration of the web server host platform operating system. The term 'superuser' refers to the userid that has no access restrictions on the host platform. For instance, in UNIX, the superuser would be the root account; in Windows, the superuser account would be the Administrator account or an account with administrative privilege. The policies in this section are applicable to both public and private web servers unless specifically addressed.

At this time, a web server configured according to the vendor's defaults will not meet the required security standards of this document. For example, a default installation setting for Netscape web servers is that automatic directory indexing is enabled. Such a setting makes it easy for malicious users to gather information about the configuration of the web server. In the case of IIS, default file extensions must be removed, and support for unlimited connections and FTP services must be disabled.

To ensure a secure web server, security controls above and beyond the defaults need to be implemented. This will enhance the security controls that have already been implemented in accordance with the operating system STIG.

- *(WG135: CAT II) The Web Manager or SA will ensure that unnecessary services are disabled from the web server, except those that are expressly permitted.*

- *(WN010: CAT II) The Web Manager will ensure the termination timeout (or equivalent parameter) is set to one second or less.*

- *(WA170: CAT II) The IAO will ensure that procedures are in place that require the SA or Web Manager to investigate any unscheduled or unanticipated disruption to the web service (i.e, this will normally involve a thorough review of the appropriate logs).*

- *(WG110: CAT II) The Web Manager will ensure the number of simultaneous requests that a web server allows is not set to unlimited.*

- *(WG520: CAT II) The Web Manager or SA will ensure the advertising of information pertaining to the operating system version, web server type and version, and web server ports is restricted.*

- *(WN020: CAT II) The Web Manager will ensure that in the case of Netscape, automatic directory indexing is turned off.*

- *(WG170: CAT II) The Web Manager will ensure the web server is configured such that a user cannot traverse from a document directory to a directory that does not contain web content.*

- *(WG170: CAT II) The Web Manager will ensure that each readable web document directory contains a default, home, index, or equivalent file.*

### 3.3 Access Controls

Many of the security problems that occur are not the result of a user gaining access to files or data for which the user does not have permissions, but rather that the user incorrectly has permissions to data the user is not authorized to access. The files, directories, and data that are stored on the web server need to be evaluated and a determination made concerning authorized access to information and programs on the server.

In most cases we can identify several types of users on a web server. These are the system SAs, web managers, auditors, authors, developers, and the clients (web users, either anonymous or authenticated). Only necessary user and administrative accounts will be allowed on the server. Accounts will be restricted to those that are necessary to maintain web services, review the server's operation and the operating system.

The Defense Originating Office (DOO) in accordance with the guidance provided in *Web Site Administration Policies and Procedures,* dated 25 November 1998 updated January 2002, initially defines these controls. The Web Manager for the web site will need to know the user community and data sensitivity (defined by the data owner) to define access restrictions for the content. Once identified, the required controls should be documented in such a manner as to address items such as who is allowed access, which persons are responsible for security, and what the process is for making changes to the web server.

The SA and web manager often work together to install and configure the web server software. The authors and developers design the web pages. Auditors monitor performance, trouble-shoot and look for breaches of security. The client uses the resources provided on the web server. Permissions on directories and subdirectories will be set to restrict access applying the least priviledge principle for operational use.

- *(WG195: CAT I) The SA will ensure that any anonymous access account is not a privileged account or a member of any group with privileged access.*

- *(WG220: CAT II) The SA or Web Manager will ensure that access to the web administration tool is restricted to the Web Manager and the Web Manager's designees.*

- *(WG230: CAT II) The IAO will ensure the SA or Web Manager performs all administrative tasks through a secure path.*

## 3.4 Restrict Remote Operations PUT and POST

Remote web users should not be able to upload files to the DocumentRoot directory structure without virus checking and checking for malicious or mobile code. A remote web user whose agency has a Memorandum of Agreement (MOA) with the hosting agency and has submitted a DOD form 2875 (*System Authorization Access Request* (SAAR)) or an equivalent document will be allowed to post files to a temporary location on the server. All posted files to this temporary location will be scanned for viruses and content checked for malicious or code. Only files free of viruses and malicious or mobile code will be posted to the appropriate DocumentRoot directory.

- *(WG235: CAT I) The SA or Web Manager will ensure that Remote authors or content providers of "Application Administrative Web Servers" are not able to upload files to the DocumentRoot directory without the use of a secure logon and secure connection.*

## 3.5 Web Log Files and Banner Page

### 3.5.1 Log Files

By reviewing log files sas, web managers, and auditors can determine site access, web resources requested, the scope and pattern of web site traffic, and the difficulties the server might be experiencing serving requests. The minimum items to be logged to achieve this are as follows:

- User ID
- Date and Time of the event
- Type of event.
- Success or failure of the event
- Successful and unsuccessful logons
- Starting and ending of access time for access to the system
- URI Query
- HTTP Status
- Referrer

The DOD 8500.2 states, "ECRR-1 Audit Record Retention - If the DOD information system contains sources and methods intelligence (SAMI), then audit records are retained for 5 years. Otherwise, audit records are retained for at least 1 year."

- *(WG240: CAT III) The SA or Web Manager will ensure that logs of web server access and errors are established and maintained.*

- *(WG250: CAT II) The SA or Web Manager will ensure that auditors are the only users with greater than read access to log files.*

***NOTE:*** This does not apply to active log files that require the system account to have full access.

- *(WA110: CAT III) The IAO will ensure that Web Access Logs are retained for a period of 1 year.*

### 3.5.2 Recommended Banner Page With Logging Policy

The following notice and consent banner has been approved by the DOD General Counsel and will be in place to make aware prospective entrants that the web site they are about to enter is a DOD Web site and their activity is subject to monitoring. From the DOD Web Site Administration Policies and Procedures Guide, "4.2. The following notice and consent banner, approved by the DOD General Counsel (reference (hh)) may be used on all DOD Web sites with security and access controls. This banner may be tailored by an organization but such modifications shall be accomplished in compliance with reference (hh), and shall be approved by the Component's General Counsel before use."
**DOD Notice:**

**NOTICE AND CONSENT LOGON BANNER**
**THIS IS A DEPARTMENT OF DEFENSE COMPUTER SYSTEM.**

**This computer system, including all related equipment, networks, and network devices (specifically including Internet access), is provided only for authorized U.S. Government use. DOD computer systems may be monitored for all lawful purposes, including ensuring that their use is authorized, for management of the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability, and operational security. Monitoring includes active attacks by authorized DOD entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied, and used for authorized purposes.**

**All information, including personal information, placed or sent over this system may be monitored.**

**Use of this DOD computer system, authorized or unauthorized, constitutes consent to monitoring of this system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use collected during monitoring may be used for administrative, criminal, or other adverse action. Use of this system constitutes consent to monitoring for these purposes.**

(COMPLIES WITH MEMORANDUM FROM GENERAL COUNSEL DATED
7 DECEMBER 1998.)

- *(WG265: CAT III) The IAO or Web Manager will ensure that an approved banner page is in place.*

### 3.6 Development Web Servers

It is recognized that in some instances, web site development for a web server must take place in an environment that replicates the ultimate physical and logical location of the web server. The preferred solution to this challenge is a test environment that simulates the production environment.

The main concern in this case is the risk to other servers located on the same subnet. Thus, providing a separate subnet for this server is recommended. A development web server installation should be documented and approved by the Program Manager, IAM, or IAO.

Development servers will comply with the standards set forth in this STIG and the STIG for the host operating system.

- *(WG260: CAT III) The SA or Web Manager will ensure that web sites still under development do not exist on a production server*.

### 3.7 Application Administrative Web Servers

As more applications are migrated to a web server interface, it is being reported that web-based applications are being delivered to Program Managers, IAMs, IAOs, SAs, and Web Managers for installation on web servers that include backend administrative components. In some cases these administrative components include freeware-based web servers with default settings. As a result, it is imperative that the SA or the Web Manager closely reviews these add-on/admin web services to ensure compliance with the security settings set forth in this STIG.

Common security concerns are ownership and permissions on the web server's files and CGI/ASP features used to accomplish administrative tasks that have been provided to the functional user or manager of the application. This may entail the creation of a group of users with unique permissions to use these features securely. Another area to examine is changes to the mime mappings that may be introduced. Finally, a test or lab environment should be used to examine these facets of any new application of this nature.

### 3.8 Classified Web Servers

When data of a classified nature is migrated to a web server, fundamental principles applicable safe guarding classified material must be followed.

- (*WG270: CAT III) The SA or Web Manager will ensure that the content of a classified web server exhibit proper labeling on each screen, be it a static page or dynamically generated page, that is appropriate to the classification of the system's content.*

- *(WA150: CAT II) The IAO will ensure that a classified web server is afforded physical security commensurate with the classification of its content (i.e., is located in a vault or room approved for classified storage at the highest classification processed on that system).*

### 3.9 File and Directory Access Rights for Web Servers

In addition to operating system restrictions access rights to files and directories can be set on a web site using the web server software. That is, in addition to allowing or denying all access rights, a rule can be specified that allows or denies partial access rights. For example, users can be given read-only access rights to files, so they can view the information but not change the files.

- *(WG270: CAT II) The SA or Web Manager will ensure that on web servers, the* htpasswd *file (if present) is owned by the SA or Web manager and has permissions of owner: read/write, group: read, and others: none (640).*

- *(WG280: CAT II) The SA or Web Manager will ensure that on web servers, the access control files, for example the .htaccess and .nsconfig files, are owned by a non-privileged web server account and have permissions of owner: read, group: none, and others: none (400).*

***NOTE:*** If the Web Manager group account has been authorized by the IAO to update and maintain the access control file the permissions would be owner: read, group: read/write, others: none (460). (Such uneven permissions will be documented.)

- *(WN040: CAT II) The SA or Web Manager will ensure that in the case of Netscape or iPlanet, the adminacl directory and its files on the administration server are accessible only by the account running the web service, the SA or Web Manager.*

- *(WA120: CAT III) The Web Manager will document the administrative users and groups that have access rights to the web server.*

- *(WG205: CAT II) The SA or Web Manager will ensure that all web server system files (web server root) are placed in a separate directory or partition from the web server document directory(ies) (web document root).*

- *(WG290: CAT II) The SA or Web Manager will ensure the web client account (i.e., IUSR_machinename, anyone, or nobody) has access to the content and necessary scripts directory structure. Access will be limited to read and where necessary execute. (In the case of IIS 4.x/5.x, execute equates to script as configured in the MMC.)*

***NOTE:*** In the case of the IIS web server, permissions for the IUSR_ accounts are specified in the Internet Services Manager console, MMC; but the NTFS permissions should be checked as well.

In cases where the web client account is used in conjunction with authentication for the completion of a management function, the client may then require and be granted write permissions. This will normally occur in conjunction with CGI or equivalent features that are designed to control the update of reports or similar files.

- *(WG210: CAT II) The SA will ensure that the web document root (web content directories) is not sharable or Network File System (NFS) mounted or exported to partitions on a private network.*

- *(WG275: CAT II) The SA will ensure that the web server, although started by superuser or privileged account, is run using a non- privileged account.*

***NOTE:*** The IIS 4.x and 5.x web server runs as system. IIS 6.x can be configured to run using a non-privileged account.

- *(WG300: CAT II) The IAO will ensure that Web Server system files conform to minimum file permission requirements.*

- *(WG310: CAT III) The IAO will ensure that a Public web server does not respond to calls by public search engines.*

***NOTE:*** This will be accomplished using the Deny All Allow list feature and a robots.txt file in the document root directory.

### 3.10 Service Packs and Patches

The software that is used to support web servers is periodically updated with vendor patches and fixes. These patches address security vulnerabilities that have been discovered on systems that have been compromised, as well as by routine updates from the vendor.

A DOD computing system is required to comply with all IAVM notices that are issued. These bulletins address specific security holes that have been identified as significant threats to the web server environment. The IAVM process does not address all patches that have been identified for the host operating system, or in this case, the web server software environment. Many of the vendors have subscription services available to notify users of known security threats. The site needs to be aware of these fixes and make determinations based on local policy and what software features are installed, if these patches need to be applied. The IAO will ensure that the site is aware of available vendor security patches. In some cases, patches also apply to middleware and database systems.

- *(WA160: CAT II) The IAO and SA or Web Manager will subscribe to the DOD-CERT/VMS bulletin mailing list.*

- *(WA170: CAT II) The IAO will ensure compliance with all IAVM notices in coordination with the SA and Web Manager.*

- *(WA180: CAT II) The IAO will ensure that all software used with the web server has all related security patches applied and documented.*

### 3.11 Microsoft Operating Systems

In some cases, currently installed hotfixes and or patches can be rendered invalid when an update is made to middleware or other software products from a third-party software vendor.

After a security patch installation, the hotfix "q" number can be found in the system registry. If server software such as Cold Fusion, Crystal Reports, or even an additional IIS plug-in is installed, certain files in the c:\%systemroot%\system32\inetsrv directory can be overwritten. This will allow the security vulnerabilities to exist once again, reverting the update that the patch

was originally intended to fix. The registry will still show the "q" number as being installed; however, the actual data on the drive has been altered.
Installations that can possibly affect the previously installed patches are as follows:

- Any IIS/MMC type software that utilizes plug-ins and or accesses systemroot/inetsrv information
- Web reporting software
- In Windows NT/2K, running the Add/Remove Windows Components screen regarding IIS (to include reinstalling IIS)

Ensure that all service packs, patches, and hotfixes are re-applied after any software mentioned above is installed or configured.

### 3.12 Public Key Infrastructure

The Public Key Infrastructure (PKI) supports the following services:

- Establishment of domains of trust and governance
- Confidentiality (sealing)
- Integrity and authentication (signing)
- Non-repudiation service
- End-to-end monitoring, reporting, and auditing of PKI services

See *Appendix E, Server Certificates,* of this document for detailed instructions for obtaining server certificates.

### 3.13 Secure Socket Layer (SSL) – Transport Layer Security (TLS)

SSL and its successor TLS are protocols that provide data security between application protocols such as HTTP (the protocol used by the web) and the networking protocol TCP/IP. SSL/TLS establishes a secure, encrypted connection between the server and an SSL/TLS-capable browser, and then encrypts and decrypts information as it is sent and received. Although both are based on Netscape's SSL 3.0 they are not interoperable. The TLS protocol does, however, provide a mechanism that allows for backward compatibility. Currently, most sites do not support the use of TLS 1.0, in this case SSL 3.0 is the preferred choice.

By encrypting data, SSL/TLS provides confidentiality and assurance that transactions are private and that information has not been altered during transmission. SSL/TLS can also authenticate the server to the browser by providing the server's certificate to the browser. The browser must be capable of using the SSL/TLS protocol, including verifying certificates and encrypting and decrypting messages. Several browsers (such as Internet Explorer and Netscape Navigator) support SSL/TLS. SSL/TLS is an open, non-proprietary protocol providing the following services: - Mutual AuthenticationIdentities of both the server and clients are authenticated through exchange and verification of their certificates.

- Privacy — All traffic between the server and the client is encrypted using a unique session key.

- Integrity — SSL/TLS protects the contents of messages exchanged between server and clients from being altered while in transit.

There are many possible scenarios for the employment of web server technology. Thus, private web servers will use, at a minimum, 128-bit SSL encryption.

- *(WG340: CAT III) The SA will ensure that private web servers use SSL/TLS security.*

### 3.14 Certificates to Control Server Access

A certificate is a digital identifier that establishes the identity of an individual or a platform. A server that has a certificate provides users with third-party confirmation of authenticity. Most web browsers perform server authentication automatically; the user is notified only if the authentication fails. The authentication process between the server and the client is performed using the SSL service. Digital certificates are authenticated, issued, and managed by a trusted Certification Authority (CA).

- *(WG350: CAT II) The IAO will ensure that each private web server has a DOD PKI server certificate.*

Refer to *Appendix E, Server Certificates,* for information on how to obtain personal certificates and server certificates.

### 3.15 Symbolic Links

As a rule, symbolic links confuse the system administration task and thus constitute poor practice in this regard. Additionally, there are numerous vulnerabilities related to applications which misuse links to temporary files as part of their installation or during the use of the application itself. Symbolic links allow a user who has accessed the system document tree to access or create additional documents (files) elsewhere in the system's file structure that are available for web access.

- *(WG360: CAT III) The SA will ensure that symbolic links are not used in the web document (content) directory tree.*

***NOTE:*** The .nsconfig file in some versions of Netscape web server software, if used, is exempt from this restriction.

### 3.16 Configuration Tools and Filters

#### 3.16.1 Microsoft

The IIS Lockdown tool is a wizard-based tool that an administrator can use to add security options. The IIS Lockdown tool wizard works by turning off unnecessary functions, thereby reducing attack surface available to attackers. The IIS Lockdown tool is available for download from the Microsoft web site. The latest version should always be used. Microsoft security policy for future versions of IIS will provide for a default installation of the web server that

follows the IIS Lockdown method. IIS Lockdown is fully configurable. IIS Lockdown's default configuration complies with most checks in this document.

URLScan is a tool that IIS administrators (Web Managers) may use to help secure their web servers. When URLScan is installed, it screens all incoming requests (e.g.HTTP requests) to the server and filters them based on rules that the administrator has set. This significantly improves the security of the server by helping to ensure that the server only responds to valid requests for service. This tool is recommended for use and will add a measure of defense in depth to any web server using IIS.

- *(WI040: CAT II) The SA or Web Manager will ensure that URLScan is used.*

### 3.16.2 Netscape

NSAPI is provided for users of iPlanet™ Web Server Enterprise Edition (iWS), Version 4.0 (and all iWS4.0 Service Packs), iWS Version 4.1 (and iWS4.1 Service Packs 1 through 6), and iPlanet Web Server FastTrack Edition Version 4.1 that cannot immediately upgrade to iWS4.1sp8 or later. NSAPI examines the HTTP headers of incoming requests. When it detects a specific malformed HTTP header it will reject the request. This helps prevent a potential data compromise. While NSAPI helps to mitigate the risk of data compromise due to the "Response Header Overflow" vulnerability, using it may incur a performance penalty.

This page is intentionally left blank.

# 4. WEB SCRIPTS AND PROGRAMS SECURITY

## 4.1 General

There are two types of web pages—static and dynamic. Static pages contain content that is displayed to the web user; no interaction with the web page is involved after it is displayed. Dynamic web pages accept and retrieve information from the web user, produce specialized or customized content, query databases, and generate web pages. This is accomplished via scripting embedded in a web page.

Scripts are programs often written in a contemporary computer language. In the context of web technologies, scripting is recognized as an effective and efficient means for implementing both client and server side actions via the web server. Because of the nature of scripting languages, scripts can be very powerful and their use must be monitored with the same digilence as a program. The term "program(s)" in this document applies to both scripts and programs.

In the case of scripts and programs developed by web authors and developers certification by the local CCB or technical group is required. The local CCB or technical group will follow the security review guidance provided in *Appendix H, Guidelines for Software Review of Vendor-provided Programs and Scripts*, before certifying a program for use on a web server.

Web developers and authors will not be allowed to install their own programs, regardless of the programming or scripting language used. (Refer to *Section 2.1, Web Server Security Administration*.)

Programs (e.g., CGI, JavaScript, JScript, PERL, VBScript, asp, aspx) will not be installed on a web server without the knowledge and consent of the Web Manager. (Refer to *Section 2.1, Web Server Security Administration.)*

- *(WA130: CAT III) The IAO will ensure that a local CCB, Program Manager (PM), or technical group reviews all programs and scripts before implementing them on the production web server.*

- *(WG370: CAT II) The IAO will ensure the Web Manager does not configure /bin/csh as a viewer for documents of type application/x-csh, application/x-sh, application/csh, or application/sh on the UNIX server.*

- *(WG380: CAT II) The Web Managers or SAs will remove all documentation, sample code, example applications, and tutorials for middleware or the web service from a production web server. Additionally, Web Managers or SAs will ensure that vulnerable programs, such as those detected by security scanning systems, are removed from the server.*

***NOTE:*** Examples of vulnerable scripts and programs include the following:

- *TextCounter Versions 1.0 - 1.2 (PERL) and 1.0 - 1.3 (C++)*
- *guestbook.cgi*
- *bndform.cgi*

- *Cachmgr.cgi*
- *Classified.cgi*
- *Count.cgi*
- *dumpenv.pl*
- *Excite Web Search Engine*
- *mail-lib.pl*
- *Glimpse (PERL scripts) Web Search Engine*
- *info2www, Versions 1.0-1.1*
- *webdist.cgi*
- *php.cgi*
- *files.pl*
- *nph-test-cgi*
- *nph-publish*
- *FormMail (PERL scripts)*
- *"phf" phone book script*

*Executables specific to Windows platform :*

- *ntalert.exe*
- *sysloged.exe*
- *tapi.exe*
- *20.exe*
- *21.exe*
- *25.exe*
- *ecware.exe*
- *nc.exe*
- *80.exe*
- *139.exe*
- *1433.exe*
- *1520.exe*
- *26405.exe*
- *i.exe*
- *newdsn.exe*
- *notworm*
- *readme.exe*
- Wink<random characters>.exe

## 4.2 CGI Programs

Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. Common applications involve acquiring data via a web page and the browser, executing the CGI application, and returning customized web content. There is a possibility of compromising security when using CGI. Compromise can occur when invalid input is used to build file names, cause a buffer-overflow, or to invoke system commands. Malicious users can provide input data (via a browser) to cause the CGI program to execute an arbitrary system command with the intent of crashing the server or producing erroneous web pages. The intent is to use this feature in a manner unintended or

unanticipated by the developer or author of the program. CGI programs that are carelessly written can grant the malicious user as much access to the server as a priveleged account.

CGI programs can be written in such languages as PERL, C, C++, shell (sh, ksh, bash, bat), JavaScript, JScript, PHP (PHP: Hypertext Preprocessor), and Windows Script(ing) Host, VBScript, C#, or Java. Each CGI program, that writes files to the server, will use a common directory for temporary files and once that task is completed, the temporary file will be deleted.

- *(WG400: CAT II) The SA or Web Manager will ensure that all CGI programs are placed in a designated (e.g., cgi-bin or equivalent) directory. This directory is owned by a non-privileged user account with permissions of owner: read/execute, group: execute, other: none (510), or more restrictive.*

- *(WG410: CAT II) The SA or Web Manager will ensure that all CGI programs are owned by the non-privileged user running the web server and have proper access controls.*

- *(WG420: CAT II) The SA or Web Manager will ensure that all CGI programs that are backups or otherwise non-operational do not exist in CGI designated directories.*

- *(WG430: CAT II) The SA or Web Manager will ensure that CGI directory contents and programs are not available to external FTP clients.*

- *(WG440: CAT II) The SA or Web Manager will ensure that all CGI programs are included in the set of files that are checked by a security monitoring software for modification.*

- *(WA030: CAT II) The Web Manager will ensure that all CGI programs used on the web server are documented, to include the language used and aim of the program, and that documentation is provided to the IAO.*

### 4.3 Improper Input

Hyper Text Markup Language (HTML) includes the ability to display selection lists, limit the length of fields to a specific number of characters, embed hidden data within forms, and specify variables provided to CGI programs. This is a great help in reducing how much error checking must be included in programs. Checks for errors from input, whether intentional or accidental, are essential because the anonymous web user can run a CGI program by simply accessing a URL. The IAO will verify that error checking is performed on all input data.
The following techniques for checking input will be used. In general, a CGI script will never accept input if any of the following exists:

- Any cookie or special tag not created by the server
- Input that exceeds the maximum length of the defined variable
- A non-alpha and non-numeric character where such characters will be used in the formation of system commands or file names, without specifically checking for and allowing such characters (e.g., quotes, tick marks, slashes, and asterisks).
- Values that are outside the defined scope of the expected value
- Microsoft Office Attachments

- Filter specific characters in dynamic elements (i.e., < > % # " ' ( ) )
- HTML input

- *(APP1020: CAT II) The IAO will ensure that the application adequately validates user inputs before processing them.*

## 4.4 Mobile Code

Mobile Code is the term given to software modules obtained from remote systems outside the enclave boundaryand then downloaded and executed on a local system without explicit installation or execution by the recipient. An example is a workstation or laptop, where the web browser is executed without explicit installation or initiation of execution by the recipient.

Category 1 (Active X, Windows Scripting Host and Shell Scripts). Technologies in Category 1 exhibit a broad functionality allowing unmediated access to host and remote system services. Category 1 technologies have known security exploits with few or no countermeasures once access is gained (e.g., all or nothing decision (run with full privileges or do not run at all)). All end systems will be configured to disallow the execution of unsigned Category 1 mobile code obtained from outside the enclave boundary.

As stated in the *DOD Instruction 8500.2*: "Category 1 mobile code is signed with a DOD-approved PKI code signing certificate; use of unsigned Category 1 mobile code is prohibited; use of Category 1 mobile code technologies that cannot block or disable unsigned mobile code (e.g., Windows Scripting Host) is prohibited. "

Category 2 (Java, MS Office VBA, Lotus Script, PerfectScript, and Postscript). Category 2 Mobile Code technologies have full functionality allowing mediated access and environment-controlled access to host system services. Category 2 technologies may have known security exploits, but also have known fine-grained, periodic, or continuous countermeasures/safeguards. Category 2 technologies may be used if they are obtained over a trusted channel (e.g., PKI server certificate, SSL/TLS, or SIPRNet) from sources specifically known to be trustworthy. All trusted channels use some form of encryption. Where feasible, protections against malicious forms of Category 2 mobile code will be employed at the end-user workstation and at the enclave boundary.

As stated in the *DOD Instruction 8500.2*: "Category 2 mobile code, which executes in a constrained environment without access to system resources (e.g., Windows registry, file system, system parameters, network connections to other than the originating host) may be used. Category 2 mobile code that does not execute in a constrained environment may be used when obtained from a trusted source over an assured channel (e.g., SIPRNET, SSL connection, S/MIME, code is signed with a DOD-approved code signing certificate)."

Category 3 (JavaScript, JScript, VBScript, PDF, and Shockwave/Flash). Technologies in Category 3 support limited functionality, with no capability for unmediated access to workstation, host, or remote system services and resources. Category 3 technologies may have a history of known exploits, but also support fine-grained, periodic, or continuous security safeguards.

Un-Categorized Mobile Code Technologies. Owing to the uncertain risk, Un-Categorized Mobile Code Technologies are prohibited unless explicitly authorized by the DOD CIO Control Board. This technology category will be blocked by all means available at the enclave boundary, workstation, and application layer.

For additional policy guidance and usage restrictions see *Assistant Secretary of Defense (C3I) Memorandum, Subject: "Policy Guidance for use of Mobile Code Technologies in Department of Defense (DOD) Information Systems," 7 November 2000* and DOD Instruction 8500.2, February 6, 2003.

## 4.5 PERL Scripts

Practical Extraction and Report Language (PERL) is an interpreted language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. The language is often used in shell scripting and is intended to be practical, easy to use, efficient, and complete. Unfortunately many widely available freeware PERL programs (scripts) are extremely insecure. This is often the result of not checking user input to a form. The safeguards noted in *Section 4.3, Improper Input*, above apply to PERL.

### 4.5.1 PERL Taint

PERL has a mechanism (taint) that protects the system from a variable that has been set from outside the program. When the data is *tainted*, it cannot be used in UNIX and C programs such as eval(), system(), exec(), pipes, or popen(). The script will exit with a warning message.

- *(APP1020: CAT II) The IAO will ensure that the application adequately validates user inputs before processing them.*

- *(WG460: CAT II) The SA or Web Manager will ensure that if PERL is being used in a web context the taint (-t) input validation-checking mechanism are used.*

For PERL Version 5, taint checking is built in and is enabled by passing the –T switch to the PERL interpreter (e.g., #!/usr/local/bin/perl -T).

## 4.6 JavaScript

JavaScript is a scripting extension of HTML. JScript is the Microsoft equivalent of JavaScript. It extends the ability of the server to respond to client events without the need for client/server communications. JavaScripts cannot exist outside of HTML code. To function, JavaScripts must be embedded in a web page. However, server-side statements that connect to databases or access the file system on the server can exist. JavaScripts is an interpreted language designed for controlling the browser. It has the ability to open and close windows, manipulate form elements, adjust browser settings, and download and execute Java applets. (Applets are mini application modules embedded in web pages [e.g., for animating a picture].) JavaScript is an object-oriented programming language used to create stand-alone applications and applets. JavaScripts that are improperly written can make the server vulnerable to outside attack by allowing unauthorized

access to sensitive server system or data files. JavaScript code can be easily manipulated and Java applets can be readily de-compiled.

- *(WG480: CAT II) The Web Manager will ensure that a JavaScript program is not used as the sole means of authentication.*

- *(WG485: CAT II) The Web Manager will ensure that a Java applet program is not used as the sole means of authentication.*

## 4.7 Java Applications

Despite the similarity of name, Java and JavaScript are two separate entities. Java is a language designed by Sun Microsystems expressly for use in the distributed environment of the Internet. Java contains a series of interlocking defenses that are in four layers:

- The language is designed to be safe, and the Java compiler ensures that the source code does not violate default security rules.

- All bytecode executed by the runtime modules is screened to ensure they obey the rules.

- The class loader ensures that classes do not violate name space or access restrictions.

- Application Program Interface (API) specific security prevents applets from performing destructive activity.

Java code may be used on DOD information systems if it is obtained from a trusted source. The resultant Java code may be in the form of a Java applet or Java application. By default, Java applets are restricted to functioning in a "sandbox" as implemented by the web browser. Java applications on the other hand may be designed to exploit any resource on a computer system.

- *(WG490: CAT III) The SA or Web Manager will ensure that only allowable Java programs such as bytecode, class files, and virtual machine types reside on the server.*

***NOTE:*** In the case of mainframe systems, the IBM Resource Access Control Facility (RACF) provides another layer of protection against malicious use of the JDK.

## 4.8 Java Servlet Engines and Java Server Pages

This combination of methods and technologies—Java Servlet Engines and Java Server Pages—builds on the CGI standard and makes that standard easier to use. The servlet engine loads Java classes to create a servlet instance when the request arrives from a Java Server Page.

Java Servlet technology provides web developers with a mechanism for extending the functionality of a web server and for accessing business systems. A servlet can be thought of as an applet that runs on the server side. Over 25 servlet engines are available to extend the functionality of web servers.

Servlets provide a component-based, platform-independent method for building web-based applications, thus they will be found in all operating system environments. Unlike proprietary server extension mechanisms (such as the Netscape Server API or Apache modules), servlets are server-and platform-independent.

JavaServer Pages (JSP) technology enables rapid development of web-based applications that are platform independent. JSP technology separates the user interface from content generation enabling designers to change the overall page layout without altering the underlying dynamic content.

When using Java Servlets Engines or Java Server Pages, any sample files that accompany the product's installation will be removed.

- *(APP1020: CAT II) The IAO will ensure that the application adequately validates user inputs before processing them.*

## 4.9 J2EE - JAVA 2 Enterprise Edition

J2EE (Java 2 Enterprise Edition) is a specification for developing enterprise and distributed applications from JavaSoft (Sun Microsystems). J2EE encompasses a large set of technologies: JavaServer Pages (JSP), Servlets, Enterprise JavaBeans (EJB), JDBC Java Naming and Directory Interface (JNDI), Java Messaging, Java Transaction Support, JavaMail and Java support for CORBA and support for XML.

J2EE is a specification, not a product. Multiple vendors have created platforms to develop and deploy J2EE environments including IBM's WebSphere, BEA's WebLogic and others.

J2EE applications are made up of components that can be deployed into different containers. These components are used to build a multitier enterprise application. For example you may have a Web-tier, EJB-tier or Application-tier. A container provides security in two forms: Declarative and Programmatic. The goal of the J2EE security architecture is to achieve end-to-end security by securing each tier. The J2EE specification assumes that a J2EE application will be integrated into an existing security architecture that implements authorization, encryption and security for the overall platform.

### 4.9.1 Declarative Security

Declarative contracts are contracts between those who develop and assemble application components and those who configure applications. In the context of application security, application providers/programmers are required to declare the security requirements of their applications in a document called a "deployment descriptor". This deployment descriptor is used to derive a security policy for use by a component's container. An application deployer then uses container-specific tools to map the application requirements that are in a deployment descriptor to security mechanisms that are implemented by J2EE containers.

Declarative security refers to an applications security structure including security roles, access control, and authentication requirements. These requirements are external to the application

meaning the container provides the security not the application itself. Changes to the security policy can be made here without changes to the underlying JSP or JAVA code.

The J2EE specification focuses primarily on authorization within J2EE components.

### 4.9.2 Programmatic Security

Programmatic security refers to security decisions that are made by security-aware applications. In this case the application not the container provides security. Programmatic security tends to be less portable in that if a security policy changes every component must be checked or changed depending on the security requirements in the current security policy.

### 4.9.3 Realms, Principals, Roles, and Role References

The J2EE specification focuses on authorization within the J2EE components for its primary security. The J2EE specification defines security terms that can be used to integrate security mechanisms with host systems that have diverse authentication mechanisms. These terms are Realms, Principals, Roles and Role References.

#### 4.9.3.1 Security Realms

A security realm is a J2EE security policy domain. This defines the way in which a user is authenticated to a component. J2EE supports Basic HTTP (the HTTP realm), HTTP Digest Authentication, Form-based authentication and Client-certificate authentication.

- *(WG340: CAT II) The SA or Web Manager will ensure that if Basic HTTP or Form-based authentication is used, SSL or TLS is used to encrypt the authentication traffic.*

- *(WG350: CAT II) The IAO will ensure that a valid DOD PKI certificate is required to authenticate to a J2EE application where Client-certificates are used.*

#### 4.9.3.2 Principals

A principal can be any entity that can be authenticated by an authentication protocol to the security service. A principal can be a user although the J2EE does not require the principal name to be the same as the user's login name.

#### 4.9.3.3 Roles

A role is a definition of the way a user will use the system, however, a role covers only a specific application or component in a J2EE server. Typical roles will be user, administrator, manager, developer, researcher, and so on. Each of these user categories is called a security role, an abstract logical grouping of users that is defined by the person who assembles the application. Assigning users to one or more authentication groups or granting privileges to users accounts usually implement a role. When an application is deployed, the deployer will map the roles to security identities in the operational environment.

- *(WA180: CAT III) The Web Manager will ensure that Security Roles are assigned with the least privilege principle applied.*

#### 4.9.3.4 Role References

A role reference is the name of a role used within the code of a J2EE application. As part of the J2EE application environment definition, the deployment descriptor, every role reference must be mapped onto a real role. The abstracting of the coded role reference from the actual role helps improve portability of a J2EE component.

### 4.10 Server Side Includes

Server Side Includes (SSIs) are a specialized form of HTML comment that allows the web server to provide web pages updated with current content. This is done by examining files with an extension of **shtml** (or any other extension requested), and replacing SSI commands with the results of the evaluation of those SSI commands that reveal details about the server configuration and provide the results serving the page to the web user or requester.

- *(WG200: CAT I) The SA or Web Manager will ensure that the capability for SSIs to execute shell (operating system) commands and programs is disabled in the web server software.*

In IIS, this setting is controlled by a radio button in the Application Settings section of the Properties/Home Directory dialog box of the web site or simply by removing the mapping to the file type shtm or shtml. In Apache web servers, the use of <exec cgi> is not permitted as this allows the execution of a file anywhere in the file system.

### 4.11 Security Settings for Windows Script Host (WSH) Scripts (IIS)

Originally, Microsoft Windows Script Host (WSH – sometimes referred to as Windows Scripting Host) did not include any mechanism for preventing the execution of WSH scripts from untrusted sources. This led many users to mistakenly assume that nothing could be done to protect a system against infection from WSH script viruses. All 32-bit Microsoft Windows operating systems contain mechanisms to protect a system against accidental infection from e-mail attachments or malicious HTML pages containing viruses.

A System Administrator can prevent scripts from being executed without removing WSH functionality from the system. A System Administrator can also specify that this behavior be valid only for certain users or for the whole system. This option is available in all 32-bit versions of Windows, but it must be activated. The way to block WSH scripts from executing differs a bit between operating systems (i.e., between Windows 2000 and Windows NT 4).

In Windows 2000 and Windows NT 4, the right to execute a file can be limited to specific user groups; so ordinary users can be blocked from executing WSH EXE files by following these steps:

1. Log on as an Administrator, and search for the files CScript.exe and WScript.exe (in the \System32 folder).

2. Right-click on each file, and choose Properties from the Context menu.

3. Click on the Security property page, click Everyone or IUSR_machinename, and uncheck Read & Execute in the Allow column.

Repeat these steps for all other user groups for which WSH is to be disabled. (Only the System and Administrator accounts have the right to execute a script. This implies the installation of Windows on an NTFS volume.)

After closing the Security property page, execution of WSH scripts is blocked for the specified users or user groups. To execute a script, log on as Administrator and execute the file.

In Windows 2000, a System Administrator can use the Microsoft Management Console (MMC) to specify applications that a user is not allowed to execute. If WScript.exe and CScript.exe have been specified, the user cannot execute scripts after the next logon.

Per *Section 4.4, Mobile Code*, Windows Script Host is a Category 1 mobile code technology. As such, its use is limited to local programs and command scripts for use by the SA or Web Manager.

- *(WG470: CAT II) The SA will assure that only privileged users (e.g., SA or Web Manager) have full control permissions to WScript.exe and CScript.exe.*

### 4.12 Active Server Pages (IIS)

Active Server Pages is a Microsoft programming environment that provides the ability to combine HTML, scripting, and components to create Internet applications that run on a web server. Files created with Active Server Pages have the extension .asp. With ASP files, a web site can be activated using any combination of HTML, scripting (such as JavaScript or Visual Basic® Scripting Edition [VBScript]), as well as components written in any language. An ASP file is simply a file that can contain any combination of HTML, scripting, and calls to components. ASP runs as a service of the web server and is optimized for multiple threads and multiple users.

When incorporating ASP into a web site, the following normally occurs:

1. The user brings up a web site where the default page has the extension .asp.
2. The browser requests the ASP file from the web server.
3. The server-side script begins to run with ASP.
4. ASP processes the requested file sequentially (top-down), executes any script commands contained in the file, and produces an HTML web page.
5. The web page is served to the browser.

Because a script runs on the server, the web server does all of the processing, and standard HTML pages can be generated and sent to the browser. This means that web pages are limited only by what the web server supports. Consequently, there are two fundamental security issues associated with this technology.

Since Active Server Pages can contain code from a variety of powerful scripting languages, these files can be designed to cause additions, deletions, or alterations to files on the web server itself, as well as those on the client machine. Consequently, code reviews must be carried out for all development efforts using this technology. Additionally, security controls must be enacted on the programmers, developers, and other web authors using this technology to ensure that only required and approved functionality is being enacted. Due to the nature of ASP files, which may contain sensitive logic and potentially reveal sensitive information about the architecture of the web server, it is vital that the end user not be able to access and examine ASP code.

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. In a **.**NET architecture this is a global.asax file.

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. Microsoft Knowledge Base article (Q232449) describes the vulnerability.

- *(WG410: CAT II) The SA or Web Manager will ensure that all CGI programs are owned by the non-privileged user running the web server and have proper access controls.*

***NOTE:*** A default.asp or equivalent welcome screen/page is the exception.

- *(WI030: CAT II) The Web Manager or SA will ensure that access to .inc files is restricted from reading by the IUSR_machinename account.*

A 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) or VBScript regular expression capabilities. The following sample code will strip a string of all invalid characters (characters that are not 0-9, a-z, A-Z, or _):

```
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 file name 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 file names:

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

Other IIS file extensions that require server-side processing, but which have been deemed vulnerable include **.**htr, .idc, .stm, .shtml, .shtm, .ida, .idq, and **.**printers. Requests to these file types can exploit a buffer overflow weakness in ISM.dll and other .dll files. These extensions if not deleted will be set to the 404.dll. This can be done manually or by the IIS Lockdown tool.

- *(WG490: CAT I) The Web Manager or SA will ensure that unused and vulnerable script mappings in IIS are removed or set to the 404.dll.*

### 4.13 ASP.NET and Open Network Environment (ONE) Web Services

The Internet is evolving from web sites that just deliver user interface pages to browsers to a generation of programmable web sites that directly link organizations, applications, services, and devices with one another. This linkage is intended to be system architecture independent. These resulting "programmable web sites" are intended to become more than passively accessed sites; they become user controlled web services. In this context, ASP.NET is a programming framework (formerly known as Active Server Pages) that can enable this new vision for web services. The SUN counterpart to ASP.NET is ONE, a predominantly Java environment.

The common language runtime provides built-in support for creating Web Services, using a programming abstraction that is consistent and familiar to both ASP.NET Web Forms developers and existing Visual Basic users. The resulting model is both scalable and extensible, and embraces open Internet standards (HTTP, XML, SOAP (Simple Object Access Protocol)), WSDL (see *Section 5, Security of Other Web Related Services*, in this STIG)) so that it can be accessed and consumed from any client or Internet-enabled device.

ASP.NET Web Forms pages are text files with an **.**aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests.

***NOTE:*** The .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests.

By taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required) an ASP.NET page is created. ASP.NET also provides support for Web Services with the .asmx file. An .asmx file is a text file that is similar to an .aspx file. These files can be part of an ASP.NET application that includes .aspx files.

### 4.14 Disable Internet Printing Protocol Support

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

Adding the following key to the registry can disable IPP:

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.

- *(WI080: CAT II) The SA will ensure that in the case of IIS the Internet Printing Protocol is disabled.*

### 4.15 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

- *(WG380: CAT II) The SA will ensure that vulnerable directories and programs have been removed from the web server.*

This page is intentionally left blank.

# 5. SECURITY OF OTHER WEB RELATED SERVICES

## 5.1 File Transfer Protocol (FTP)

FTP is a commonly exploited service that should not be installed on a server that also provides web publishing services, e-mail, or database services. In some case, FTP servers invoke parameters for other system utilities, such as tar in UNIX, without checking the validity of the parameters input by a user. This compounds the vulnerability.

FTP is primarily a tool for transferring large files or simply to provide a repository of files for users to view and download. However, there are serious security problems associated with FTP. First, FTP can allow anonymous login and/or logons using local or domain user accounts—a major back door into systems if it is enabled. Second, FTP is a non-encrypted protocol that transmits logon userids and password in clear text. Third, under Windows NT, 2000 and 2003, FTP logons are not subject to account-lockout restrictions.

FTP is designed to do the following:

- Promote the sharing of files (computer programs and/or data).
- Encourage indirect or implicit (via programs) use of remote computers.
- Shield a user from variations in file storage systems among hosts.
- Transfer data reliably and efficiently.

Because FTP is a non-encrypted protocol that transmits userids and passwords in clear text, it would be easy for a malicious user to sniff packets off your Internet link looking for user accounts and passwords, or to initiate a brute-force attack against a known user account without fear of that account being locked out. If permissions on the FTP content directory are not correctly set, the malicious user could transfer files and utilities of his choosing for execution on the targeted server.

Solutions to the problems outlined above include the following:

- Avoid using FTP if possible.

- If file transfer is necessary, configure an FTP server on a standalone system with no valuable data stored on it.

- Look for a secure file transfer solution, such as using HTTP 1.1 and SSL, for file transfer via web sites.

- Audit access to your FTP root and server in general.

- (*WG320: CAT III) The IAO will ensure that FTP write access is restricted to administrators and authorized authors.*

- *(WG520: CAT II) The IAO will ensure that FTP use of a secure file transfer solution (e.g., SSH) is restricted.*

## 5.2 Simple Mail Transfer Protocol (SMTP)

The Simple Mail Transfer Protocol (SMTP) is the cornerstone of messaging interoperability across the Internet. In 1982, the Internet Engineering Task Force (IETF) defined SMTP in Requests for Comments (RFCs) 821 and 822. The original protocol was simple and concentrated on the task of sending 7-bit plain text messages across an IP link between a client and a server. Port 25 is the default port for all SMTP operations.

Since 1982, SMTP has evolved to incorporate the changes that today's messaging environment requires. Extended SMTP (ESMTP) and Multimedia Internet Mail Extension (MIME) are the two major advances that have enabled SMTP to deliver highly functional messaging systems.

A proven mail program that does not use shell escapes may be used on a web server. In the UNIX environment, if sendmail is used by a CGI program, the program /usr/lib/sendmail will be used. The programs /usr/bin/mailx or /usr/bin/mail will never be used to send mail because these mail programs allow shell escapes.

In the Microsoft product arena, a web-based e-mail solution can be accomplished via Outlook Web Access (OWA) in Exchange 2000. IIS, which is integrated with the Windows 2000 operating system and must be installed for OWA to function, handles incoming HTTP requests from web browsers and sends HTTP responses from an Exchange 2000 Server or Outlook Web Access. IIS receives a client request, looks at the namespace, and passes the appropriate information for the context of the URL back to the web browser. If the server houses the Exchange 2000 database, Outlook Web Access uses a high-speed channel to access the mailbox store. If the server is a front-end server, Outlook Web Access directs the request to a back-end server-using HTTP. SSL/TLS provides the best level of security in an OWA situation because the entire communications session is encrypted. SSL/TLS is not an authentication mechanism itself. Rather, SSL/TLS provides a secure channel for any authentication process and ensuing transactions. Although any authentication mechanism can be used with SSL/TLS, the most common implementation is Basic with SSL/TLS.

- *(WG330: CAT II) The SA will ensure that a public web server is only capable of processing outbound e-mail.*

***NOTE:*** A public web server will not process inbound e-mail.

- *(WG340: CAT III) The SA or Web Manager will ensure that an e-mail service accessible via a web browser utilizes HTTPS (SSL/TLS) security.*

## 5.3 Instant Messaging

Use of Instant Messenger (IM) products through the Internet represents a risk to the client infrastructure and network. IM messages can contain viruses, worms, or other forms of malicious code. Unlike e-mail messages, which can be analyzed and stripped of malicious content at the mail server, Instant Messages go directly to the workstation of the IM user. In order to protect assets from these risks, access to public Internet IM sites will be blocked at the firewall. This action may result in personal mail systems not being accessible where IM and

personal mail sites are not separate. The primary providers of Internet IM services at this time are AOL, ICQ, Jabber, Yahoo!Messenger, MSN (Microsoft Network) Chat, MSN Messenger, and MS Exchange IM.

- *(IM060: CAT III) The IAO will ensure that IM servers are subject to keyword searching schemes at the discretion of the organization employing the service.*

- *(WG340: CAT III) The SA or Web Manager will ensure that IM services employ encrypted logons (userids and passwords).*

## 5.4 Web Services

Web Services is the term used to convey the notion of *conversations* or *transactions* that can take place over the Internet between applications. From an enterprise viewpoint, this is a powerful concept. Implicit in this paradigm is the notion that applications for invoicing a bill of materials will interact directly with partner systems and produce orders and shipping manifests without the need for human involvement. In this context, a web service is any service that satisfies these three conditions:

- Available over the Internet
- Uses a standard messaging system
- Not tied to any one operating system or programming language

Web Services incorporates several technologies and protocols to achieve these transactions. The eXtensible Markup Language (XML) is the data format/language by which these different applications record data. The Simple Object Access Protocol (SOAP) is the method by which the XML file is transported over HTTP, HTTPS or SMTP. Web Services Description Language (WSDL) is the means to describe a web service beyond the XML schema. Universal Data Description Language (UDDI) is a means of advertising the web service. WS-Security describes several security methods by which the SOAP message can be secured. The Web Services Interoperability Group has developed WS-Security. The technologies and languages standards discussed in this section have been adopted or are in the process of adoption by the Organization for the Advancement of Structured Information Standards (OASIS) and the WWW Consortium (W3C). Both organizations serve as the standard for Web Services technologies.
These technologies are defined or can be defined in schema format and each has a representative name space which defines its' format standard.

The term, *web portal*, is also used in this context to describe a sophisticated dissemination of document, search, and other interface features available to users in a distributed manner over the Internet. Typically, the application components will be distributed across the Internet and not reside on a single server or even a single network. The data exchanges in this environment are designed to be automatic and more efficient than the traditional model of the human (manual) involvement process of web browser requests to a web server. Nonetheless, application centric web services can augment human processes.

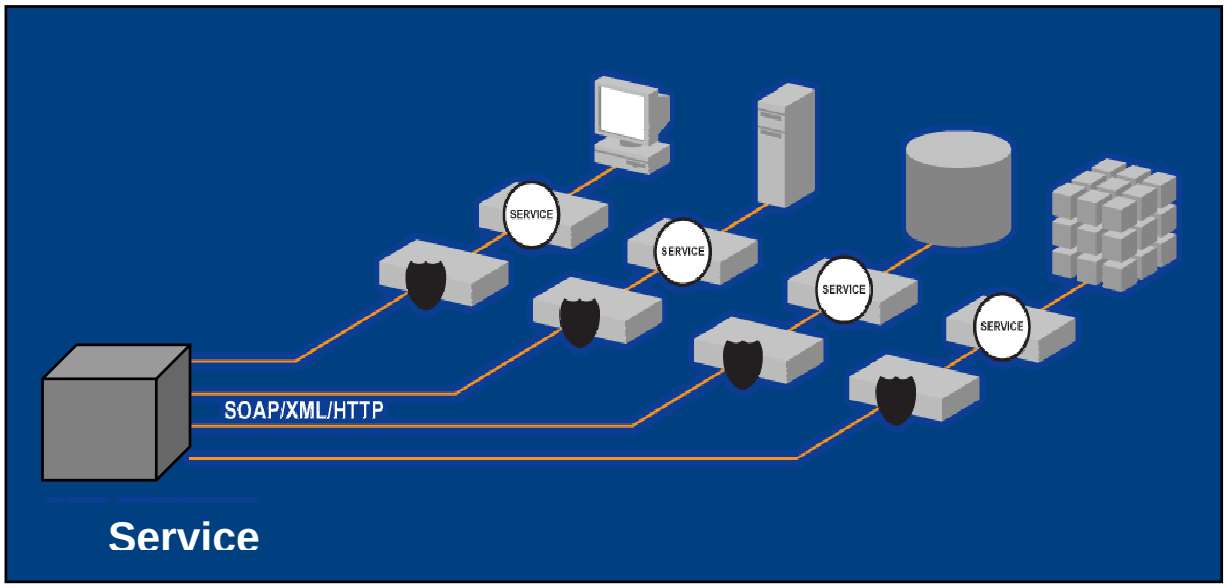


**Figure 2. Basic Web Services Architecture**

In this scenario a request is made via a web services portal (the SERVICE). The portal in tern sends the request to the appropriate services using SOAP to transport the XML over HTTP. Not shown is the opportunity for each independent service to interact with the others. This allows for platform independence and data transfer without user intervention.

### 5.4.1 XML

HTML is about displaying information; XML is about *describing* information or creating metadata—that is data about data. XML is a standard language used to structure and describe data that can be understood by different applications. XML enables diverse computer systems to share data, regardless of operating system and programming language. The data of an XML file is described in an XML Schema. This schema defines the data structure and format of the XML file. XML, while designed to provide interoperability among application components on the Web, is intended to work in many environments outside the Web, including publishing, data interchange, and commercial e-commerce applications. Finally, XML utilizes HTTP/HTTPS as a transport, allowing remote method requests and data to pass through enterprise firewalls via standard ports, such as 80/443.

Existing secure web standards, such as HTTPS and SSL/TLS, are not able to address XML specific issues such as partial document signing and the fact that XML documents are often processed in stages along loosely coupled network paths. To solve these problems, developers will use XML Encryption to encode individual parts of the XML document; XML Signature to manage the integrity of XML as it moves through the web, again along loosely coupled network paths, and XML Key Management Specification to deal with PKI verification and validation.

Simple XML: Library Example

```
<?xml version="1.0">
<Library>
        <Book>
                <Title>Green Eggs and Ham</Title>
                <Author>Dr. Seuss</Author>
        </Book>
        <Book>
                <Title>The Cat In The Hat</Title>
                <Author>Dr. Seuss</Author>
        </Book>
</Library>
```

As seen in this sample XML file data is described using tags and elements. An xml schema, which is application independent, is used to describe the format of the data in the xml file.

#### 5.4.1.1 XML Digital Signature: XML DSIG

XML Digital Signature is a way of ensuring integrity of a document. SOAP messages, wholly or in part, are first digested. The digest is a hash value equivalent to a human fingerprint. The digest, along with other sensitive data, is then digitally signed using the sender's private key and then encrypted using the receiver's public key. Because the signature is encrypted using the receiver's public key, only the receiver can decrypt it and verify the signature and message digest. Any tampering during the transmission will lead to a signature/hash verification failure.

#### 5.4.1.2 XML Data Encryption:

Sensitive data can also be encrypted using either session keys or a public/private key. Even with the message sent in the clear, the part that is encrypted will be opaque and difficult to crack. The W3C draft, XML Encryption, defines the process and format of the encrypted XML data. Both the request and response of a SOAP method are signed and verified by the SOAP client and server. In addition, any parameter values are encrypted before sending to server; and the returned values from the server are also encrypted.

### 5.4.2 SOAP

SOAP is used to send XML-based unencrypted or encrypted commands and XML messages. SOAP has been described as an envelope for XML. SOAP runs on top of HTTP and thus inherits the security holes common to HTTP implementations. SOAP conveys XML messages and is designed to pass through firewalls as HTTP, HTTPS and SMTP. In so doing, SOAP uses standard HTTP methods such as POST. Developers will define in the application permissions and rights that specify who and what has access to data, executable components and system resources.

SOAP transactions/messages can be strongly protected through digital signature and encryption.

**Authentication:**

Users of SOAP services can be authenticated in many different ways including token-based authentication and digest authentication. Token-based authentication requires users to supply credentials through a secure channel. SOAP servers respond with a token that can be used for all subsequent requests.

An example of using SOAP to send basic credentials to be used in identification and authentication is:

```
<S:Envelope>
xmlns:S=http://www.w3.org/2001/12/soap-envelope
xmlns:ws=http://schema.xmlsoap.org/ws/2002/04/secext>
    <S:Header>
    <ws:Security>
    <ws:UsernameToken>
        <ws:Username>Name</ws:Username>
        <ws:Password>password</ws:Password>
    </ws:UserToken>
    </ws:Security>
    </S:Header>
</S:Envelope>
```

### 5.4.3 WSDL

Where the XML Schema leaves off the Web Services Description Language file picks up. WSDL is a specification defining how to describe web services in a common XML grammar. WSDL describes four critical pieces of data:

- Interface information describing all publicly available functions
- Data type information for all message requests and message responses
- Binding information about the transport protocol to be used including ports
- Address information for locating the specified service

WSDL represents a contract between the service requestor and the service provider, in much the same way that a Java interface represents a contract between client code and the actual Java object. The crucial difference is that WSDL is platform and language independent and is used primarily (although not exclusively) to describe XML based Web Services.

### 5.4.4 UDDI

UDDI is the discovery layer within the web services protocol stack. UDDI is a technical specification for publishing and locating businesses and web services. Step one of UDDI is the building of a distributed directory or a registry of businesses and web services.

UDDI has been described as the Yellow Pages of Web Services or a directory of Web Services and their descriptions. A UDDI entry is an XML file that describes a business and the services it offers. A UDDI entry may contain three parts: The White Pages, The Yellow Pages and The Green Pages. The White Pages describes the business – Name, address and contacts. The Yellow Pages describes the type of business or industry. The Green Pages describe the Web Service interface. This includes a document called the Type Model or tModel. The tModel usually includes a WSDL file.

### 5.4.5 WS-Security

WS-Security is the foundation for all other Web Services security specifications. It is the fundamental way to add security to SOAP messages. WS-Security defines extensions to SOAP that provide for token passing and provides for end-to-end message level security. WS-Security describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. WS-Security also provides a general-purpose mechanism for associating security tokens with messages. No specific type of security token is required by WS-Security. It is designed to be extensible (e.g. support multiple security token formats).

Additionally, WS-Security describes how to encode binary security tokens. Specifically, the specification describes how to encode X.509 certificates and Kerberos tickets as well as how to include opaque encrypted keys. It also includes extensibility mechanisms that can be used to further describe the characteristics of the credentials that are included with a message.

SOAP Example Using x509 Certificate:

```
<WSSE:Security
xmlns:wsse=http://schemas.xmlsoap.org/wd/2002/12/secext>
<wsse:UserNameToken>
        <wsse:Username>”Name”</wsse:Username>
        </wsse:UsernameToken>
</wsse:Security>
```

- *(WG350: CAT II) The SA or Web Manager will ensure that WS-Security incorporates the use of DOD PKI certificates for authentication, integrity and confidentiality.*

#### 5.4.5.1 Security Assertions Markup Language: SAML

SAML is designed to facilitate the secure exchange of authentication and authorization information between partners regardless of their security systems or e-commerce platforms. It establishes assertion and protocol schemas for the structure of the documents that transport

security. It provides a standard way to define user authentication, authorization and attribute information in XML documents.

The main components of SAML include the following:

Assertions: SAML defines three kinds of assertions, which are declarations of one or more facts about a user (human or computer). Authentication assertions require that the user prove his identity. Attribute assertions contain specific details about the user, such as his credit line or citizenship. The authorization decision assertion identifies what the user can do (for example, whether he is authorized to buy a certain item).

Request/response protocol: This defines the way that SAML requests and receives assertions. For example, SAML currently supports SOAP over HTTP. In the future, the SAML request and response format will bind to other communications and transport protocols.

Bindings detail exactly how SAML requests should map into transport protocols such as SOAP message exchanges over HTTP.

Profiles: These dictate how SAML assertions can be embedded or transported between communicating systems.

- *(WG340: CAT II) The IAO will ensure that SAML assertions are encrypted.*

## 5.5 Web Application Servers

Application servers are servers designed to interact with web servers or web portal servers to provide applications over the web. These platforms such as BEA WebLogic, Apache Tomcat, Jboss, IBM WebSphere, Microsoft Application Center 2000 and others give developers a platform for application development and deployment. Application servers extend a Web server's capabilities to handle Web application requests, typically using Java or J2EE (JAVA 2 Enterprise Edition) technology. The application server makes it possible for a server to generate a dynamic, customized response to a client request. Application server technology provides an excellent platform for the development and production of Web Services.

These platforms can provide their own web server, http/https server, or may provide plugins for most popular web servers such as Netscape Enterprise Server, Apache, Sun Java System Web Server and Microsoft IIS.

### 5.5.1 BEA WebLogic

BEA WebLogic Server can be used as the primary Web server for advanced J2EE Web applications or a plugin can be used enabling one of several popular Web servers to act as the web interface. A J2EE Web application can be a collection of HTML or XML pages, JavaServer Pages, servlets, Java classes, applets and other file types. WebLogic server supports virtual hosting, proxy server configurations, load balancing and failover. Multiple Web applications can be hosted on a single WebLogic Server or in a clustered server configuration.

#### 5.5.1.1 Installation

Several security issues are present with a default installation of the product. Choose the "custom" option when installing WebLogic in order to remove these options.

- *(WL020: CAT III) The SA or Web Manager will ensure that XML Spy does not reside on a WebLogic server.*

- *(WL030: CAT III) The SA or Web Manager will ensure that the Pointbase database is removed from the WebLogic Server.*

- *(WL040: CAT III) The SA or Web Manager will ensure that the MedRec sample JAR file is removed from the WebLogic server.*

- *(WL050: CAT III) The SA or Web Manager will ensure that development tools such as the Configuration Wizard, WebLogic Builder and jCOM tools are removed from the WebLogic server.*

- *(WL060: CAT III) The SA or Web Manager will ensure that components of the JAVA SDK, with the exception of the JAVA JRE, are removed from the WebLogic server.*

- *(WL070: CAT III) The SA or Web Manager will ensure that if JRockit is in use, components of the JAVA SDK not in the JRocket JRE are removed from the WebLogic server.*

#### 5.5.1.2 WebLogic Administration Server

The Administration Server is the WebLogic Server used to configure and manage all the WebLogic Servers in its domain. A domain may include multiple WebLogic Server clusters and independent WebLogic Server instances. If a domain contains only one WebLogic Server, then that server is the Administration Server. The Administration Server settings are configured with an Administration Console or command-line interface that connects to the Administration Server.

- *(WL080: CAT II) The SA or Web Manager will ensure that Administrative traffic between WebLogic servers uses the secure WebLogic Administration channel.*

#### 5.5.1.3 General Security

An unpriviledged operating system account is to be created to to run WebLogic Server. This account will be granted the following permissions: read, write and execute permissions to the BEA Home Directory, WebLogic Server product directory tree and domain directories. No other user will have access to these files and directories. On a Windows platform run the WebLogic Server as a service using the operating system account created.

- *(WG200: CAT II) The SA or Web Manager will ensure that only the user created to run WebLogic has permission to the BEA Home Directory, WebLogic Server product directory tree and domain directories.*

- *(WG275: CAT II) The SA or Web Manager will ensure that on a Unix server WebLogic is run using a non-priveleged account.*

*NOTE:* This can be accomplished in one of 2 ways:

1. Start WebLogic under a privileged account, bind to privileged port (postbind) then change uid to a non-privileged account.
2. Start WeblLogic with a non-privileged account and use Network Address Translation (NAT) software to remap ports.

- *(WG275: CAT II) The SA or Web Manager will ensure that on a Windows platform, the WebLogic Server is run as a service using a non-priveleged WebLogic user account.*

WebLogic Server Security Service provides security parameters for invalid logon attempts, invalid attempt time frames, account lockout and duration of account lockout. The settings for these requirements are found in the appropriate host operating system STIG.

WebLogic Server Security Service also provides for hostname verification sometimes disabled during installation.

- *(WL090: CAT III) The SA or Web Manager will ensure that hostname verification is enabled.*

WebLogic Server Security Service provides restriction of message size settings and message time out settings.

- *(WL100: CAT II) The SA or Web Manager will ensure that Maximum message size is set no higher that 2 gigabytes.*

- *(WL110: CAT II) The SA or Web Manager will ensure that Complete message timeout is set no higher that 480 seconds.*

WebLogic Server has the ability to restrict its response header information. This means that WebLogic can be configured not to send the server's name and WebLogic Server version number as a response to a request.

- *(WL120: CAT III) The SA or Web Manager will ensure that the Send Server Header attribute is disabled.*

It is recommended that the "Servlet" and "FileServlet" servlets not be used in a production environment. Instead it is recommended to map URIs explicitly. It is recommended to remove all existing mappings between servlets and the "Servlet" servlet in a production environment.

- *(WL130: CAT II) The SA or Web Manager will ensure that the "Servlet" servlet and "FileServlet" servlet are NOT used in a production environment.*

### 5.5.2 Tomcat

The Tomcat server is a Java based web application container that was created to run Servlets and JavaServer Pages (JSP) in Web applications. Tomcat implements security based on the Java servlet specification and J2EE specification. Tomcat does not provide traditional Web server security. Tomcat does however have the ability to interact with other Web server software to act as the Web "front end" to Tomcat resources. The Apache Web server is one such choice.

Tomcat security relies primarily on the declarative and programmatic constraints described in J2EE – JAVA 2 Enterprise Edition pg. 30 of this document. The deployment descriptor for each unique web application is the web.xml file. The web.xml file is a required xml-formatted configuration file found in each web applications WEB-INF directory. The WEB-INF directory is the default location for a Web applications configuration files.

***NOTE:*** Security controls that rely on the J2EE specification will be in place to safeguard the applications.

#### 5.5.2.1 General Security

- *(WG275: CAT II) The SA or Web Manager will ensure that the Tomcat server is run using a non-privileged user account*

- *(WT040: CAT II) The SA or Web Manager will ensure that only the user created to run Tomcat has access to the WEB-INF directory.*

- *(WT060: CAT II) The SA or Web Manager will ensure that only the user created to run Tomcat has access to the WEB Applications directory.*

- *(WT070: CAT II) The SA or Web Manager will ensure that the Tomcat default administrative Web port is not in use.*

### 5.6 Collaboration (Message Board) Servers

Web message board and collaboration servers are powerful and easy to use tools accessible via a web browser. Web message board servers foster communication in corporate intranets, extranets, educational institutions, and departmental workgroups.

The Standard Version of Web Message Board Servers holds up to 100 boards and also includes chat features. Advanced versions of these servers are tailored for large communities and contain unlimited boards, enhanced management tools, and enterprise level database support.

Principal uses for Web message board servers are as follows:

- Mailing list
- Customer service/Technical support
- Online education
- Project collaboration

- Virtual meetings
- Foreign-language conferences

Such servers must be approved for use by the DAA. If so approved, the following security measures will be followed.

- *(WA070: CAT III) The IAO will ensure that all web message board and collaboration servers are implemented behind a firewall.*

- *(WG340: CAT III) The SA or Web Manager will ensure that web message board and collaboration servers employ SSL/TLS to encrypt traffic.*

- *(WG240: CAT III) The IAO and SA will ensure that message board and collaboration servers log SMTP activity, JavaScript chat, uploads, errors, activity, and all HTTP requests to the board.*

- *(WG310: CAT III) The SA or Web Manager will ensure that a web message board and collaboration servers employ robot.txt to exclude Internet search robots from cataloging the message board and collaboration site.*

## 5.7 LDAP Server Security

Light Weight Directory Access Protocol (LDAP) is an open network protocol standard designed to provide access to distributed directories. LDAP provides a mechanism to query or modify information that exists in a directory information tree (DIT). A DIT may contain a broad range of information about different types of objects that might include users, printers, applications, and other network resources.

The Netscape (iPlanet) Directory server consists of an Administrative server and a Directory server.

### 5.7.1 Limiting Access to the LDAP Server

By default, the LDAP Directory server will permit anonymous access (i.e., *read*, *search*, *compare only*) to all data in the directory. The Access Control Instruction (ACI) that controls access to data in the Directory server cannot be edited or removed permanently. Fortunately, the Deny permission overrides the Allow permission. Thus creating an ACI that denies access to data in the Directory server can eliminate this vulnerability.

To create a ACI that denies access to the data perform the following: in the Directory server, select the folders/icons for schema; monitor and config (in turn); go to Object and select Set Access Permissions. This will open the Manage Access Control for GUI for that Object.

Select New and paste in:

(target="ldap:///cn=schema")(targetattr != "objectclass") (version 3.0;acl "anonymous, deny"); deny (all)(userdn = "ldap:///anyone");)

Repeat on monitor and config objects with entry; (target="ldap:///cn=monitor") and (target="ldap:///cn=config")

Similarly, this ACI can be further modified to restrict access from a particular ip address as well. For example:

(target="ldap:///uid=*")(targetattr = "*") (version 3.0;acl "anonymous, deny"); deny (all)(userdn = "ldap:///anyone" and ip address= ”xxx.xxx.xxx.xxx”);)

Then go to Tasks and Restart the Directory Server.

The Deny will override the Allow Anonymous Access ACI that we cannot seem to delete.

Conduct this manual test from a browser: ldap:\\<ip address>/cn=schema

A negative response will be the result.

- *(WG195: CAT I) The SA or Web Manager will ensure that the anonymous user does not have access to the LDAP Schema.*

- *(WG200: CAT I) The SA or Web Manager will ensure that the anonymous user does not have access to directory content beyond that needed to authenticate.*

- *(WG230: CAT II) The SA or Web Manager will ensure that all administrative connections to the LDAP server use LLS or TLS.*

### 5.8 Web Proxy Servers

A web proxy server has two network adapters—one cabled to the Internet and one cabled to the internal network (DMZ). The proxy server software intercepts each inbound or outbound Internet message and subjects it to scrutiny before handing the message to the other network adapter. The proxy server can distinguish between the legitimate incoming messages that are responses to browsing the web site to being protected and illegitimate incoming messages not asked for by the web server (i.e., hacking attempts). The software also uses Network Address Translation (NAT) to substitute a hidden, internal IP address for the web server’s publicly known Internet IP address.

A web proxy server offers different levels of security, including packet level, circuit level, and application layer (looking inside content). It supports all significant networking transport protocols and can operate as a dedicated firewall, dedicated cache, or combination firewall/cache. A web proxy server-protected network can contain both Windows and non-Windows web servers.

- *(WG550: CAT II) The SA or Web Manager will ensure that a proxy web server filters Internet requests at the application network layer.*

- *(WG560: CAT II) The SA or Web Manager will ensure that when using a proxy web server, the internal IP address are not routable on the Internet.*

## 5.9 Wireless Enabled Web Servers

The Wireless Access Protocol (WAP) is focused on enabling the interconnection of the web server and wireless terminals. To this end, significant focus has been given to mobile telephones and pagers. The goal of WAP is to enable an extremely wide range of wireless terminals, ranging from mass-market mobile telephones and pagers to more powerful devices (i.e., PDAs), to enjoy the benefits of Web technology and interconnection.

The wireless medium is inherently uncontained, which means that maintaining security can be difficult. When a radio modem transmits information, anyone can potentially intercept that broadcast. Aside from a special order, proprietary solution, WAP phones currently do not support advanced authentication and encryption methods such as Secure Sockets Layer (SSL) or end-to-end Wireless Transport Layer Security (WTLS). There are several vendors that offer WAP-2 compliant gateways (e.g., Neomar, etc.) that provide end-to-end WTLS security so the security protocol translation described below is not necessary). In most cases, these gateways perform the WAP to HTTP translation inside the wireless gateway that is usually located inside a DMZ of the enterprise.

Currently, browser requests sent from a WAP device to WAP enabled web server are sent first as a Wireless Session Protocol (WSP) request to a Wireless Access Protocol (WAP) gateway. Most WAP browsers and gateways support Wireless Transport Layer Security (WTLS), which means the data from these devices is sent securely from the device over the air to the WAP gateway. The WAP gateway converts the request to HTTP or HTTPS (the session will be encrypted using secure sockets layer (SSL)) and establishes a session over the Internet.

The same safeguards that now protect the integrity and confidentiality of enterprise data in wireline networks also apply to wireless networks. Any wireless solution that promises to extend the reach of enterprise Intranets (LANs) must include technologies that do the following:

- Protect userids and passwords from interception by unauthorized users
- Protect corporate data from exposure

Security requirements for WAP enabled web server deployment are as follows:

- *(WW600: CAT II) The IAO/PM will ensure that a WAP enabled web server deployment Implements an IPSec policy for secure communications*

- *(WW610: CAT II) The IAO/PM will ensure that a WAP enabled web server deployment Implements HTTPS for secure browsing.*

- *(WW620: CAT II) The IAO/PM will ensure that a WAP enabled web server deployment uses wireless accounts (either auxiliary domain or Access User topologies).*

- *(WW630: CAT II) The IAO/PM will ensure that a WAP enabled web server deployment isolates the WAP enabled web server in a DMZ.*

***NOTE:*** WAP can also refer to the Wireless Application Protocol (WAP) Forum that has introduced a set of protocols optimized for wireless networks that complement existing Internet-standard protocols.

This page is intentionally left blank.

# APPENDIX A. RELATED PUBLICATIONS

## *GOVERNMENT PUBLICATIONS*

Department of Defense (DOD) Directive 8500.1, “Information Assurance,” 24 October 2002.

Department of Defense (DOD) Instruction 8500.2, “Information Assurance IA Implementation,” 6 February 2003.

Department of Defense (DOD) Instruction Number 8520.2 issued April 2004 “Public Key Infrastructure (PKI) and Public Key (PK) Enabling.”

DISA Memorandum: DISA Web Policy, Enforcement, and Operational Security, 12 March 2003.

DISA World Wide Web Handbook Version 5.0.

DOD Web Policy, “Web Site Administration Policies and Procedures,” 25 November 1998 (updated 11 January 2002). (Also see http://www.defenselink.mil/webmasters, DOD Web Site Administration Policy.)

Chairman of the Joint Chiefs of Staff (CJCS) Manual 6510.01, "Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)," 15 March 2002.

Defense Information Systems Agency (DISA) OS/390 Security Technical Implementation Guide, Version 4, Release 1 (2 volumes), 4 August 2003.

Department of Defense Directive 5200.40, “DOD Information Technology Security and Accreditation Process (DITSCAP),” 30 December 1997.

Defense Information Systems Agency Instruction (DISAI) 630-230-19, “Security Requirements for Automated Information Systems (AIS),” July 1996.

Defense Information Systems Agency Instruction (DISAI) 630-255-7, “Internet, Intranet, and World Wide Web,” 6 September 1996.

Defense Information Systems Agency Instruction (DISAI) 630-230-31, “Enclave Security,” 30 March 2001.

Defense Information Systems Agency (DISA) Naming Convention Standards, February 1996.

Defense Information Systems Agency (DISA) Computing Services Security Handbook, Version 3, 1 December 2000.

Defense Information Systems Agency (DISA) Application Security Checklist v2 r1.4

Defense Information Systems Agency (DISA) Network Infrastructure Security Technical Implementation Guide, Version 4, Release 2.

Addendum to the NSA Guide to Securing Microsoft Windows NT Networks and NSA Guides to Securing Windows 2000, Version 43 (to match NSA Guide), Release 1, 26 November 2002.

Defense Information Systems Agency (DISA) UNIX Security Technical Implementation Guide, Version 4, Release 2..

National Security Agency (NSA), "Information Systems Security Products and Services Catalog" (Current Edition).

National Institute of Standards and Technology (NIST), "Guidelines on Securing Public Web Servers," Special Publication 800-44.

Defense Logistics Agency Regulation (DLAR) 5200.17, "Security Requirements for Automated Information and Telecommunications Systems," 9 October 1991.

AR 25-2, Information Assurance, dated 14 November 2003.

Air Force Systems Security Instruction (AFSSI) 5021, *Time Compliance Network Order (TCNO) Management and Vulnerability and Incident Reporting*, 15 August 1996.

Air Force Systems Security Instruction (AFSSI) 5023, *Viruses and Other Forms of Malicious Logic*, 1 August 1996.

Air Force Systems Security Instruction (AFSSI) 5027, *Network Security Policy*, 27 February 1998.

Secretary of the Navy Instruction (SECNAVINST) 5239.2, "Department of the Navy Automated Information Systems (AIS) Security Program," 15 November 1989.

Navy Staff Office Publication (NAVSO Pub) 5239-15, "Controlled Access Protection Guidebook," August 1992.

Public Law 100-235, 100th Congress, An Act cited as the "Computer Security Act of 1987," 8 January 1988.

Memorandum for Secretaries of Military Departments, et al, "Web Site Administration," 7 December 1998.

Memorandum for Secretaries of Military Departments, et al, "DOD Public Key Infrastructure," 12 August 2000.

Memorandum for Secretaries of Military Departments, et al, "Policy Guidance for the Use of Mobile Code Technologies in Department of Defense (DOD) Information Systems," 7 November 2000.

## *OTHER PUBLICATIONS*

### *International Business Machines Corporation*

OS/390 HTTP Server Planning, Installing and Using, Version 5.2 (SC31-8903)

OS/390 HTTP Server Planning, Installing and Using, Version 5.3 (SC31-8690)

## *GENERAL INFORMATION SITES*

| | |
|---|---|
| http://iase.disa.mil | Defense Information Systems Agency Information Assurance |
| http://www.disa.mil/handbook/toc.html | DISA/NCS World Wide Web Handbook, Version 2 |
| http://www.cert.mil | Department of Defense Computer Emergency Response Team (CERT) |
| http://www.cert.org | A focal point for the computer security concerns of Internet users |
| http://csrc.nist.gov/publications | National Institute of Standards and Technology's Computer Security Resource Clearinghouse |
| http://www.cerias.purdue.edu | Center for Education and Research in Information Assurance and Security (formerly COAST) |
| http://www.redbooks.ibm.com/ | "How to" books, written by very experienced IBM professionals from all over the world |
| http://www.microsoft.com/technet/security/current.asp | Microsoft Security Bulletin and Patch Listings |
| http://www.netscape.com/security/notes/index.html | Netscape Security |
| http://hoohoo.ncsa.uiuc.edu/cgi/security.html | Writing secure CGI scripts |
| http://language.perl.com/faq/ | PERL FAQ |
| http://www.cis.ohio-state.edu/cs/Services/rfc/rfc.html | RFC Index |
| http://www.nipc.gov | National Infrastructure Protection Center (an FBI program) |

| | |
|---|---|
| http://www.defenselink.mil/webmasters | DOD Web Site Administration Policy |
| http://www.ibm.com/software/webservers/ | IBM HTTP Server documentation |
| http://java.sun.com/j2ee/tutorial/ | Sun JAVA Tutorials and Documentation |
| http://www.samspublishing.com/ | Articles and documents on J2EE Security and other systems |
| http://www.oasis-open.org | Information Resourceson Web Services |
| http://www.w3.org/ | Information and Resources on everything Web |
| http://www.bea.com | Resource for BEA WebLogic and J2EE framework |
| httpservers/library.html | |

# APPENDIX B. NETSCAPE SERVER CONFIGURATION

This appendix provides uniform resource locators (URLs) pointing to documentation, provided by Netscape Communication Corporation, pertaining to the administration and maintenance of Netscape Administration Server and Netscape Enterprise Server. (In the previous release of this document, the web content was extracted and printed in this document.)

## B.1 Managing Netscape Servers

http://developer.netscape.com/docs/manuals/enterprise/mngserv/index.htm

## B.2 Configuring Netscape Enterprise Server 4.1

### B.2.1 Menu Page for all Environments:

http://developer.netscape.com/docs/manuals/

The following sections describe the installation of the Netscape Enterprise Server in both the UNIX and NT environments.

WARNING**:** You must install your Version 4.0 servers in a server root directory separate from the server root directory that contains your 3.6 servers. Do not install Enterprise Server on an NFS-mounted drive due to potential security and file locking restrictions on remote partitions.

Logging In as the Correct User:

Before you install the server, you must log in as root unless you meet both of these conditions:

- You plan to install the server on a port greater than 1024.
- The location where you plan to install the server (the server root directory) is writable with a non-root logon.

If you meet both conditions, you do not need to log on as root to install the server; instead log on as the user account that the Enterprise Administration Server will use. However, you may still prefer to log on as root, even though you meet the conditions.

WARNING: You must log on as root if you are planning to use express installation.

Unpacking the Files:

To get the Netscape Enterprise Server files and unpack them, follow these steps:

Installing from a CD-ROM:

1. Put the CD-ROM in the drive.

2. Change to the Enterprise Server CD in the CD-ROM directory.

3. Change to the directory on the CD-ROM labeled with the UNIX operating system that your computer uses (e.g., type cd solaris).

4. Type cd entprise to change to the installation directory.

5. Copy the entprise.tar file from the CD-ROM directory to your home directory or a temporary directory. Copying the file may take a little time.

6. Change to the directory on your UNIX machine where you copied the file.

7. Untar the file by typing tar -xvf entprise.tar.

This command unpacks the server files and creates a temporary directory structure under the current directory. Unpacking the file may take a little time. When the files are unpacked, you see an Enterprise directory, and three files—LICENSE.TXT, setup, and setup.inf.

Installing a Downloaded Server:

1. Download the file from the Netscape web site http://home.netscape.com and save it in a temporary directory.

2. Change to the directory on your UNIX machine where you copied the file.

   Unpack the **.**gz file by typing gunzip filename.tar.gz. The file name is in the format:

   enterprise-4.[x]-security.platform.tar.gz

   (For example: enterprise-4.[x]-export-us.hppa1.1-hp-hpux11.00.tar.gz)

3. Untar the unzipped file by typing tar -xvf filename.tar.

   This command unpacks the server files and creates a temporary directory structure under the current directory. Unpacking the file may take a little time. When the files are unpacked, you see an Enterprise directory and three files—LICENSE.TXT, setup, and setup.inf.

Running Setup:

Run the setup program to install Enterprise Server.

During the installation process, you have the choice of three kinds of installation—express, typical, or custom. Most users should choose typical or custom (they are identical). Typical or custom gives you more flexibility in the components you can install and the settings you can configure.

Express installation is for users who have little experience or are evaluating the product. It makes assumptions about such things as port number and which components to install.

When running the installer, use the following commands:

- Press the Enter key to accept defaults specified in brackets (e.g., [All]).

- Press CTRL+b to return to a previous screen

***NOTE:*** This sequence does not work on all screens.

- Press CTRL+c to cancel the installation program.

- Enter comma-separated lists of numbers when you want to select multiple items (e.g., 1, 2, 3).

***NOTE:*** For express installation, you must be logged in as root when running setup.

Typical or Custom Installation:

Most users should install the typical or custom installation.

To run setup, follow these steps:

1. If you are not in the directory already, change to the directory where you unpacked the file.

2. Type **./**setup to start the server installation.

   If you are not logged in as the root user (superuser), or if you do not have sufficient *write* permissions, you will get one or more error messages.

   A welcome screen appears.

3. Press Enter to continue with the installation.

4. Choose whether you accept the software license agreement by typing Yes to accept, or No to decline.

5. Choose Typical or Custom.

   Press Enter.

6. Type a server root directory or accept the default (/usr/netscape/server4).

   This directory is where the server files and directory structure will be installed.

   You need to run the server as a user that has *write* access to this directory (e.g., the directory owner).

   If you already have other Netscape 4.x servers installed, and you want to register Enterprise Server with Netscape Console, install all 4.x servers into the same directory.

7. Choose Netscape Enterprise Server.

   Press Enter.

8. Choose the Enterprise Server components to install.

   If you do not install a component and later decide you want to use it, you can run the installer again to install just the missing component. However, you cannot uninstall individual components once they are installed.

By default, all components except WAI support are installed.

Netscape Enterprise Server Core:

1. Netscape Enterprise Server Core installs Enterprise Administration Server and the first instance of Enterprise Server. The six components you can choose to install are described in the following numbered sections.

   WARNING: You must install the Netscape Enterprise Server Core component the first time you install Enterprise Server. If you install additional components later, you are not required to reinstall the Core component.

2. Java Runtime Environment

   If you are planning to use Java, you must have a Java Runtime Environment (JRE) or a Java Developer's Kit (JDK). You can install the provided JRE, or supply your own JDK. For more information, see *Installing a Java Developer's Kit*.

   WARNING: If you are using HP-UX, you must install the JRE or supply a JDK. Without one or the other, the HP-UX Enterprise Server will not run.

3. Netscape Enterprise Server Java Support

   Install this component if you are planning to use Java servlets.

   You also need a JRE or JDK to use Enterprise Server's Java support. Install the JRE provided with Enterprise Server (see the previous component) or install a JDK.

4. Netscape Enterprise Server Server Side JAVA Script (SSJS) Database Support

   Install this component if you are planning to use server-side JavaScript database connectivity (LiveWire). If you plan to use this feature, you also need to install Netscape Enterprise Server Java Support (item 3), and either install the JRE (item 2) or supply a JDK.

5. Netscape Enterprise Server WebPub Support

   Install this component if you are planning to use the Web Publishing, Netshare, or search features.

6. Netscape Enterprise Server SNMP Support

   Install this component if you are planning to use SNMP.

7. Netscape Enterprise Server WAI Support

   - Install this component if you are planning to run Web Application or Web Services Interface programs. If you are using a Web Application Interface (WAI), you must also have an Object Request Broker (ORB). You will have to provide the installation program with the path to the ORB on your system later in the installation process.

   - Enter the machine name or accept the default.

   - Enter the UNIX user and group names to use when running the default instance of Enterprise Server.

   - The default user is nobody.

   - Enter the UNIX username to use when running the Enterprise Administration Server.

     In most cases this user is root.

   - Enter the Enterprise Administration Server username and password to use for authentication. You are asked to enter your password twice.

This user is not a UNIX user, but a username and password in the Directory Server. You must make sure that the Enterprise Administration Server user exists in the Directory Server, and that it has access permissions to the Directory Server to perform user and group management tasks.

- Type the Enterprise Administration Server port number or accept the default of 8888.

- Make sure you remember the port number for the Enterprise Administration Server.

- For the most flexibility, choose a port number above 1024. If you want to use a port number lower than 1024 for your Enterprise Administration Server, you must be logged on as root to start the server.

- This Enterprise Administration Server is not the same as the Netscape Console.

- Type the port number of Enterprise Server. The default is 80. This port should be different than the Enterprise Administration Server port. The suggested range of ports (1024 to 65535) only applies if you are not installing as root.

  If you use a port other than the default port (port 80), the URL used to gain access to your home page will change. For example, if your computer is called www.mozilla.com and you choose port 9753, your server's URL will be http://www.mozilla.com:9753/.

- Specify whether you are using an LDAP-based directory server (enter **Yes** or **No**).

  You must use an LDAP directory server if you want to use user and group functionality in Enterprise Server.

- If you use an LDAP directory server, enter the following:

  The LDAP URL in the format ldap://hotname:port/base DN

  (For example, ldap://mozilla.com:389/o=airius.com)

  The bind DN (e.g., cn=Directory Manager) and the directory server password

- Type the root directory where your server's content files will reside. The default is server_root/docs.

- If you want to use your own JDK, enter Yes.

  If you are installing Java support, you must have a JRE or a JDK. You can install the supplied JRE component or use a JDK already installed on your system. For more information, see *Installing a Java Developer's Kit*.

- If you are installing Java support, and you either did not choose to install the provided JRE, or you chose to use your own JDK, enter the absolute path to the directory where you installed the JDK on your system.

  The JDK must already exist in the specified directory.

- After the installation program extracts and installs the Enterprise Server components, press Enter.

- Go to the https-admserv directory under your server root directory and start the Enterprise Administration Server by typing ./start. You can also type ./startconsole in the server root directory.

If you have a Netscape console installed, start console starts the console. If you do not have a console installed, it starts Enterprise Administration Server and launches a browser to the Enterprise Administration Server administration pages.

- To configure your Enterprise Server, use the Enterprise Administration URL:

  http://server_name:administration_port

  (For example: http://mozilla:8888)

- Enter your administration username and password.

You can now configure your Enterprise Server. For more information, see the *Netscape Enterprise Server Administrator's Guide*.

Express Installation**:**

Express installation is for users who have little experience or are evaluating the product. It makes assumptions about such things as port number and which components to install.

The following table lists the assumptions made by the Express installation. If you would like to use different installation settings, use Typical or Custom installation.

Express installation settings:

| Installation Setting | Value |
|---|---|
| Administration port | 8888 |
| Administration URL | http://machine name:administration port |
| HTTP port number | 80 |
| Document root | server_root/doc |
| UNIX user to run server | Root |
| LDAP users and groups | Not using |
| JDK | None |
| JRE | Default JRE shipped with Enterprise Server |

After installing, you can configure the server to use LDAP users and groups, and to use a JDK. For more information on how to make these changes, see the *Netscape Enterprise Server Administrator's Guide*.

WARNING: You must log in as root to use express installation.

To run setup, follow these steps:

1. If you are not in the directory already, change to the directory where you unpacked the file.

2. Type **./**setup to start the server installation

   A welcome screen appears.

3. Press Enter to continue with the installation.

4. Choose whether you accept the software license agreement by entering Yes to accept, or No to decline.

5. Choose Express and press Enter.

6. Type a server root directory or accept the default (/usr/netscape/server4).

   This directory is where the server files and directory structure will be installed.

   You need to run the server as a user that has *write* access to this directory (e.g., the directory owner).

   If you already have other Netscape 4.x servers installed, and you want to register Enterprise Server with Netscape Console, install all 4.x servers into the same directory.

7. Choose Netscape Enterprise Server.

   Press Enter.

8. Enter the UNIX user and group names to use when running the default instance of Enterprise Server.

   For Express installation, this user must be root. The group must be the group to which root belongs.

9. Enter the UNIX username to use when running the Enterprise Administration Server. In most cases, this user is root.

10. Enter the Enterprise Administration Server username and password to use for authentication. You are asked to enter your password twice.

    This user is not a UNIX user, but a username and password set up in the Netscape environment.

11. After the installation program extracts and installs the Enterprise Server components, press Enter.

12. Go to the https-adminserv directory under your server root directory and start Enterprise Administration Server by typing ./start. You can also type ./startconsole in the server root directory.

13. To configure your Enterprise Server, use the Enterprise Administration URL:

    http://server_name:administration_port

    (For example: http://mozilla:8888)

    Enter your administration username and password.

You can now configure your Enterprise Server. For more information, see the *Netscape Enterprise Server Administrator's Guide*.

Installing Enterprise Server for Windows NT:

The following sections describe the installation of the Netscape Enterprise Server.

Any errors that occur when the server starts are logged in the Event Viewer. Once started, the server logs errors to the normal error log file.

***WARNING:*** *Please keep the following warnings in mind:*

You must install your Netscape Version 4.x servers in a separate server root directory from the server root that contains your 3.x servers.

If you have beta versions of Netscape 4.x servers installed, uninstall them before installing the final version.

Because of DLL conflicts, if you install Enterprise Server 4.x and Enterprise Server 3.6 on the same machine, Netscape recommends that you uninstall Enterprise Server 3.6 after you migrate it.

Do not install Enterprise Server on an NFS-mounted drive due to potential security and file locking restrictions on remote partitions.

Unpacking the Files:

To get the Netscape Enterprise Server files and unpack them, follow these steps:

Installing from a CD-ROM:

1. Put the CD-ROM in the drive. Click the icon representing your CD-ROM drive.

2. Double-click the "ntx86" folder.

3. Double-click the "entprise" folder.

   The setup.exe file is inside this folder.

Installing a Downloaded Server:

1. Download the file e41diuSP8.exe or later (U.S. domestic) and save it in a temporary directory.

2. Unzip the file to extract the files.

   The setup.exe file is inside this folder.

Running setup.exe:

Run the setup.exe program to install Enterprise Server.

1. Double-click the setup.exe file.

2. The Welcome screen appears. After reading the Welcome screen, click "Next".

3. The Software License Agreement appears. Click "Yes" to accept the license.

4. The Select Installation Type screen appears. Choose the kind of installation you want:

   Express, Typical, or Custom

   Express installation is for users who have little experience or are evaluating the product. It makes assumptions about such things as port number and which components to install.

This option is not recommended if you are using your server in a production environment.

Typical and Custom installation give you more options, and for this release do the same thing.

If you chose Express installation, the following settings are set automatically for you. If you want to have different values for these settings, choose Typical or Custom installation.

Express installation settings:

| Installation Setting | Value |
|---|---|
| Administration port | 8888 |
| Administration URL | http://machine name:administration port |
| HTTP port number | 80 |
| Document root | server_root/doc |
| LDAP users and groups | Not using |
| JDK | None |
| JRE | Default JRE shipped with Enterprise Server |

Click "Next."

5. The Choose Installation Directory dialog box appears. The default location for the server files is C:\Netscape\Server4, where C: is the letter of the drive on which you are installing the server. Click "Next."

   If you want to install Enterprise Server in a non-default location, use Browse to navigate to the folder you want.

   If you have other Netscape 4.x servers installed, and you want to register Enterprise Server with Netscape Console, install all 4.x servers into the same directory.

***WARNING:*** *You cannot install Enterprise Server on the same machine as Directory Server.*

6. The Select Products dialog box appears. Select Enterprise Server from the list.

***NOTE:*** Because the administration server is now an Enterprise Server instance, it is installed automatically when you install Enterprise Server, and is not a choice on this product list.

   By default, the installer installs a default set of components. If you want to change the components you install, click "Change" and continue to Step 7.
   If you did not change components, continue to Step 8.

7. The Select Sub-Components dialog box appears. You can select a component for installation by checking the box next to it. You can decline to install it by unchecking the box. If you later decide you want to install components you did not check, you can run the installer again

and choose to install only the missing components. However, you cannot uninstall separate components once they are installed.

By default, all components except WAI are installed.

Netscape Enterprise Server Core. Install this component to install Enterprise Administration Server and the first instance of Enterprise Server.

***WARNING:*** *You must install the Netscape Enterprise Server Core component the first time you install Enterprise Server. If you install additional components later, you are not required to reinstall the Core component.*

Java Runtime Environment:

If you are installing Java and servlets support, you must have a JRE or a JDK. You can install the supplied JRE component or use a JDK already installed on your system. If you want to use your own JDK, enter "Yes". For more information, see *Installing a Java Developer's Kit*.

Java and Servlets:

Install this component if you are planning to use Java servlets.

You need a JRE or JDK to use Enterprise Server's Java support. If you do not have one installed on your system, you should install the JRE provided with Enterprise Server (see the previous component) or install a JDK.

Server Side JavaScript Database Connectors:

Install this component if you are planning to use server-side JavaScript database connectivity (LiveWire). If you plan to use this feature, you also need to install Java and servlets (see the previous component) as well. You also need to install the JRE or supply a JDK.

Web Publishing:

Install this component if you are planning to use the Web Publishing, Netshare, or search features.

WAI:

Install this component if you are planning to run Web Application or Services Interface programs. If you are using WAI, you must also have an ORB. You will have to provide the installation program with the path to the ORB on your system later in the installation process.

SNMP:

Install this component if you are planning to use SNMP.

8. The Enterprise Administration Server Authentication dialog box appears. Type the username for Enterprise Administration Server access; the default is **"**admin"**.** Type the administration server access password; type it again for verification. Click "Next."

   If you are using LDAP-based authentication, you must make sure that this user has access permissions to the LDAP server to perform user/group management tasks.

   If you chose Express Installation, skip to Step 14.

9. The Enterprise Administration Server Port Selection dialog box appears. Type the port number the Enterprise Administration Server runs on. This can be any number from 1 to 65535. The URL for administration access is displayed. You might want to make a note of this URL. The default port is 8888.

   If Enterprise Server 3.6 is installed on your system, do not use the same Enterprise Administration Server port number for Enterprise Server 4.0 as you used for Enterprise Server 3.6.

   Click "Next."

10. The Default HTTP Server dialog box appears. A default instance of Enterprise Server is installed automatically.

    Type a port number for the default instance of your Enterprise Server. This port should be different than the Enterprise Administration Server port. If you use a port other than the default port (port 80), the URL used to gain access to your home page will change. For example, if your computer is called **www.mozilla.com** and you choose port 9753, your server's URL will be http://www.mozilla.com:9753.

    Type the path for the default server's primary document directory where the server's content files will be stored. The default is C:\Netscape\Server4\docs. Click "Browse" to navigate your file system.

11. The Using LDAP for Users and Group Administration dialog box appears. If you want to use LDAP, click the checkbox and specify the LDAP URL in the format **ldap://hostname:port/base DN** (e.g., **ldap://mozilla.com:389/o=airius.com**).

    You must also enter the bind DN (e.g., cn=Directory Manager) and the directory server password.

    If you do not want to use LDAP, leave the checkbox unchecked.

    Click "Next."

12. If you are installing Java and servlets, the JRE Selection dialog box appears. If you are using Java on your Enterprise Server, you must have a Java Runtime Environment (JRE) or a Java Development Kit (JDK). You can use the JRE included with Enterprise Server (for more information see Step 7), or you can use a custom JDK that already resides on your system.

    To use your own JDK, click the Use Custom Java Development Kit checkbox and enter the path to the folder where you installed the JDK on your system.

13. The Configuration Summary dialog box appears. This dialog box contains information about the settings for your Netscape Enterprise Server and Enterprise Administration Server. It also contains a list of the Enterprise Server components you selected for installation. This dialog box gives you the opportunity to review your settings before the installation is complete. If they are correct, click "Next."

    The server files are installed. The Setup Complete dialog box appears. You must reboot your computer to complete the installation before you can use the server.

***NOTE:*** You should not cancel the installation process while the files are being copied. If you do, you will have a partial installation you need to clean up. If an uninstaller for Enterprise Server exists, use it to uninstall the portion of Enterprise Server you installed. If the uninstaller does not exist, manually delete all the files that are in the server root.

14. Click "Finish."

15. To configure your Enterprise Server, use the Enterprise Administration URL:

http://server_name:administration_port

(For example: http://mozilla:8888)

Enter your administration username and password to administer the server.

Of special note is access to the Netscape Administration Server. This access can be restricted based on the user. The administration server uses a list of users in the Administrators group (the group set up for distributed administration) to determine access rights for the user requesting a resource. The list of users is stored either in a database on the server computer, or in a Lightweight Directory Access Protocol (LDAP) server such as Netscape Directory Server. Before access controls are set, ensure that the database has both users and the Administrators group in it.

Access can be allowed or denied to everyone in the Administrators group, or specific people can be allowed or denied by using wildcard patterns or lists of users. To configure access control with users and groups, follow the general directions for restricting access. A form appears in the bottom frame when the Users/Groups field is clicked. The following list describes the options in the form.

| ***TYPE*** | ***DESCRIPTION*** |
|---|---|
| Anyone (No Authentication) | Is the default, and means anyone in the Administrators group that is specified with distributed administration can access the form. |
| Authenticated people | Only allows you to restrict users within the Administrators group. |
| All in the authentication database | Matches any user who has an entry in the Administrators group in the database. |
| Only the following people | Allows you to specify certain users to match. |
| Prompt for authentication | Allows you to specify message text that appears in the authentication window. |

The following process is recommended for Netscape web server environments (a remote LDAP server, which has an LDAP database).

The IAO should ensure that distributed administration is enabled, an LDAP directory is established, and an Administrators group is set up in the LDAP directory. Finally, the IAO should store the list of users and groups in an LDAP database.

This page is intentionally left blank.

# APPENDIX C. UNIX CONFIGURATION

## C.1 World Wide Web Services and Protocols

The World Wide Web (WWW) allows users to send and receive information across the Internet. It uses graphical interfaces and HTML links to direct information retrieval and flow. Anonymous FTP and HTTP are used to connect transparently to other systems and to find and display information. Because of the open nature of the application, the WWW presents security concerns.

### C.1.1 UNIX Web Servers

For web servers running on UNIX systems, following are some general security precautions to take:

1. Limit the number of login accounts available on the machine. Delete inactive users.

2. Make sure that people with login privileges choose good passwords. The Crack program will help detect poorly-chosen passwords:

   **ftp://ftp.cert.org/pub/tools/crack/**

3. Turn off unused services. For example, if you do not need to run FTP on the web server host, physically remove the FTP daemon. Do the same for tftp, sendmail, gopher, Network Information Services (NIS) clients, NFS, finger, systat, and anything else that might be unnecessary. Check the **/etc/inetd.conf** file for a list of daemons that may be lurking, and comment out the ones not being used.

4. Remove shells and interpreters that are not absolutely needed. For example, if you do not run any PERL-based CGI programs, remove the PERL interpreter.

5. Check both the system and web logs regularly for suspicious activity. The Tripwire program is helpful for scanning the system logs and sensitive files for break-in attempts:

   **ftp://coast.cs.purdue.edu/pub/COAST/Tripwire/**

6. Make sure that permissions are set correctly on system files to discourage tampering. The Computer Oracle and Password System (COPS) program is useful for this:

   **ftp://ftp.cert.org/pub/tools/cops/**

Be alert to the possibility that a *local* user can accidentally make a change to the web server configuration file or the document tree that opens up a security hole. Set file permissions in the document and server root directories such that only trusted local users can make changes. Many sites create a **WWW** group to which trusted web authors are added. Only members of this group make the document root writable. To increase security further, the server root where vital configuration files are kept is made writable only by the official Web Manager.

### C.1.2 Securing the Server

It is good practice on a web server to restrict the use of the superuser account. Creating other accounts to run the web service, own key configuration files, and modify web site content is the best way to attain this posture. In this case, the following permissions can be set on production servers for the given directories and be checked weekly by ESM (or the appropriate substitute until ESM is implemented). Create a non-privileged (e.g., nobody) user such as **WWW** to run the web instance; webmgr to own these directories; and a non-privileged group (e.g., webnp). These accounts and their use will be part of the server baseline documentation.

**Table C-1. Permission For Production Servers**

| ***REPRESENTATIVE DIRECTORY NAMES*** | ***OWNER*** | ***GROUP*** | ***PERMISSIONS*** |
|---|---|---|---|
| cgi-bin* | webmgr | webnp | 510 |
| config | webmgr | webnp | 550 |
| Htdocs*** | webmgr | webnp | 550 |
| Icons | webmgr | webnp | 550 |
| Logs** | webmgr | webnp | 510 |

***** If more restrictive permissions are used (e.g., **510**), then the cgi programs under the cgi-bin directory must be owned by the *uid* of the user running the web server with permissions of **500**.

****** Files under Logs will be owned by webmgr with *gid* being the non-privileged group (e.g., webnp) allowing *uid* of user running web server having permissions of **460**.

******* Top-level subdirectories under HTDOCS may have the following:

**Table C-2. Web Data Area – No Updates**

| ***HTDOCS*** | ***OWNER*** | ***GROUP*** | ***PERMISSIONS*** |
|---|---|---|---|
| Subdirectories | webmgr | webnp | 510 |
| Files | webmgr | webnp | 440 |

**Table C-3. Web Data Area – Application Account**

*(e.g., webact with gid being non-privileged group [e.g., webnp]
allowing uid of application account and uid of the user running the web server)*

| *HTDOCS* | *OWNER* | *GROUP* | *PERMISSIONS* |
|---|---|---|---|
| Subdirectories | webact | webnp | 750 |
| Files | webact | webnp | 640 |

### C.1.3 File Configuration Parameters

Because the following files (or their equivalent) can affect the operation of the server, they will be protected. Check the server documentation, determine the exact file name for the function, and set the permissions of those files to correspond to the following. In this example, all files will be owned by webmgr with *gid* being the non-privileged group (e.g., webnp) allowing *uid* of user running the web server. They will have permissions of **640** or more restrictive.

**Table C-4. Configuration Sample File Names**

| *SAMPLE FILE NAMES* | *FUNCTION* |
|---|---|
| Access.conf | Controls access to server files |
| Httpd.conf | Configuration file for server |
| Mime.types | Determines mapping of file extension to MIME types |
| Srm.conf | Server Resource Map configuration information |
| Magnus.conf | Configuration file for server |

### C.1.4 CGI Scripts

CGI is a protocol used to create programs for WWW pages. There is a possibility of compromising the security of your WWW browser when using CGI scripts. The problem with CGI scripts is that some inputs can cause the CGI program to crash or behave in an unexpected way. The danger is reduced if the vendor supplied the CGI script.

- The cgi-bin directory will have permissions of **550** or more restrictive.
- All CGI scripts will have permissions of **550** or more restrictive.
- All backup copies of CGI scripts that are automatically generated will be removed from the system.
- No script will be downloadable by anonymous FTP users.
- All CGI scripts will be checked regularly to ensure that unauthorized personnel have not modified them.

This page is intentionally left blank.

# APPENDIX D. MICROSOFT INTERNET INFORMATION SERVER CONFIGURATION DETAILS

## D.1 Security Guidelines for Microsoft Internet Information Server

Additional IIS controls are discussed in the following sections of this appendix. These items are in addition to the *Web Server STIG* and the *NSA Guide to the Secure Configuration and Administration of Microsoft Internet Information Server 5.0 and the NIST Guidelines for Securing Public Web Servers.*

## D.2 Certificate Server Enrollment Pages

The ASP pages that are installed for Certificate Server are by default not secured. To prevent unauthorized access to these, use one of the following methods:

- Remove the pages.

- Set to perform the following:

  Set the following ACLs on the **%systemroot%/certsrv** directory:

  Administrators (Full Control)
  System (Full Control)
  Certificate Issuers (Full Control)

- Add trusted certificate operators to the Certificate Issuers group.

## D.3 IISADMPWD Virtual Directory

This directory allows users to change their password via the web interface. If you are not using this functionality, this directory needs to be removed from the web server. It is located under the Default Web Site. If this functionality is required for your applications, ensure that access to these files is tightly controlled and is monitored via the various logging options within IIS and NT. For complete instructions on this issue, please refer to Microsoft Knowledge Base article Q184619.

## D.4 Unused Script Mappings

IIS is pre-configured to support common file name extensions such as **.asp** and **.shtm**. When IIS receives a request for a file of one of these types, the call is handled by a DLL. If unused mappings remain on the web server, malicious files may be placed on the web server and processed. Script mappings have proven to be the source of several IIS vulnerabilities.

To remove these mappings, open the Management Console and perform the following:

1. Right-click the web server
2. Properties
3. Master Properties
4. WWW Edit
5. Home Directory
6. Configuration

From here you can remove the entries you do not use. For example, if you are not using web-based password reset, you can remove the **.htr** entry. Other entries may be **.idc** for Internet Database Connectors and **.shtm**, **.stm**, or **.shtml** for Server-Side Includes.

The IIS Lockdown tool can also be used to mitigate this finding. This tool is by default configured to map unused script extensions to the 404.dll. This will also eliminate the need to re-delete a mapping if an installation of said mapping occurs.

## D.5 RDS Support

When incorrectly configured, Remote Data Services can make a server vulnerable to denial of service and arbitrary code execution attacks. This capability needs to be removed or its access restricted using ACLs.

Refer to Microsoft Security Bulletins MS98-004 and MS99-025 and Microsoft Knowledge Base article Q184375 for further information.

If this capability is turned on, you can monitor for signs of attack by looking for the following in the IIS log files:

**2000-01-20 02:12:13 – POST /msadc/msadc.dll**

## D.6 COM Components

Some COM components are not required for most applications and should be removed if possible. Most notably, consider disabling the File System Object component; however, this will also remove the Dictionary object. Be aware that some programs may require components you are disabling, so it is highly recommended that this be tested completely before implementing on your production web servers.

The following will disable the File System Object:

**regsvr32 scrrun.dll /u**

## D.7 Calling Command Shell with #exec

The command can be used to call arbitrary commands at the web server from within an HTML page. IIS disables this by default. You can double-check this by making sure the following is set to zero (0) or is missing:

**Table D-1. Commands**

| | |
|---|---|
| **Hive** | HKEY_LOCAL_MACHINE\SYSTEM |
| **Key** | CurrentControlSet\Services\W3SVC\Parameters |
| **Name** | SSIEnableCmdDirective |
| **Type** | REG_DWORD |
| **Value** | 0 |

## D.8 Content-Location Header

When using static HTML pages, a Content-Location header is added to the response. By default, Internet Information Server (IIS) 4.0 Content-Location references the IP address of the server rather than the FQDN or Hostname.

This header may expose internal IP addresses that are usually hidden or masked behind a Network Address Translation (NAT) firewall or proxy server.

There is a value that can be modified in the IIS metabase to change the default behavior from exposing IP addresses to sending the FQDN instead.

The value that needs to be set is the **w3svc/UseHostName**, and it needs to be set to **True**.

The other option to prevent this from occurring is to use Active Server Pages instead of static HTML pages and create a custom header that sends back a specific Content-Location. For complete instructions on this issue, please refer to Microsoft Knowledge Base article Q218180.

## D.9 Sample Files

By default IIS installs the “adminsamples” files in c:\winnt\system32\inetsrv. These files should be removed, and then open the IIS manager and remove the adminsamples virtual directory from all web instances. All sample files from all installed applications or plug-ins will be removed from a web server.

## D.10 The Metabase File IIS 4.x and 5.x

The Metabase is stored in a special format disk file, by default named **Metabase.bin** in the \WINNT\system32\inetsrv directory. This is the Default installation directory for IIS. The Metabase loads from disk when IIS starts, is stored to disk when IIS shuts down, and is saved periodically while IIS is running. It is important to protect this file from unauthorized use because sensitive data is stored within the file. Store the **Metabase.bin** file on an NTFS partition and use Windows NT security to protect it. When IIS 4.0 is installed, the Administrators group and System are given Full Control to the **Metabase.bin** file. There is no need to modify this setting.

To hide this file from unauthorized users, move or rename the file. To relocate or rename the Metabase file, stop IIS, move or rename the file, and modify the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InetMgr\Parameters. Add a new REG_SZ value to this key named MetadataFile to specify the new complete path of the Metabase file, including the drive letter and file name.

Restrict the permissions on the **mdutil.exe** file to Full Control for the SA and Web Manager only.

## D.11 Document Directory

Move this directory to a path that is not the default and that is on drive or partition separate from the web server system software and network operating system resources.

# APPENDIX E. SERVER CERTIFICATES

## E.1 User Certificates

To obtain a user PKI certificate, please contact your local Limited Registration Authority (LRA) for specific information. If you are unsure of whom your LRA is, contact your Security Manager (SM) or IAM.

## E.2 Server Certificates

To obtain a server certificate, connect to the following URL and follow the instructions on the site for requesting a Server PKI Certificate:

**http://DODpki.c3pki.chamb.disa.mil**

Before requesting a server certificate from the DOD PKI, it is important that the submitted request meets specific guidelines, or it will be rejected. When entering the distinguished name information for your server, adhere to the following guidelines. These are the fields that you need to enter and the values that go with them:

| | |
|---|---|
| Key Size: | **1024 *only!!!*** |
| Common Name: | The fully qualified hostname of your server (e.g., www.adu.acom.mil) |
| Organization: | **U.S. Government** |
| Organizational Unit: | **'C/S/A', ou=PKI, ou=DOD** (where 'C/S/A' is the C/S/A for which the server is used) |
| Locality: | Leave blank |
| State: | Leave blank |
| Country: | **US** |

After completing the request for a server certificate, the Web Manager or SA must go to http://pluto.ncr.disa.mil:88/pki and complete the form on this page using the certificate request number obtained from a successfully submitted request at http://DODpki.c3pki.chamb.disa.mil.

This page is intentionally left blank.

# APPENDIX F. IBM HTTP SERVER (IHS) WEBSPHERE

## Mid-Tier
## Security Recommendations

## F.1 Introduction

The IBM HTTP Server (IHS) 1.3.6.2 is freely available. The IBM HTTP Server is packaged so that only the required functions will be installed. However, some functions depend on other functions to be installed. The Solaris installation tools do not automatically install dependencies. To ensure that the necessary functions are installed, install any dependencies first or a warning will be received during the install.

To install the HTTP AdminServer, do the following:

1. Install the IBM HTTP Server documentation.
2. Install the IBM HTTP Server Documentation Base.
3. Install the AdminServer messages.
4. Install the HTTP AdminServer.

To install the IBM LDAP Module, do the following:

1. Install the IBM LDAP Client first.
2. Install the IBM LDAP Module.

To install the IBM SSL Module, do the following:

1. Install the IBM SSL Module Common.
2. Install the level of encryption required.

## F.2 Specific Practices

### F.2.1 Installation

The directory where IHS is installed in an NT environment is typically **c:\Program Files\IBM HTTP Server**.

### F.2.2 Administration Server

Before you start the Administration Server to administer the configuration data for the IBM HTTP Server, you must perform some preliminary administrative tasks. These tasks may be performed by executing the **setupadm** script. The script will prompt you for all the necessary input.

The basic intent of the Administration Server tasks is to allow the Administration Server *read/write/execute* access to the necessary configuration files and one executable file. The Administration Server should obtain *read/write* access through a unique Userid and Group, which must be created. The User and Group directives of the Administration Server's configuration file should be changed to the unique Userid and Group. The **Group access permissions** for the Administration Server's configuration files should be changed to allow *read/write* Group access. In addition, there is a utility program that should have **Group execute permissions** and **Set Userid Root permissions**. This executable must run as Root in order to request restarts for the IBM HTTP Server and the Administration Server.

The Administration Server is installed with Authentication enabled for the directory containing all configuration forms. This means that after installation, the Administration Server will not service a page without a userid and password. This is done to protect the IBM HTTP Server Configuration file from unauthorized access immediately after successful installation of the IBM HTTP Server and the Administration Server. At installation, the password file (**admin.passwd**) is empty, therefore until a userid and password is supplied in the Administration Server Password file (**admin.passwd**), you will not have access to the IBM HTTP Server Configuration pages through the Administration Server.

1. For NT, type **htpasswd -m conf\admin.passwd <userid>** (from directory **/Program Files/IBM HTTP Server**).

2. For AIX, Solaris, and Linux, type **./htpasswd -m /conf/admin.passwd <userid>** from the following directory structures:

   AIX — **/usr/HTTPServer/bin type**
   Solaris — **/opt/IBMHTTPD/bin**
   Linux — **/opt/IBMHTTPServer/bin**

3. You will be prompted for a password and then prompted to retype the same password for verification.

This will be the userid and password that will allow access to the Administration Server Configuration GUI. This userid should be unique for access to the Administration Server. The Administration Server directive **User** should **not** be the same userid for access to the Administration Server.

### F.2.3 SSL Initialization

If you are using a stash file to automate SSL initialization, you must ensure that the key database password in the stash file is protected. The key database password is altered in the stash file so that it cannot be recognized by a casual observer, but it is not encrypted. Do not allow unauthorized persons access to either the stash file or the key database file. As with all Web server resources, managing proper file permissions and protections is vital to the security of the system.

### F.2.4 Production Environments

Keep IHS separate from the WebSphere Application Server (WAS) and any legacy databases. Ideally, separate servers will be used for each.

### F.2.5 Notes on Solaris Installs

The existing configuration file is preserved as **httpd.conf,** and the configuration file for the new version is saved as **httpd.default**.

1. Uninstall any previous versions of the server:

   a. Log on as root.

   b. Stop the server by changing to **/opt/IBMHTTPD/bin**.

   c. Type **./apachectl/stop**.

   d. At the command prompt, type **admintool** and do the following:

      1) Select **Browse > Software**.
      2) Select **IBM HTTP Server**, **IBM HTTP SSL module**, and **GSK**.
      3) Select **Edit > Delete**.
      4) Answer **Yes** to the confirmation messages.

2. If you plan to use the Key Management (IKEYMAN) utility to create server certificates for SSL, install the Java Development Kit (JDK), 1.1.6 or higher.

3. Log on as root.

4. Extract all downloaded files to a temporary directory.

5. At a command prompt, type **admintool** and do the following:

   a. Select **Browse > Software**.
   b. Select **All** software.
   c. Select **Edit > Add**.
   d. Select the options from the list you would like to install.

The path **/opt/IBMHTTPD** is used as the base directory. See the *Web Server Security Checklist* relating to common web checks and Apache web checks for further details.

**IBM HTTP Server AfpaCache DoS Vulnerability**

The IBM HTTP Server contains the AfpaCache directive, which turns the Fast Response Cache Accelerator function on or off.

In February 2001 it was reported that the IBM HTTP Server is subject to a denial of service attack. Requesting multiple malformed HTTP GET requests will cause the consumption of kernel memory and eventually lead to a denial of service. This condition is due to the AfpaCache module not releasing allocated memory after "Bad Request" HTTP requests. A restart of the service is required in order to gain (regain) normal functionality. It should be noted that WebSphere is built based on the IBM HTTP Server and is subject to this vulnerability.

**Recommended Precaution:**

Check the permissions on the **httpd.conf** file in which the AfpaCache directive resides. Permissions of **510** are recommended.

**References:**

http://www-4.ibm.com/software/webservers/httpservers
http://www.websphereadvisor.com/

# APPENDIX G. IBM HTTP SERVER (IHS) FOR OS/390

## Security Recommendations

## G.1 Introduction

The IBM HTTP Server (IHS) for OS/390 is a full-featured web server that runs on OS/390. It provides a standards-compliant environment that supports static and dynamic content as well as interfaces to application servers. IHS for OS/390 offers several advanced features including the following:

- Client authentication using resident Access Control Program (ACP) facilities
- Thread-level security using the client's authorization level
- Support for CGI programs written in compiled and interpreted languages
- Support for programs coded to utilize the Go Webserver Application Programming Interface (GWAPI)
- Support for Java programs written as CGI programs and, with an application server, as servlets
- Support for Secure Sockets Layer (SSL) sessions
- Access to data in MVS data sets as well as HFS files
- A dynamic data cache using the Fast Response Cache Accelerator
- Logging and reporting facilities.

The current version of IBM HTTP Server for OS/390 is bundled with the OS/390 operating system, but it is not a new product. IHS for OS/390 has evolved over several years and has been a standalone product and a bundled component of other products. It was based on the original CERN web server and has been identified by various names:

- In early 1996 the product was known as the Internet Connection Server (ICS).
- By late 1996 the product had become the Internet Connection Secure Server (ICSS). Versions 2.1 and 2.2 were released.
- By late 1997 the product was renamed the Lotus Domino Go Webserver (DGW). The product was released as Versions 4.6, 4.6.1, and 5.0.
- In 1999, the name IBM HTTP Server for OS/390 was designated and the product was released as Version 5.0.

The purpose of this appendix is to document security issues that are specific to the IHS for OS/390 product. Additionally, because OS/390 and the requisite hardware provide a significantly different security environment, some adaptations of general *Web Server STIG* security requirements are necessary.

Information in this appendix is written to address Versions 5.2 and 5.3 of the IBM HTTP Server for OS/390. These versions were bundled with OS/390 Releases 2.8 and 2.10 respectively. Information that is version specific is indicated as such. This appendix does not cover using IHS for OS/390 as a proxy server, a function provided by the IBM Web Traffic Express component. Use of the OS/390 Workload Management (WLM) feature is also not covered. Sites using these features should consult the IBM documentation.

Descriptions of software and parameters in this appendix are based on the applicable versions of IBM's *OS/390 HTTP Server Planning, Installing, and Using* document. Sites should refer to the IBM documents for precise details. In addition, the most current documentation is available from IBM's web site at:

http://www.ibm.com/software/webservers/httpservers/library.html#os390

It should be noted that z/OS, the follow-on to OS/390, is now generally available to licensed IBM customers and is being deployed within DISA. Version 5.3 of IBM HTTP Server is bundled with z/OS and is labeled as IBM HTTP Server for z/OS. As indicated by the version numbering, this product should work substantially the same as the OS/390 version. However, careful examination of the IBM documentation is highly recommended.

## G.2 Web Server Principle Requirements

DISA's *Web Server STIG* was written to address security requirements for web servers. The original target for the document was servers on mid-tier UNIX servers and Windows servers. Those platforms have considerably different security architectures than OS/390. In some cases those architectures provide less robust security implementations and strict measures are required to achieve an acceptable level of security.

The OS/390 operating system benefits from a long legacy of secure design principles. Basic architectural features such as address space isolation, storage protect keys, and the System Authorization Facility (SAF) help to ensure that OS/390 provides a secure computing platform. These features also make OS/390 a somewhat unique environment. This uniqueness is amplified by characteristics such as the distinct IBM instruction set and data encoding in EBCDIC rather than the more common ASCII. The result of this distinctiveness is that OS/390 applications, even those that are functionally equivalent to applications on other platforms, have different security requirements than applications running on other platforms. It also results in requirements to perform ports of common UNIX applications before they are able to run on OS/390.

While the basic principles of the requirements in the *Web Server STIG* need to be applied to the IBM HTTP Server for OS/390, there are some requirements that require adaptation. DISA Field Security Operations is currently working to provide this adaptation. Until such time as this is complete in all areas, the following guidelines should be followed for OS/390 with respect to requirements stated in the *Web Server STIG*:

- Where the STIG requirements specify that programs or utilities will be removed, it is considered acceptable if those programs or utilities are secured by the ACP. In some cases, this means that users accessing the web server will not be able to access the programs while users authorized for other OS/390 interactive applications (e.g., TSO) may have access.

- Where the STIG requirements specify that userids for web users will be unique or will have access only to web documents, this is not to be interpreted that users would require different IDs for web server access than for access to other OS/390 applications for which they are authorized.

- Where the STIG requirements specify UNIX permission bit assignments that are also discussed here, the requirements in this appendix will be applied.

- Where the STIG requirements specify that development items, backup items, or sample items (documents or programs) will not exist on a production server, this is to be interpreted that items will not be accessible to a production web server.

- Where the STIG requirements specify restrictions on the use of FTP, these restrictions will not apply to platforms running **Private** web servers.

It is anticipated that these guidelines will be refined as more experience is gained with DISA's use of IHS for OS/390. As always the sites are encouraged to use the feedback mechanisms specified in the STIG to provide comments.

### G.2.1 Public and Private Web Server Requirements

As noted in the *Web Server STIG*, access to a **Public** web server is available to all Internet users. In contrast, access to a **Private** web server is restricted by network controls, userids and passwords, or digital certificates. This means that users of a **Public** web server are gaining access to the information system without identification or authentication. It is clear that such an environment requires strict controls.

The DISA's *STIG on Enclave Security* specifies, "A DMZ will be established within the Enclave Security Architecture to host any publicly accessible systems (e.g., ECEDI, public web servers, mail servers, external Domain Name Service [DNS], X.500 directories, etc.). The approved architecture is to build the DMZ on a separate branch (network interface) of the Enclave Perimeter firewall."

Implementations of **Public** web servers on OS/390 hosts are required to conform to the DMZ architecture specifications in the DISA *STIG on Enclave Security*. In addition, steps must be taken to ensure that data on other site systems is not accessible.

- *The OS/390 host of a **Public** web server will be configured as follows:*

  - *Be isolated in a distinct logical partition (LPAR) or on a dedicated machine on which only public information is processed*

  - *Be network connected only to the site's DMZ*

  - *Have no storage peripherals (i.e., DASD) that are shared with other LPARs or machines*

  - *Have no network-based resource sharing software such as OS/390 Network File System (OS/390 NFS) or OS/390 Distributed File Service (OS/390 DFS) enabled*

**Private** web servers are expected to enable appropriate controls to prevent access by users not specifically authorized. In addition, uncoordinated access to server data by search engines is not considered necessary. The unofficial standard honored by some search engine software is a file named **robots.txt** that contains search restrictions.

- *A **robots.txt** file will be created in the directory specified by the **ServerRoot** directive of all **Private** web servers.*

## G.3 IBM HTTP Server for OS/390 Platform Requirements

Configuring security for the IBM HTTP Server for OS/390 involves a number of tasks. Although some of these are similar to those required for traditional interactive OS/390 applications, many are related to the OS/390 UNIX environment and some are the result of the unique services provided by a web server.

This section provides detailed discussions of the security considerations and the resulting requirements that are specific to IHS for OS/390. The following topics are discussed:

- Server and administrator IDs and startup configuration
- Server data sets
- Server HFS objects
- Client identification and authentication
- Access control directives and protection setups
- Access Control List (ACL) files
- Resource mapping directives
- Miscellaneous directives
- Secure Sockets Layer (SSL) configuration
- Application interfaces
- Environment variables
- MVSDS DLL service
- Fast Response Cache Accelerator (FRCA)

- Access and error logging
- IBM Communications Server considerations
- Open Source software.

### G.3.1 Web Server and Administrator Identification and Startup Configuration

The IBM HTTP Server for OS/390 is similar to other interactive OS/390 applications in terms of the security identification for the address space in which it runs. An ID, known as the web server ID, is used when the server is started and in access control decisions for tasks that belong to the server.

Unlike some other applications, IHS for OS/390 creates individual threads for user-initiated requests and those threads run under the security context of an ID that is provided by the client or assigned by the server. In this way, access from clients to individual host resources is controlled according to the rules that apply to the client. The ID that is assigned to the thread is defined as an access control ID and is distinct from the web server ID. The use of access control IDs is discussed in *Section G.3.4, Client Identification and Authentication*.

Administering a web server involves tasks that are common in OS/390 environments. A web server administrator changes configuration statements, moves files to appropriate locations, and performs software maintenance tasks. In the OS/390 environment it is not necessary to establish a unique account dedicated for web server administration and shared between users. Designating certain users as web server administrators and supplementing their accounts with required special privileges is a better solution because it provides individual accountability that is missing from a shared administrator account.

A restriction on how IDs are defined is specified by policy in the *Chairman of the Joint Chiefs of Staff Manual (CJCSM) 6510.01, "Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)*." That document states, "All factory-set, default, or standard userids and passwords will be removed or changed prior to the system going operational." This restriction is implemented in the requirements documented in this appendix.

#### G.3.1.1 Web Server Identification

The IBM HTTP Server for OS/390 normally runs as a started task in the OS/390 environment. This implementation allows the operations staff to manage a web server in the same way as other interactive application servers on OS/390. Multiple instances of the web server can be run simultaneously, as multiple started tasks, to support different applications. Although a web server could be started through the OS/390 UNIX shell environment, the issues of the web server ID, proper parameter specification, and operations management make this an undesirable choice.

Each instance of a started task for a web server can execute under a different ID. Because the ID can be defined without powerful privileges such as UNIX UID(0), access controls can be established to provide appropriate security environments for different applications. By defining a local OS/390 UNIX group for servers, access can more easily be controlled for the HFS files associated with that server.

- *Each instance of the web server will be assigned a unique userid.*

- *One or more local OS/390 UNIX groups will be defined for web servers.*

- *The userid assigned to a web server will be defined with the following characteristics:*

  - *Not named WEBSRV*

  - *Defined to run as a started task*

  - *Defined with the following OS/390 UNIX attributes:*

    - *Non-zero UID*
    - *Home directory **'/usr/lpp/internet'***
    - *Shell program **/bin/sh***
    - *Defined as a member of a local OS/390 UNIX group for web servers and the OS/390 UNIX **IMWEB** group*

  - *Has **read** access to the **BPX.DAEMON** and **BPX.SMF** SAF resources*

  - *Has **update** access to the **BPX.SERVER** SAF resource*

The sites should consider a web server ID naming convention that reflects the application or workload being supported. A convention similar to what is used for individual CICS region IDs could be used.

To define a web server ID with a non-zero UID, the site needs to address the following issues:

- Initial versions of IHS for OS/390 were designed to execute under UID(0). IBM provides maintenance under APAR PQ41777 to allow web server IDs with non-zero UIDs.

- HFS directories and files that a web server needs to access must have appropriate permission bit settings. While many directories and files are discussed in this document, the web administrator will have to ensure that specific objects have the required access settings.

In addition to access to SAF resources already mentioned, web server IDs may require additional access depending on configured options. Briefly these resources are as follows:

- **Surrogate Access Control IDs** – When a surrogate ID is used as an access control ID, the web server ID must have *read* access to the **SURROGAT** class SAF resource **BPX.SRV.*user***, where ***user*** is the surrogate ID, for each surrogate ID. The use of surrogate IDs is discussed in *Section G.3.4, Client Identification and Authentication*.

- **Digital Certificates** – If the web server is configured to support Secure Sockets Layer (SSL) connections and the certificates are stored in the ACP, the web server ID must have *read* access to selected resources in the **FACILITY** SAF class. The use of SSL connections is discussed in *Section G.3.9, Secure Sockets Layer (SSL) Configuration*.

- **Cryptographic Hardware** – If the web server is configured to support SSL connections and hardware encryption has been enabled on the host, the web server ID must have read access to selected resources in the **CSFSERV** SAF class. The use of hardware encryption is discussed in *Section G.3.9, Secure Sockets Layer (SSL) Configuration*.

### G.3.1.2 Web Server Administrator Identification

As noted earlier, it is not necessary or desirable to establish unique web server administrator IDs. By adding required privileges to the IDs of users designated as web administrators, individual accountability is maintained. By defining a local OS/390 UNIX group for web administrators, access can more easily be controlled for the HFS files that administrators need to change.

- *One or more local OS/390 UNIX groups will be defined for web administrators.*

- *IDs assigned to users designated as web server administrators will be defined with the following characteristics:*

  - *Not named WEBADM*

  - *Defined with the following OS/390 UNIX attributes:*

    *Defined as a member of a local OS/390 UNIX group for web administrators, a local OS/390 UNIX group for web servers, and the OS/390 UNIX **IMWEB** group*

Web administrators need special privileges to perform the following duties:

- Moving and changing the ownership of HFS files that contain content and software used by the web server
- Altering the program-controlled extended attribute bit for programs in HFS files
- Depending on local operations procedures, starting, stopping, and restarting web servers

The recommended approach to assigning special privileges is as follows:

- Allow the web administrator *read* access to the **BPX.SUPERUSER** FACILITY class SAF resource. This allows the administrator to have superuser status when required. A less desirable alternative is to define the administrator's ID with UID(0).

- Allow the web administrator access to certain SAF resources in the UNIXPRIV class so that some tasks can be done without switching to superuser status. The following access levels are needed:

  - *Control* access to **SUPERUSER.FILESYS**

  - *Read* access to **SUPERUSER.FILESYS.CHOWN**, **SUPERUSER.PROCESS.GETPSENT**, and **SUPERUSER.PROCESS.KILL**

- Allow the web administrator read access to the **BPX.FILEATTR.PROGCTL** and **BPX.FILEATTR.APF** FACILITY class SAF resources. This allows the administrator to assign the program-controlled and authorized program facility attributes to HFS executable files that are loaded into the server's address space.

In accordance with the policy in the DISA *OS/390 STIG* concerning the use of special privileges, ACP logging of the use of the privileges should be enabled.

IBM specifies **WEBADM** with an OS/390 UNIX group **IMWEB** as a sample administrator account and supplies some tools based on this ID and group. In order to use an alternative to **WEBADM**, the sites may need to perform one or more of the following steps:

- Make changes to setup scripts in the /usr/lpp/internet/sbin directory.

- Ensure that the Protection directive for the Configuration and Administration resources in the server's configuration file specifies the IDs of local web administrators. See *Section G.3.5, Access Control Directives and Protection Setups*, for a discussion of the Protect directives.

- Ensure that the directories (and their files) indicated in the Protect directives for the Configuration and Administration resources in the server's configuration file have permission bit settings that allow the local web administrators to use them. See *Section G.3.5, Access Control Directives and Protection Setups*, for a discussion of the Protect directives.

### G.3.1.3 Web Server Startup Configuration

IBM provides multiple methods to start the IBM HTTP Server for OS/390. It can be started by an operator start command, through an automation tool, submitted by a user as a batch job, or as an OS/390 UNIX process by a user in the OS/390 UNIX shell environment.

As noted in *Section G.3.1.1, Web Server Identification*, starting IHS for OS/390 as an MVS started task offers the most advantages. Using this option, a Job Control Language (JCL) procedure (PROC) is written with the necessary JCL statements for the server. IBM supplies a PROC named IMWEBSRV as a model. Creating a unique PROC for each web server simplifies security and facilitates the setup to run multiple servers on the same host.

- *Each web server will be started from a unique JCL PROC.*

There are two HFS files that provide specifications for virtually all of the server's options. The first file is called the configuration file. The second is the environment variables file. While there are default values for the names of these files, uncertainty is reduced and security auditing is enhanced by explicitly specifying the names of the files in the startup JCL.

- *The JCL for each web server will explicitly specify the '**-r**' server startup option with the name of the server's configuration file.*

Using the sample JCL supplied by IBM, the following JCL statement illustrates to how to meet this requirement:

```
//        ICSPARM=' -r /etc/websrv1/httpd.conf'
```

- *The JCL for each web server will explicitly specify the **_CEE_ENVFILE** variable, set to the name of the server's environment variables file or a reference to another DD statement that specifies the file.*

Using the sample JCL supplied by IBM, the following JCL statement illustrates one way to meet this requirement:

```
//    LEPARM='ENVAR("_CEE_ENVFILE=/etc/websrv1/httpd.envvars")'
```

The following JCL statements illustrate an alternative way to meet this requirement:

```
//        LEPARM='ENVAR("_CEE_ENVFILE=DD:HTTPENV")'
…
//HTTPENV   DD PATH='/etc/websrv1/httpd.envvars ',
//                          PATHOPTS=(ORDONLY)
```

Permission bit settings for the configuration file and the environment variables file are discussed in *Section G.3.3, Web Server HFS Objects*.

Additional requirements for the environment variables file are discussed in *Section G.3.11, Environment Variables*.

### G.3.2 Web Server Data Sets

Some of the vendor-supplied components of IHS for OS/390 are stored in data sets as follows:

- Distribution data sets hold the master copy of the product's elements. There is no typical need for general users to access these data sets. The standard naming convention for these data sets is to use the prefix SYS1.IMW.AIMW.

- Target data sets hold the execution copy of the product's elements. General users may need *read* access to some of these data sets. The standard naming convention for these data sets is to use the prefix SYS1.IMW.SIMW.

- ***Update** and **alter** access to product data sets will be restricted to systems programming personnel.*

The use of products that add functionality to the web server can require that a STEPLIB be used in the web server's JCL. Access to data sets specified as a STEPLIB must be controlled so that malicious code is not introduced into the server's address space.

- *Unless specified differently by requirements in the* DISA *OS/390 STIG*, ***update** and **alter** access to data sets specified by the STEPLIB statement in the web server's JCL will be restricted to systems programming personnel and web server administrators.*

On systems where Secure Sockets Layer (SSL) processing is enabled, there may be cases where PKI certificate information is maintained in data sets. These data sets are commonly given a suffix of .ARM, but this is not a requirement of the software.

To help ensure the security of encryption key information, all access to data sets that contain PKI certificate-related data for web servers should be restricted to security administrators and web server administrators.

### G.3.3 Web Server HFS Objects

Almost all of the data and programs used by the IBM HTTP Server for OS/390 are stored in Hierarchical File System (HFS) directories. Protecting the files and directories, collectively called HFS objects, is critical to maintaining security. Setting appropriate UNIX permission and audit bits and owner and group IDs accomplishes this in most cases. Some additional measures are required in other instances.

To define appropriate access controls, it is helpful to categorize the HFS objects that IHS for OS/390 uses. The following subsections discuss these categories:

- Vendor server software directories
- Local server standard directories
- Local server configuration files
- Local server log directories
- Other server directories and files

The following notes apply to the requirements in all these discussions:

- If an owner field indicates ***UID(0) user***, any system ID with a UID(0) specification is acceptable.

- Where an owner field indicates ***websrv1***, the ID of the web server is intended.

- Where a group field indicates ***webadmg1***, the ID of a local web server administration group is intended. **IMWEB** is not a valid local group.

- The site is free to set the permission and audit bit settings to be more restrictive than the documented values.

### G.3.3.1 Vendor Server Software Directories

Many of the IBM components for IHS for OS/390 are installed in a set of HFS directories. The content in these directories would be the same for all web servers. It is not expected that the site will customize these directories.

- *The permission and user audit bits and owner and group settings for the vendor server software directories will be configured according to the settings in the following table:*

**Table G-1. IHS Vendor Server Software Hfs Object Security Settings**

| *IHS VENDOR SERVER SOFTWARE HFS OBJECT SECURITY SETTINGS* | | | | |
|---|---|---|---|---|
| ***DIRECTORY or FILE*** | ***PERMISSION BITS*** | ***USER AUDIT BITS*** | ***OWNER*** | ***GROUP*** |
| /usr/lpp/internet | 755 | fff | *UID(0) user* | IMWEB |
| /usr/lpp/internet/bin | 755 | fff | *UID(0) user* | IMWEB |
| /usr/lpp/internet/sbin | 750 | fff | *UID(0) user* | IMWEB |

### G.3.3.2 Local Server Standard Directories

Many of the IBM components for IHS for OS/390 are installed in a set of HFS directories that should be unique for each web server. While the content of these directories may originally be from IBM, it is expected that the site will customize content in some of these directories.

- *The permission and user audit bits and owner and group settings for the local server standard directories* will *be configured according to the settings in the following table:*

**Table G-2. IHS Local Server Standard Hfs Object Security Settings**

| *IHS LOCAL SERVER STANDARD HFS OBJECT SECURITY SETTINGS* | | | | |
|---|---|---|---|---|
| ***DIRECTORY or FILE*** | ***PERMISSION BITS*** | ***USER AUDIT BITS*** | ***OWNER*** | ***GROUP*** |
| …/*websrv1_root*/ | 555 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/Admin | 550 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/admin-bin | 550 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/cgi-bin | 551 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/fcgi-bin | 550 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/pub | 555 | fff | *websrv1* | *webadmg1* |

The directory ***websrv1_root*** is a site-selected name, but must be unique for each web server.

### G.3.3.3 Local Server Configuration Files

The web server configuration files control most of the options for the web server. These files must be unique for each web server.

- *The permission and user audit bits and owner and group settings for the local server configuration files* will *be configured according to the settings in the following table:*

**Table G-3. IHS Local Server Configuration Hfs Object Security Settings**

| *IHS LOCAL SERVER CONFIGURATION HFS OBJECT SECURITY SETTINGS* | | | | |
|---|---|---|---|---|
| ***DIRECTORY or FILE*** | ***PERMISSION BITS*** | ***USER AUDIT BITS*** | ***OWNER*** | ***GROUP*** |
| /etc/*websrv1*/httpd.conf | 460 | faf | *websrv1* | *webadmg1* |
| /etc/*websrv1*/httpd.envvars | 560 | faf | *websrv1* | *webadmg1* |
| /etc/*websrv1*/mvsds.conf | 460 | faf | *websrv1* | *webadmg1* |

The directory ***websrv1*** is a site-selected name, but must be unique for each web server. The actual name of the **mvsds.conf** file, if used, is specified in the **ServerInit** directive for the MVSDS DLL service.

### G.3.3.4 Local Server Log Directories

The web server log files contain access and error information that could be critical for security audit tasks. See *Section G.3.14, Access and Error Logging*, for a description of the requirement to place these directories in a file system that is separate from server software or documents. This is required to reduce the potential system impact and to protect the data. These directories must be unique for each web server.

- *The permission and user audit bits and owner and group settings for the local server log directories* will *be configured according to the settings in the following table:*

**Table G-4. IHS Local Server Log Hfs Object Security Settings**

| *IHS LOCAL SERVER LOG HFS OBJECT SECURITY SETTINGS* | | | | |
|---|---|---|---|---|
| ***DIRECTORY or FILE*** | ***PERMISSION BITS*** | ***USER AUDIT BITS*** | ***OWNER*** | ***GROUP*** |
| …/*websrv1_root*/logs | 750 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/logs/httpd-log | 750 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/logs/httpd-errors | 750 | fff | *websrv1* | *webadmg1* |
| …/*websrv1_root*/logs/cgi-error | 750 | fff | *websrv1* | *webadmg1* |

The directory ***websrv1_root*** is a site-selected name, but must be unique for each web server.

The directories listed here contain the active access and error log files. In *Section H.3.14, Access and Error Logging*, there are requirements to maintain archives of those files. While the site can choose the names and storage locations (HFS files, MVS data sets, tape data sets), access to those archive files must be limited.

- *All access to archived web server access and error logs* will be *restricted to systems programming personnel, security administrators, and web server administrators.*

### G.3.3.5 Other Server Files and Directories

There are a number of other files and directories that the web server uses that require access control. These objects do not have fixed names, so the information here is relative to the configuration directive or environment variable that is associated with the object.

Because a web server provides an application environment, controlling access to the application data and programs (referred to as web content) of that environment requires a configuration management policy that balances the security and stability of the server with practical operating procedures for web content owners. Some common types of policies that are used for access control are administrator managed, content owner managed, and a combination of those two.

- **Administrator managed** – Under this type of policy, the web administrator is responsible for moving web content from staging directories to the directories specified in web server directives. The content owners do not have access to update the directories or files specified in web server directives. This policy is recommended for managing production web content because it enhances the security of the environment.

- **Content owner managed** – Under this type of policy, content owners are responsible for moving web content from staging directories to the directories specified in web server directives. This policy is recommended for managing development or test web content because it simplifies procedures and reduces implementation time.

Sites can choose a web content configuration management policy that is appropriate for their environment as long as certain requirements are followed. The policy must be consistent with the *DOD Web Site Administration Policies and Procedures*, dated 25 November 1998. It must be documented to the IAO, and the programs and data referenced by the web server must have appropriate access control.

- *The content configuration management policy for each web server* will *be documented to and approved by the IAO. The documentation* will *include the general procedures used to update the directories and files referenced by the web server and the parties authorized to perform those procedures.*

- *The permission bits and owner and group settings for the web content directories and files identified in server directives will be configured according to the settings in the following table:*

**Table G-5. IHS Web Content Hfs Object Security Settings**

| *IHS WEB CONTENT HFS OBJECT SECURITY SETTINGS* | | | | |
|---|---|---|---|---|
| ***DIRECTIVE TYPE*** | ***POLICY TYPE*** | ***PERMISSION BITS*** | ***OWNER*** | ***GROUP*** |
| **Pass** | Administrator | 555 | *websrv1,* software-owning ID, or ID with UID(0) | *webadmg1* or other software-owning group |
| | Content Owner | 775 | ID managed by Content Owner group | Content Owner group |
| **Exec** | Administrator | 555 | *websrv1,* software-owning ID, or ID with UID(0) | *webadmg1* or other software-owning group |
| | Content Owner | 775 | ID managed by Content Owner group | Content Owner group |
| **GWAPI** | Administrator | 555 | *websrv1,* software-owning ID, or ID with UID(0) | *webadmg1* or other software-owning group |
| | Content Owner | 775 | ID managed by Content Owner group | Content Owner group |

The following notes apply to these settings:

- Sites are not required to change the vendor-specified permission bit settings for software directories and files that are covered by a Vendor Integrity statement unless the site modifies the contents of the directories. The site may be required to provide a copy of the vendor documentation that confirms the settings.

- More restrictive permission bit settings (i.e., 0, 1, or 4) for the **other** category should be used where possible.

- More restrictive permission bit settings (i.e., 1 or 4) for the **group** category should be used where possible.

- Where "software-owning group" is indicated, the group should not be one to which non-privileged users are assigned.

- Where "ID managed by Content Owner group" is indicated, the ID should be assigned to a specific individual.

- *For directories specified by **Pass** directives, the following is required:*

  - *The directory and all subdirectories will not contain any Java source (**.java**) files, CGI program (**.cgi**, **.class**, **.sh**) files, or DLL program (**.so**) files.*

  - *If symbolic links are used in the named directory or any subdirectories, the target of the link must be owned by the same ID and group as the directory in which the link resides.*

- *For directories specified by **Exec** directives, the following is required:*

  - *The directory and all subdirectories will not contain any Java source (.java) files.*

  - *For production web servers, the directory and all subdirectories will not contain any sample or backup copies of programs.*

- *For directories specified by GWAPI directives listed in Section G.3.10.3, Go Webserver API (GWAPI) (e.g., **ServerInit, Service, and ServerTerm**) directives, the following is required:*

  - *The directory and all subdirectories will not contain any Java source (.java) files.*

  - *For production web servers, the directory and all subdirectories will not contain any sample or backup copies of programs.*

Some directories and files represent common elements that are used by, and may be functional components of, other software. To ensure the security of all the environments that use these elements, some specific access controls are required.

- *For directories specified by the **LIBPATH, NLSPATH**, and **PATH** environment variables, the following is required:*

  - *The directory will have permission bits of 755 or more restrictive.*

  - *The directory will be owned by the web server ID, an ID with UID(0), or a software-owning ID controlled by the systems programming group.*

- *If Java is installed and enabled to the web server, for directories specified by the **JAVA_HOME** and **CLASSPATH** environment variables, the following is required:*

  - *The directory will have permission bits of 755 or more restrictive.*

  - *The directory will be owned by the web server ID, an ID with UID(0) , or a software-owning ID controlled by the systems programming group.*

Some directories and files are used in specific server configurations. To ensure the security of the server and the operational procedures used with the server, some specific access controls are required.

- *For servers configured for SSL connections and running Version 5.2 of IHS for OS/390, the file specified by the **KeyFile** directive and the associated stash file will have permission bits of 700, be owned by the web server ID, and reside in a directory with permission bits of 700. No other files or subdirectories will reside in the directory containing the **KeyFile** file.*

- *For servers configured for LDAP access using a password, the file specified by the **ServerPasswordStashFile** subdirective of the **LDAPInfo** directive will have permission bits of 700, be owned by the web server ID, and reside in a directory with permission bits of 700. No other files or subdirectories will reside in the directory containing the **ServerPasswordStashFile** file.*

- *The directory containing the file specified by the **PidFile** directive will have permission bits of 700 and be owned by the web server ID. The directory will not be a subdirectory of **/tmp** or any directory in a temporary file system (TFS).*

### G.3.4 Client Identification and Authentication

In IHS for OS/390 each request that a web server receives from a client executes as a unit of work under a specific security context. That is to say, each request is associated with a specific user identifier. The associated ID can come from data supplied by the client or from server configuration statements; it is referred to as the access control userid.

The client can explicitly provide the access control userid. Identification data from the client can be in the form of a response to a prompt for an ID and password that is defined to the ACP. The client can also supply a digital certificate during a Secure Sockets Layer (SSL) connection. A digital certificate is mapped to an ID defined to the ACP. The server calls the ACP to validate either form of identification before the request is served.

Server configuration statements can automatically assign an access control userid to a request. This is usually done when users are not required to identify themselves, such as for an **Open** web server. It can also be done when users are required to identify themselves; but after identification and authentication, are to be grouped for the purpose of transaction security. The assigned access control ID is referred to as a surrogate ID. Because the web server sometimes assigns a surrogate ID without user identification or authentication, caution is essential in configuring the use of surrogate IDs. In order for a web server to use a surrogate ID, the server must be given access to the appropriate System Authorization Facility (SAF) resources as discussed in *Section G.3.1.1, Web Server Identification*.

This section discusses the configuration directives and subdirectives that are involved in selecting an access control userid and in specifying whether and how authentication is performed. Although the subdirectives discussed here are part of protection setups discussed in *Section G.3.5, Access Control Directives and Protection Setups*, they are described here because they directly impact client identification and authentication.

The **UserID** directive defines the default access control userid used by the web server. It is used to assign a surrogate ID or force the client to provide an ID when there is no applicable protection setup that supplies one. Options for the **UserID** directive include the following:

- **%%CLIENT%%** - When **%%CLIENT%%** is used, the user is prompted for an ID and password that is validated by the ACP. The supplied ID is assigned as the access control userid.

- **%%CERTIF%%** - The **%%CERTIF%%** option applies to SSL sessions and causes the web server to attempt to match the client PKI certificate to one that is defined for an ID known to the ACP. If the certificate maps to an ACP-defined ID, the mapped ID is assigned as the access control userid. If certificate validation cannot be done, identification proceeds as if **%%CLIENT%%** was specified.

- **%%SERVER%%** - When **%%SERVER%%** is used, the ID of the web server itself is assigned as the access control userid.

- A Surrogate ID – A specific surrogate ID that is defined to the ACP can be assigned automatically as the access control userid.

Because the **UserID** directive defines the default access control userid, a value that forces the client to provide identification and forces the ACP to validate it is the appropriate choice for a default. Using **%%SERVER%%** could be a significant vulnerability because all files accessible to the web server's ID would also be accessible to any client. Using a surrogate ID could allow access without any identification or authentication.

- *The **UserID** directive will be specified with a value of **%%CLIENT%%** or **%%CERTIF%%**.*

Protection setups are groups of subdirectives used as the primary means of defining access controls for requests to a web server. The subdirectives that impact client identification and authentication are **AuthType**, **GroupFile**, **Mask** (including **DeleteMask**, **GetMask**, **Mask**, **PostMask**, **PutMask**), **PasswdFile**, **ServerID**, and **UserID**. These subdirectives and any required settings are discussed in this section. There are additional subdirectives that can be used for connections using SSL. These subdirectives limit access on the basis of elements in the client's certificate. Please refer to *Section G.3.9, Secure Sockets Layer (SSL) Configuration*, for information on those subdirectives.

The **AuthType** subdirective specifies that an ID and password must be supplied for the request. The only valid value in IHS for OS/390 is **Basic**. It is important to remember that when **AuthType Basic** is specified, the client-supplied ID and password are not encrypted for transit between client and server.

In order to cause ID and password information to be encrypted in transit, sites should consider specifying SSL connections when **AuthType Basic** is specified.

The **GroupFile** subdirective can specify the path and file name of a server group file to be used in the associated protection setup. A server group file names one or more groups and defines the users in the groups. The users can be specified in terms of userids, other group names, and address templates. The **GroupFile** subdirective can also specify one or more LDAP servers that are defined by an **LDAPInfo** directive.

Groups can be used in mask subdirectives within protection setups and in ACL files. When implemented, group files impact access control decisions. As a result it is necessary that update access to the files be restricted.

- *If a **GroupFile** subdirective specifies a file, the named file must have permission bits of 660 or more restrictive and be owned by the web server ID and a web administrator group.*

The **DeleteMask**, **GetMask**, **Mask**, **PostMask**, and **PutMask** subdirectives are collectively called mask subdirectives. Mask subdirectives are used to identify users, groups, and address templates of clients authorized to make requests. The **DeleteMask**, **GetMask**, **PostMask**, and **PutMask** subdirectives correspond respectively to the DELETE, GET, POST, and PUT HTTP methods. The **Mask** subdirective applies when a more specific mask subdirective does not apply.

A mask subdirective can be used in multiple ways:

- It can specify IDs and groups of IDs that are authorized for the matching request. In addition, a value of **All** or **Users** indicates any ID authenticated via the mechanism defined by the **PasswdFile** subdirective.

- It can specify that no ID or password information is required from the client. The use of the values **@**, **Anybody@**, **Anyone@**, or **Anonymous@** without an address template or with an address template of (*****) indicate no protection for the request.

- It can specify that authentication is to be based on an address template. An address template is a template for the host name or IP address of the client.

The following notes apply to the use of mask subdirectives:

- Mask specifications are case sensitive. Accordingly a specification of **Mask WEBSYS1** is not equivalent to **Mask websys1**.

- When mask subdirectives use host names rather than IP addresses, the **DNS-Lookup** directive must have a value of **On**. This allows the server to get an IP address for the host name in the directive.

- Only groups defined in a group file can be used. Groups defined in the ACP cannot be used.

- *When used, mask subdirectives will not specify **@** (by itself), **Anybody@**, **Anyone@**, or **Anonymous@** without an address template that limits the connection source. The only* exception t*o this requirement is for servers defined as Public web servers and protection setups using SSL-specific subdirectives.*

The **PasswdFile** subdirective specifies the source to be used to authenticate IDs supplied by clients. Options for the **PasswdFile** subdirective include the following:

- **%%SAF%%** - When **%%SAF%%** is used, it specifies that SAF calls to the resident ACP are used to perform authentication.

- **%%LDAP%%:***LDAPInfo-label* - When **%%LDAP%%:***LDAPInfo-label* is used, it specifies that an LDAP server, defined by an **LDAPInfo** directive, is used to supply information for authentication.

- A Password File - A file identified by path and file name specifies that an HFS file contains IDs and passwords to be used for authentication.

The requirement in the DISA *OS/390 STIG* is that users of resources be defined using ACP facilities to control identification and authentication. This policy affects permitted values for **PasswdFile**.

- *When used, the **PasswdFile** subdirective will specify **%%SAF%%** or **"%%LDAP%%:***LDAPInfo-label***"** so that authentication is always managed ultimately by an ACP.*

- *If the **PasswdFile** subdirective specifies **"%%LDAP%%:***LDAPInfo-label***"**, the backend database used in the referenced LDAP server must be an ACP (i.e., ACF2, RACF, or TOP SECRET).*

The **ServerID** subdirective specifies the server name to be associated with the password mechanism as specified by the **PasswdFile** subdirective. The value of the **ServerID** subdirective impacts the behavior of some browser clients. The browser usually displays this name when prompting for userids and passwords. The browser may also cache userids and passwords under this name so that the ID/password prompt is only issued once per server name.

Although the stated requirements for the **PasswdFile** subdirective result in a common password mechanism (i.e., **%%SAF%%**) in most cases, a single value should generally not be used for all or most **ServerID** subdirectives. A unique **ServerID** subdirective value should be used in all protection setups for each logically distinct application. For example, each protection setup for elements of application A should specify **ServerID Server_App_A** and each protection setup for elements of application B should specify **ServerID Server_App_B**.

Using unique **ServerID** subdirective values for each logical application provides two benefits. It enables individual applications to utilize unique IDs if desired. It also enhances security and auditability by forcing users to identify and authenticate themselves for each application.

The **UserID** subdirective specifies the access control userid to be used by the web server for requests covered by the protection setup in which the **UserID** subdirective appears. When a **UserID** subdirective is not specified, the default access control userid (i.e., the value from the **UserID** directive) is used.

The options for the **UserID** subdirective are the same as those described earlier for the **UserID** directive—**%%CLIENT%%**, **%%CERTIF%%**, **%%SERVER%%**, or a surrogate ID. The vulnerability described when using **%%SERVER%%** is the same for the subdirective as for the directive. However the use of a surrogate ID in the **UserID** subdirective is permitted in appropriate circumstances.

- *The **UserID** subdirective will be specified with a value of **%%CLIENT%%, %%CERTIF%%**, or a surrogate ID.*

The use of surrogate IDs is appropriate for **Public** web servers (i.e., those accessible to all Internet users). Using surrogate IDs is also appropriate for web servers with limited access under the following conditions:

- If users are authenticated using IDs and passwords or digital certificates, use of a surrogate ID is acceptable.

- If users are not authenticated, use of a surrogate ID is acceptable only if requests are restricted based on a mask directive that specifies an address template.

- *If a **UserID** subdirective specifies a surrogate ID, at least one of the following requirements will be met:*

  - *The web server will meet the requirements for a **Public** web server.*

  - *The protection setup in which the **UserID** subdirective specifies a surrogate ID will also specify **AuthType Basic** and **PasswdFile %%SAF%%**.*

  - *The protection setup in which the **UserID** subdirective specifies a surrogate ID will also specify a mask subdirective that restricts access on the basis of client network address.*

  - *The protection setup in which the **UserID** subdirective specifies a surrogate ID will also specify one of the SSL-related subdirectives that limit access on the basis of an element of the client's or the CA's Distinguished Name.*

- *The userid assigned to a web server will have **read** access to the needed **BPX.SRV.user SURROGAT** class SAF resources, where **user** is a surrogate ID.*

When surrogate IDs are defined to the ACP, the principle of least privilege should be followed when defining those IDs. Because surrogate IDs do not require privileges that may be common for other IDs, certain restrictions available through the ACP are required.

- *Web server surrogate IDs will not have ACP privileges to logon to other online systems. This includes, but is not limited to, TSO, CICS, and ROSCOE. Implementing this through ACP controls that restrict system entry by prohibiting logon passwords (i.e., **RESTRICT** for ACF2, **NOPASSWORD** for RACF, and **PASSWORD(NOPW,0)** for TOP SECRET) is an acceptable approach.*

- *One or more local OS/390 UNIX groups will be defined for web server surrogate IDs.*

- *Web server surrogate IDs will be defined to the ACP with the following OS/390 UNIX attributes:*

  - *Non-zero UID*
  - *Home directory '/' or a directory unique to the surrogate user*
  - *Shell program **/bin/sh***
  - *Defined as a member of a local OS/390 UNIX group for web server surrogate IDs*

An additional restriction on how surrogate IDs are defined is specified by policy in the *Chairman of the Joint Chiefs of Staff Manual (CJCSM) 6510.01, "Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)*". That document states, "All factory-set, default, or standard userids and passwords will be removed or changed prior to the system going operational." **WEBADM**, **PUBLIC**, **INTERNAL**, and **PRIVATE** are defined in IBM documentation as surrogate IDs and fit the classification of a standard userid.

- ***WEBADM**, **PUBLIC**, **INTERNAL**, and **PRIVATE** will not be defined as Web server surrogate IDs.*

A general exception to the restrictions on surrogate IDs applies to the use of Lotus Notes software. Because an alternative authentication method is employed, userids defined to the ACP with Lotus Notes identification data (e.g., RACF LNOTES data) may be defined as surrogate IDs and the other protection setup access restrictions are not required.

### G.3.5 Access Control Directives and Protection Setups

Configuration statements that define resource access controls for IHS for OS/390 are referred to as access control directives. The server uses these directives to define the identification and authentication parameters used to determine if a given request for a file is allowed or denied.

The access control directives are accompanied by subdirectives. These subdirectives are used in combinations to define how the server controls access to the specified resources. A group of protection subdirectives is called a protection setup.

The access control directives are as follows:

- **DefProt** - A **DefProt** directive specifies a template for a file requested by a client and specifies either a) a named protection setup (defined by a **Protection** directive), b) in-line protection subdirectives, or c) the path and name of a file containing protection subdirectives. **DefProt** directives are used in conjunction with **Protect** directives with request templates that match. A **DefProt** directive can also supply the access control userid to be used for the security environment of the request, but the **UserID** subdirective is the preferred method of specifying the ID.

- **Protect** - A **Protect** directive specifies a template for a file requested by a client and specifies either (a) no protection setup or subdirectives so that **DefProt** values apply, (b) a named protection setup (defined by a **Protection** directive), (c) in-line protection subdirectives, or (d) the path and name of a file containing protection subdirectives. A **Protect** directive can also supply the access control userid to be used for the security environment of the request, but the **UserID** subdirective is the preferred method of specifying the ID.

- **Protection** - A **Protection** directive names and defines a protection setup. Protection subdirectives specify the parameters that make up the protection setup.

The sequence of statements in the server configuration file affects how they are interpreted. An incorrect sequence can cause protection setups to be ignored.

- *All **DefProt**, **Protection** and **Protect** directives will be placed in the server configuration file (httpd.conf) before any **Pass** or **Exec** directives whose templates could match.*

Because coding access control directives can be complex and that complexity could introduce errors, some basic standards enhance security.

- ***DefProt** and **Protect** directives will not specify a path and file name for protection setups. Protection setups will be coded in-line or through **Protection** directives.*

- ***DefProt** and **Protect** directives will not specify an access control ID. The **UserID** subdirective will be used for this purpose.*

Protection subdirectives specify the parameters that make up a protection setup. The protection subdirectives include **ACLOverride**, **AuthType**, **GroupFile**, **Mask** (including **DeleteMask**, **GetMask**, **PostMask**, and **PutMask**), **PasswdFile**, **ServerID**, and **UserID**. Except for **ACLOverride**, these subdirectives are directly related to identifying and authenticating clients. Please refer to *Section G.3.4, Client Identification and Authentication*, for guidelines on those subdirectives.

The **ACLOverride** subdirective can specify that rules in an Access Control List (ACL) file override the masks specified in a protection setup. If this subdirective were used, changes to an ACL file could change or remove the intended access controls as defined in the protection setup.

- *The **ACLOverride** subdirective will not be used.*

The IHS for OS/390 product includes resources called Configuration and Administration forms. These resources consist of HTML forms and CGI programs that can be used to administer the server. In addition to the permission bit protection specified elsewhere, access control directives are used to protect these resources.

- *Access control directives will be coded to restrict access to the Configuration and Administration resources to web server administrators defined to the ACP. Access will require that an ID and password be entered. This setup will cover the following resources:*

  - */admin-bin/**
  - */Docs/admin-bin/**
  - */reports/**
  - */Usage**

The following Protection and Protect directives illustrate to how to meet this requirement:

```
Protection  IMW_Admin {
   ServerID      IMWEBSRV_Administration
   UserID         %%CLIENT%%
   AuthType     Basic
   PasswdFile  %%SAF%%
   Mask           websys1, WEBSYS1, websys2, WEBSYS2
}
Protect /admin-bin/*   IMW_Admin
Protect /Docs/admin-bin/*   IMW_Admin
Protect /reports/*   IMW_Admin
Protect /Usage*   IMW_Admin
```

In this example, *websys1* and *websys2* (with their uppercase versions) represent the IDs of web server administrators at the site.

To provide explicit default protection for requests that do not match any other **Protect** directives, a **Protect** directive with a request template of only an asterisk is used.

- *A **Protect** directive to provide default request protection will be coded and will be placed in the configuration file before all other **Protect** directives. The directive and its in-line subdirectives will be coded as follows:*

```
Protect  *    {
   ServerID      System_Logon
   UserID         %%CLIENT%%
   AuthType     Basic
   PasswdFile  %%SAF%%
   Mask           All
}
```

The following notes apply to this requirement:

- In this specification, ***System_Logon*** represents a name selected by the site. Although the value ***System_Logon*** can be changed, the **ServerID** subdirective cannot be omitted.

- A more restrictive value than ***All*** for the **Mask** subdirective can be coded at the site's discretion.

Although this directive is provided by the software as a vendor default when the **UserID** directive specifies **%%CLIENT%%**, an explicit specification is used to enhance security in the event that the vendor default is changed.

### G.3.6 Access Control List (ACL) Files

An Access Control List (ACL) file can provide a more granular level of access control for files in a protected directory. A protected directory can have only one ACL file. It must be named **.www_acl** and must be present in the protected directory. The **UseACLs** directive controls whether ACL files are checked in access control decisions.

ACL files can limit access based on file name, HTTP method, and authorized users, groups, or IP addresses. If SSL client authentication is being used, parts or all of a client's Distinguished Name (DN) or the Certificate Authority's DN can be specified in the ACL file as access criteria.

Because ACL files impact access control decisions, it is necessary that update access to the ACL files be restricted.

- *If used, ACL files will have permission bits of 660 or more restrictive. The files will be owned by (a) the web server ID and a web administrator group, or (b) the same owner and group as the directory in which the ACL file resides.*

### G.3.7 Resource Mapping Directives

In IHS for OS/390, resource mapping directives define documents and programs as resources that can or cannot be served to clients. These directives can also translate client-specified names into the names of files that are physically on the server. By allowing "virtual" names to be specified, the directives hide the server's physical file structure from clients. This provides greater flexibility and security.

The **Pass** and **Exec** directives define the requests that the server accepts for processing. **Pass** directives define documents; **Exec** directives define CGI programs. **Fail** directives define resources for which the server rejects requests.

Document directories, as defined by the **Pass** directives, are subject to the welcome file requirement documented in *Section G.3.8, Miscellaneous Directives*. These document directories are also subject to the permission bit requirements documented in *Section G.3.3, Web Server HFS Objects*.

Program directories, as defined by the **Exec** directives, are subject to the permission bit requirements documented in *Section G.3.3, Web Server HFS Objects*.

Sites should code **Protect** directives to correspond with each **Pass** and **Exec** directive. Although the requirements for default protection (i.e., Protect *) specified in *Section G.3.5, Access Control Directives and Protection Setups*, provide generic access control, more specific protection is needed to adhere to the principle of least privilege. Also as required in *Section G.3.5, Access Control Directives and Protection Setups*, **Pass** and **Exec** directives must follow all related **DefProt**, **Protection,** and **Protect** directives in the server configuration file because sequence does impact the server's application of the directives.

The **ExecDirPass** directive controls server behavior when a request matches an **Exec** directive and the resulting path matches a directory rather than a file. If **ExecDirPass** is enabled, the request is treated as if it matched a **Pass** directive and a directory listing or welcome page is displayed. This is undesirable because it might allow a client to see a directory listing of programs the client could attempt to execute.

- *The **ExecDirPass** directive will be specified as **ExecDirPass Off**.*

Although not required, sites should consider the use of the following directives if applicable to their environment:

- **Map** – The **Map** directive is used to change client-specified requests to new values. The new values are checked in subsequent directives in the configuration file. This is one way of allowing virtual rather than physical directory and file names to be given to clients.

- **Redirect** – The **Redirect** directive is used to forward client requests to a different server. When used with the multiple IP address support and the host name operand, it can redirect clients from a specific network (e.g., .com) to another server that could perform different authentication services.

- **InheritEnv** and **DisInheritEnv** – The **InheritEnv** and **DisInheritEnv** directives affect which environment variables are passed to CGI programs. When these directives are not specified, all environment variables are passed to the program. A site might use these directives to effectively exclude variables considered sensitive for the workload in a specific server.

### G.3.8 Miscellaneous Directives

In addition to those discussed elsewhere in this appendix, there are numerous configuration directives for IHS for OS/390 that define operational parameters. These directives span categories such as basic, directories and welcome page, system management, and timeouts. This section discusses the directives in those categories that have impacts on server security.

The **InstallPath** directive specifies the directory in which the server software is installed. The **ServerRoot** directive specifies the current working directory of the server. While the directory for the server software is typically the same for all servers on one host, the working directory is typically unique.

- *The **InstallPath** and **ServerRoot** directives will be explicitly specified in the **httpd.conf** file.*

- *The **ServerRoot** directive will specify a unique directory for each web server.*

- *The directory specified by the **ServerRoot** directive will not be available via network file sharing services such as the Network File System (NFS) or Distributed File Service (DFS).*

Permission bit settings for the directories specified in the **InstallPath** and **ServerRoot** directives are discussed in *Section G.3.3, Web Server HFS Objects*.

The **AlwaysWelcome**, **DirAccess**, and **Welcome** directives impact the server's behavior when a request references an HFS directory rather than a file name. Displaying a welcome file rather than a directory listing improves server security by preventing clients from displaying directories to scan for potentially sensitive content.

- *The **AlwaysWelcome** directive will be specified as **AlwaysWelcome On**.*

- *The **DirAccess** directive will be specified as **DirAccess Off**.*

- *One or more **Welcome** directives will be specified with valid HFS file name(s).*

- *The server root directory (as indicated by the **ServerRoot** directive) will have a welcome file.*

Sites should consider a method to return helpful information to clients for errors that occur when a directory or file is not found. This can be accomplished by including a welcome file in each document directory or by using **ErrorPage** directives with the **multifail** and **notfound** operands to specify a customized error page. The welcome file or customized error page could include a point of contact to which the error could be reported.

The **imbeds** directive controls server-side include (SSI) processing. SSIs allow information to be dynamically inserted at the time a file is served. SSIs are invoked by embedding SSI directives into files from content directories or files created by CGI processing. The server intercepts the SSI directives and performs any required substitution.

The **exec** SSI directive is intended to invoke a CGI program during SSI processing. Because the use of these CGIs would not be subject to the access control directives in the server's configuration file, security vulnerabilities could be introduced. The **noexec** operand on the **imbeds** directive is used to disable the **exec** SSI directive.

It is possible to restrict the type of documents for which SSI processing is enabled by specifying that only those documents with a content type of **text/x-ssi-html** are to be processed. A document's content type is indicated by the document's suffix (e.g., **.shtml**) and the **AddType** directives in the server's configuration file. Restricting SSI processing in this way provides better control of access to potentially vulnerable SSI capabilities. The **SSIOnly** operand on the **imbeds** directive is used to restrict SSI processing.

- *If the site uses SSI processing, the **imbeds** directive will be specified as **imbeds noexec SSIOnly**. If not used, it will be specified as **imbeds off**.*

The **MaxActiveThreads** directive controls the number of threads that can be active at one time. Although the value of this directive is primarily a performance issue, an invalid value might allow a denial of service condition to develop.

- *Unless documented and justified to the IAO, the **MaxActiveThreads** directive will specify a value between 10 and 250.*

The **LDAPInfo** directive, along with its subdirectives, allows IHS for OS/390 to use an LDAP server in access control decisions. The nature of this function makes it essential that the configuration of LDAP access be appropriate.

- *For all **LDAPInfo** directives, the **Host** subdirective will specify the name of a host that is within the security enclave of the site.*

To use an LDAP server, the web server connects as a client to the LDAP server. LDAP servers may support both anonymous and authenticated client access. If the LDAP server requires authentication, the web server uses its own identity or the identity of a web server client depending on the type of transaction. Because the connection to the LDAP server might carry sensitive information such as IDs, passwords, and user privilege information, security controls on the connection are required.

- *For all **LDAPInfo** directives, the **ServerAuthType** subdirective will be explicitly specified as **ServerAuthType Basic**.*

When **ServerAuthType Basic** is specified, the **ServerPasswordStashFile** subdirective has to identify a file containing the encrypted password used to access the LDAP server. Please see *Section G.3.3, Web Server HFS Objects,* for the permission bit requirements for this file.

- *For all **LDAPInfo** directives, if the host referenced in the **Host** subdirective is not the host on which the web server is running, the **Transport** subdirective will be specified as **Transport SSL**.*

IHS for OS/390 provides support for the Simple Network Management Protocol (SNMP) to enable status information to be retrieved. Support can be enabled through the **SNMP** directive. The **SNMPCommunityName** directive specifies the SNMP community name that is used to control access. Consult the DISA *Network Infrastructure STIG* for the requirements for using SNMP to perform network management. In addition to the **SNMP** directive, the SNMP support can be enabled through a server startup option. As a result, coding the **SNMPCommunityName** directive with a non-default value provides a minor security enhancement if the SNMP support is inadvertently enabled.

- *The **SNMPCommunityName** directive will be coded and will specify a value other than the default 'public'.*

There are several directives in IHS for OS/390 that control the amount of elapsed time certain types of operations may take before the connection or operation is terminated. Although values for these timeout directives are primarily performance issues, invalid values might allow a denial of service condition to develop.

Due to the wide variations among system and network configurations, specific timeout values are not required. The sites should consider values within the ranges noted in the following table and explicitly code those values in the server configuration file.

**Table G.6. IHS Timeout Directives**

| *IHS TIMEOUT DIRECTIVES* | | |
|---|---|---|
| ***DIRECTIVE*** | ***DESCRIPTION*** | ***RECOMMENDED RANGE*** |
| **InputTimeout** | The time allowed for a client to send a request after making a connection. | 30 seconds - 2 minutes |
| **OutputTimeout** | The time allowed for the server to send output for local files to a client. | 2 minutes - 5 minutes |
| **ScriptTimeout** | The time allowed for a program (e.g., CGI) started by the server to finish. | 2 minutes - 10 minutes |
| **PersistTimeout** | The time allowed between client requests on a persistent connection. | 5 seconds - 10 seconds |

The values indicated as the lower end of a range are the default values. IBM has stated that high timeout values can cause performance degradation on a heavily loaded system.

There are some additional timeout directives that only impact SSL connections. For requirements concerning those directives, please see *Section G.3.9.1, SSL Connection Options*.

### G.3.9 Secure Sockets Layer (SSL) Configuration

The IBM HTTP Server for OS/390 is capable of using the Secure Sockets Layer (SSL) protocol in sessions with compatible web browser clients. The use of SSL can provide server authentication, data integrity, and, optionally, client authentication and data encryption.

IHS for OS/390 is capable of supporting Versions 2 and 3 of the SSL protocol. As discussed in a following subsection, combinations of **SSLCipherSpec** directives control whether the server supports one or both versions. If client authentication is desired, SSL Version 3 is required.

Four areas for consideration are discussed in this section:

- **SSL Connection Options** – SSL connection and session options impact the security provided by an SSL connection.

- **Authentication and Access Control** – Server authentication is assumed in SSL, but client authentication is optional. Appropriately configured protection setups enable ID and password authentication to be replaced by client certificate information.

- **Certificate Management** – Server certificates, Certificate Authority (CA) certificates, and, optionally, user certificates have to be managed. Newer software supports certificate storage in the ACP database.

- **Encryption** – Different strengths or no encryption are configuration options.

In Version 5.3 of IHS for OS/390, the server calls the OS/390 System SSL component to support SSL processing. In the DISA environment where certain authorization resources are secured with the **BPX.DAEMON** and **BPX.SERVER** SAF resource definitions, the System SSL program data set (usually SYS1.GSK.SGSKLOAD) must be marked as program controlled or the server is unable to support SSL connections.

### G.3.9.1 SSL Connection Options

Several configuration file directives are involved in specifying how the server supports SSL connections. The **SSLMode** and **SSLPort** directives provide basic control options. The **SSLV2Timeout** and **SSLV3Timeout** directives affect how long negotiated SSL session options are honored.

The **SSLMode** directive specifies whether SSL connections are accepted. If **SSLMode** specifies **on**, connections are accepted over the IP port number specified by the **SSLPort** directive. From a functional perspective, the **SSLMode** and **SSLPort** directives provide the specification for SSL connections that is equivalent to the **NormalMode** and **Port** directives for non-SSL connections.

Sites should consider whether servers that are intended to have only SSL-enabled connections should have the **NormalMode** directive specified as **NormalMode Off**.

During the establishment of an SSL connection, a process known as the SSL handshake is conducted. During the handshake, several tasks take place, including negotiation of the encryption keys and algorithms to be used during the session. Because the handshake process represents significant processor overhead, the SSL protocol allows for the concept of an SSL session that can be reused by subsequent requests. For the lifetime of a session, the encryption key negotiation part of the SSL handshake process does not have to be repeated. To control the length of time an SSL session can persist, timeout values are required. The **SSLV2Timeout** and **SSLV3Timeout** directives specify the number of seconds for which SSL session IDs are considered valid for SSL Version 2 and Version 3 sessions respectively.

The current IBM documentation lists the maximum and default value for the **SSLV2Timeout** directive as 100 seconds. The default value listed for the **SSLV3Timeout** directive is 1000 seconds; the maximum is 86,400 seconds. Due to the potential impact of these values, allowing the directives to be assigned default values is not satisfactory.

- *If the **SSLMode** directive specifies **on**, the **SSLV2Timeout** directive will be explicitly specified with a value between 1 and 100.*

- *If the **SSLMode** directive specifies **on**, the **SSLV3Timeout** directive will be explicitly specified with a value between 1 and 1000.*

### G.3.9.2 Authentication and Access Control

Authentication is one of the primary features of SSL processing. The identity of the server and optionally the client is authenticated through the use of digital certificates. The IBM HTTP Server for OS/390 supports server only or server and client authentication. If client authentication is performed, information from the client's certificate can also be used in access control decisions.

Server authentication is performed for all SSL connections. It is the process in which the client authenticates the server using the certificate provided by the server during connection processing. The **SSLServerCert** directive is optionally used to specify which certificate in a key database is the server's certificate. The label of the certificate and the IP address of the server are specified. **SSLServerCert** is needed in cases where the same server is addressable through multiple IP addresses; otherwise **SSLServerCert** is optional. Since the server's name varies by address, different server certificates are required for a server answering multiple addresses. If the **SSLServerCert** directive is not specified, the certificate marked as the default in the key database identified by the **KeyFile** directive is used.

To provide complete documentation, the sites should specify the **SSLServerCert** directive in the server's configuration file.

Client authentication is optional for SSL connections. It is the process in which the server authenticates the client using the certificate provided by the client during connection processing.

A special note on the use of client certificates is necessary. A complete Public Key Infrastructure includes a means for canceling valid certificates. This may become necessary when a change to access authorization is required. The means to detect a cancelled certificate is Certificate Revocation List (CRL) processing. When a certificate is identified on a CRL, it is considered invalid for use by an SSL-enabled server.

IHS for OS/390 does not directly support storage of a CRL. Use of third party software or an LDAP server may be required. Before enabling client authentication, sites should consider whether a solution that enables CRL processing is required for their environment.

To enable client authentication, the directives necessary for server authentication and an **SSLClientAuth** directive are required. The **SSLClientAuth** directive provides three options that enable client authentication:

- **local** – The **local** option specifies that the server authenticates clients by validating that their certificates are from a Certificate Authority (CA) that is marked as trusted. Trusted CAs are located in the key database specified by the **KeyFile** directive.

- **passthru** – The **passthru** option specifies that the server requests, but does not validate, client certificates. Authentication is the responsibility of a site-provided CGI or GWAPI program.

- **strong** – The **strong** option specifies that the server authenticates clients as in the **local** option, except that the trusted CAs are retrieved from an X.500 directory server specified by the **SSLX500Host** directive.

- *If **SSLClientAuth passthru** is used, a GWAPI program that performs certificate validation will be used. The GWAPI program will be submitted to* DISA *Field Security Operations for Program Integrity Analysis. The program will be reviewed and approved by Field Security Operations prior to implementation in a production web server.*

The **SSLX500CARoots**, **SSLX500Host**, **SSLX500Port**, **SSLX500UserID**, and **SSLX500Password** directives are used when *'***SSLClientAuth strong***'* is specified:

- The **SSLX500CARoots** directive can be used to specify that the certificates of trusted CAs can be found by checking the local key database (**local_only** option) or by checking the local key database followed by the X.500 server specified by the **SSLX500Host** directive (**local_and_x500** option).

- The **SSLX500Host** and **SSLX500Port** directives specify the address and IP port number for the X.500 server to be used to retrieve trusted CA certificates.

- The **SSLX500UserID** and **SSLX500Password** directives specify the distinguished name (DN) and corresponding password to be used by the web server to access the X.500 server.

- *If the **SSLX500Host** directive is coded, it will specify the name of a host that is within the security enclave of the site.*

When client authentication is enabled through the **SSLClientAuth** directive, data from client certificates can be used in protection setups to provide access control. In addition to the **UserID** subdirective, there is one general protection setup subdirective and several SSL-specific subdirectives that can be used.

When the **UserID** subdirective specifies a value of **%%CERTIF%%**, it indicates that the client certificate should be used to establish the access control ID for the request. The certificate provided by the client is matched to one already defined to the resident ACP. The ID associated with the certificate from the ACP is used as the access control ID. Note that if there is a problem with the SSL processing, the server behaves as if '**%%CLIENT%%**' had been specified instead of **%%CERTIF%%**.

The **SSL_ClientAuth** subdirective can be specified with a value of **client** when none of the SSL-specific subdirectives are specified. **SSL_ClientAuth client** specifies that all requests for data covered by the applicable protection setup must be through a session in which a client certificate is supplied.

There are two sets of SSL-specific subdirectives. The first set is composed of certificate elements from the client's Distinguished Name. The second set is composed of certificate elements from the Distinguished Name of the Certificate Authority that signed the client's certificate.

The subdirectives for client DN elements are **CommonName**, **Country**, **Locality**, **StateOrProvince**, **Organization**, and **OrgUnit**. The subdirectives for the CA DN elements are **IssuerCommonName**, **IssuerCountry**, **IssuerLocality**, **IssuerStateOrProvince**, **IssuerOrganization**, and **IssuerOrgUnit**.

The following **Protection** and **Protect** directives illustrate the use of the subdirectives for an SSL session:

```
Protection  App1_Docs {
   ServerID      App1_Server
   UserID         %%CERTIF%%
   OrgUnit="DOD"
   Organization="U.S. Government"
   Mask           Anybody@(*)
}
Protect /App1_Docs_Dir/*   App1_Docs
```

The following notes apply to this example:

- When DN elements are specified, they have to be enclosed in double quotes if they contain blanks or special characters. The supplied information must match the certificate data exactly, including case and embedded spaces.

- When using client certificates, the **Mask** subdirective has to specify one of the generic operands (such as **Anybody**, **Anyone**, or **Anonymous**) in order to avoid a prompt for an ID and password. However, if the certificate provided by the client cannot be matched to one defined in the ACP, the client is prompted to supply an ID and password.

### G.3.9.3 Certificate Management

Digital certificates are a primary requirement of SSL. In this section the following considerations in managing certificates are discussed:

- Location – There are multiple options for storing certificates that the IBM HTTP Server for OS/390 can access.

- Origin – The logical origin of a certificate, the Certificate Authority, is crucial in determining if the certificate should be trusted.

- Name filtering – Multiple certificates can be mapped to a single ID.

Different versions of IHS for OS/390 provide different options for certificate management:

- Version 5.2, bundled with OS/390 Release 2.8, requires the use of an HFS file (manipulated by the IKEYMAN utility) for certificate storage.

- Version 5.3, bundled with OS/390 Release 2.10, allows the use of use of an HFS file (manipulated by the gskkyman utility) or the use of the resident ACP for certificate storage.

The **KeyFile** directive specifies the location where certificates used by IHS for OS/390 are stored. An HFS file, or for Version 5.3, the resident ACP can be specified. Use of the ACP is consistent with security practices required in the DISA's *OS/390 STIG*.

- *For systems running IHS for OS/390 Release 5.3 and above, the **KeyFile** directive, if used, will specify the **SAF** parameter, indicating that the resident ACP manages the digital certificates in use.*

For systems running IHS for OS/390 Release 5.2 that specify the **KeyFile** directive with an HFS file, access to the named file and the associated stash file has to be restricted. Please see *Section G.3.3, Web Server HFS Objects*, for the permission bit requirements for these files.

If certificates are stored in the resident ACP, resources in the FACILITY SAF class are used to control access to these certificates. The ID associated with the web server will require *read* access to the **IRR.DIGTCERT.LIST** and **IRR.DIGTCERT.LISTRING** resources. Please refer to the definition of the web server IDs in the ACP-specific sections within this appendix for implementation details.

Each digital certificate includes Certificate Authority (CA) information as the logical origin of the certificate. The presence of the CA's information indicates, to some level of trust, that the owner of the certificate is recognized by that CA to be who they claim to be. Each host must maintain a list of CAs that are considered trusted. When client authentication is utilized, the CA from the client's certificate is compared to the host's list. If there is a match, a major criterion of SSL authentication is satisfied. Therefore, the list of CAs maintained on the host has a crucial impact on authentication decisions.

Software is available on most host platforms, including OS/390, that allows a host to act as a Certificate Authority. When certificates are created on that host for use on that host, the certificates are considered to be self-signed. Certificates that are self-signed are generally considered to be of limited security value because no independent oversight of user identification is maintained.

- *For hosts running production web servers, the list of Certificate Authorities considered trusted by the OS/390 host will be limited to those with a trust hierarchy that leads to a DOD PKI Root Certificate Authority.*

- *For production web servers, self-signed certificates will not be used.*

Certificate name filtering is a facility that allows multiple certificates to be mapped to a single ACP ID. It was introduced in OS/390 Release 2.10. Rather than matching a certificate stored in the ACP to look up an ID, certificate name filtering uses criteria rules stored in the ACP. A filter rule uses parts of the distinguished name of the certificate owner and/or issuer (CA) to determine an ID to assign to the user. Depending on the filter criteria, a large number of client certificates could map to a single ID.

- *Certificate name filtering will not be used unless the filtering rules have been documented to, and approved by, the IAM.*

### G.3.9.4 Encryption

A key benefit of SSL is the data privacy that is provided by session encryption. During the SSL connection process, a mutually acceptable encryption algorithm is selected by the server and client. This algorithm is used to encrypt the data that subsequently flows between the two. However, the level or strength of encryption can vary greatly. In fact, certain configuration options can allow no encryption to be used; others can allow a relatively weak 40-bit algorithm to be used.

**SSLCipherSpec** directives control which encryption algorithms that IHS for OS/390 allows to be used. One **SSLCipherSpec** directive with a cipher identifier is coded for each algorithm to be allowed. The sequence in which the directives appear in the server's configuration file determines the order of preference used in negotiating an SSL connection.

Cipher identifiers in the range 21 to 27 support SSL Version 2. Identifiers in the range 30 to 3A support SSL Version 3. If no identifiers within a range are specified, connections using that version of SSL are not allowed by the server. Please refer to the appropriate version of IBM's *OS/390 HTTP Server Planning, Installing, and Using* document for details on supported algorithms.

- *To prevent the use of null encryption, no **SSLCipherSpec** directive will specify any of the following operands: **30**, **31**, or **32**.*

The strength of encryption to be used depends on the site's processing capacity and the capability of the software used by clients. Sites should consider the following, in the sequence listed, to increase the level of security of their SSL connections:

- Sites should not specify **SSLCipherSpec** directives with the following operands—22, 24, 33, or 36. These specifications permit the weaker 40 bit encryption to be used.

- Sites should not specify **SSLCipherSpec** directives with the following operands—26 or 39. These specifications permit the weaker 64/56 bit encryption to be used.

- Sites should consider specifying the following sequence of directives to cause the highest strength encryption that is available to be used:

  - **SSLCipherSpec 3A**
  - **SSLCipherSpec 35**
  - **SSLCipherSpec 34**
  - **SSLCipherSpec 27**
  - **SSLCipherSpec 21**
  - **SSLCipherSpec 23**

If available and configured on the site's processor, hardware encryption support is automatically used. In Version 5.3 of IHS for OS/390, the encryption processes are handled via calls to the OS/390 System SSL component that, in turn, performs calls to the Integrated Cryptographic Service Facility (ICSF) software. In this configuration, certain resources in the **CSFSERV** SAF class are used to control access to the ICSF services.

The IDs associated with the web server and web users must have read access to certain **CSFSERV** resources when hardware encryption is enabled. Please refer to the definition of the web server IDs in the ACP-specific sections within this appendix for implementation details.

### G.3.10 Application Interfaces

The IBM HTTP Server for OS/390 supports multiple interfaces that enable the execution of application programs. Each interface offers different capabilities and security issues. This section describes some commonly used interfaces including the following:

- Common Gateway Interface (CGI)
- FastCGI
- Go Webserver API (GWAPI)
- Java Servlet

An issue that applies to multiple application interfaces is the designation of controlled programs. Because the web server is able to use security privileges authorized by the **BPX.DAEMON** and **BPX.SERVER** SAF resources, all programs loaded into the server's address space must be known to maintain system integrity. This is indicated through the program-controlled attribute. If a program that is not marked program-controlled is loaded, the address space is marked "dirty" and the server is no longer allowed to perform any of the privileged functions authorized by **BPX.DAEMON** and **BPX.SERVER**.

There are two methods for accommodating the controlled program requirement. By specifying the **_BPX_SHAREAS** environment variable as **NO**, eligible programs are loaded into a new address space rather than the server's address space. This allows programs not marked program-controlled to be executed. See *Section G.3.11, Environment Variables*, for the related requirement.

For programs that must be loaded into the server's address space, the necessary steps must be taken to mark the program or library as program-controlled. Although COTS applications are generally installed with the required attribute, it is the responsibility of the web server administrator or the security administrator to assign the attribute to non-COTS files and libraries. Programs in HFS files are marked as program-controlled through the OS/390 UNIX **extattr** command. Programs in MVS data sets are marked as program-controlled according to ACP-specific procedures. Because assigning the attribute might allow a program to perform privileged functions, the application programming staff should not have authorization to assign this attribute.

#### G.3.10.1 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a de-facto standard interface between web servers and applications. On OS/390 hosts, CGI applications can be written in compiled programming languages such as C++ and in interpreted scripting languages such as REXX. If IBM's Java™ 2 Technology Edition product has been installed, CGI applications can also be written in the Java language. All of these applications are called CGI programs.

Several IBM products supply CGI programs to enable web client access. These products include the Net.Data database utility, the BookManager BookServer documentation server, and the WebSphere MQ (formerly MQSeries) transaction system.

CGI programs are enabled in IHS for OS/390 through the **Exec** directive in the server configuration file. Requirements related to **Exec** directives are documented in *Section G.3.7, Resource Mapping Directives*.

CGI programs are subject to the controlled-program requirement discussed earlier. Using the **_BPX_SHAREAS** environment variable as noted causes CGIs to execute in a separate address space and addresses this requirement.

CGI programs written in the Java language involve some additional issues worth noting. When a Java CGI program is executed, the server loads the OS/390 Java run-time libraries, the OS/390 Java class libraries, and finally the CGI program. The run-time libraries come from the directories specified in the LIBPATH environment variable. The class libraries and Java CGI program come from the directories specified in the CLASSPATH environment variable. As noted in *Section G.3.3, Web Server HFS Objects*, the directories specified by those variables have permission bit setting requirements.

### G.3.10.2 FastCGI

FastCGI is a hybrid application interface designed to combine the ease of CGI programming with the higher performance of a product-specific server interface such as the Go Webserver API (GWAPI). FastCGI is an open standard that is maintained by Open Market, Inc.

IHS for OS/390 provides only part of the components necessary to utilize FastCGI. The FastCGI Developer's Kit has to be obtained from Open Market, Inc., and installed in order for applications to use the interface.

At this time, DISA Field Security Operations has not evaluated the FastCGI components from Open Market and no Vendor Integrity Statement for the product is available. Because the interface would require the program-controlled system privilege, installation of FastCGI is subject to the requirements in the DISA's *OS/390 STIG* for assurance of system integrity.

- *FastCGI will not be enabled to a production web server until the product has been reviewed and approved by* DISA's *Field Security Operations.*

### G.3.10.3 Go Webserver API (GWAPI)

The Go Webserver Application Programming Interface (GWAPI) is an interface that allows applications to run as extensions of the functions provided by the IBM HTTP Server. GWAPI programs can be distinct applications, gateways to other OS/390 applications, and even customized replacements of standard IHS functions.

A GWAPI program must be packaged as a dynamic link library (DLL). IBM defines a DLL as a file containing executable code and data bound to a program at load or run time. Some platform specific considerations apply to the use of DLLs with IHS for OS/390:

- DLLs are subject to the controlled-program requirement discussed earlier.

- DLLs can be HFS files or members of MVS partitioned data sets.

- For DLLs in HFS files, the directory in which the DLL resides must be part of the path specified by the LIBPATH environment variable.

- For DLLs in MVS data sets, an external link must be defined using the OS/390 UNIX **ln** command. This creates a pointer from the HFS name specified in the configuration file directive to the partitioned data set member containing the executable program. The link must specify a name that conforms to member naming standards for partitioned data sets.

  An example of this command is
  '**ln -e *PDSNAME* /usr/lpp/internet/bin/*dllname.so***',
  where *PDSNAME* is the data set member name and *dllname.so* is the HFS name used in the configuration directive.

- The MVS data sets containing DLLs must be part of the Link Pack Area (LPA), system link list, or specified in the STEPLIB for the server.

Several IBM products supply GWAPI programs to enable web client access. IHS for OS/390 provides the MVSDS DLL described in *Section G.3.12, MVSDS DLL Service*, and the GWAPI REXX support. Other IBM products that supply GWAPI programs include the WebSphere Application Server (WAS) and the CICS Transaction Server.

GWAPI programs are enabled in IHS for OS/390 through more than a dozen directives in the server configuration file. The most commonly used GWAPI directives are **ServerInit**, **Service**, and **ServerTerm**. These directives are used primarily to add new applications to the server. Other GWAPI directives such as **Authentication**, **Authorization**, **Log**, and **Error** can be used to change the normal processing done by the server.

GWAPI programs are subject to the controlled-program requirement discussed earlier. In IHS for OS/390, GWAPI programs execute as threads within the server's address space. These programs can use the privileged functions available to the web server and so could become security vulnerabilities if not coded properly.

Because some of the GWAPI directives allow changes to security-sensitive functions of the server, use of these directives requires special attention.

- *Programs specified by **Authentication**, **Authorization**, **Log**, or **Error** directives will be submitted to* DISA *Field Security Operations for Program Integrity Analysis. Programs will be reviewed and approved by Field Security Operations prior to implementation in a production web server.*

GWAPI program directories are specified in the following directives—**Authentication**, **Authorization**, **DataFilter**, **Error**, **Log**, **NameTrans**, **ObjectType**, **PICSDBLookup**, **PostExit**, **PreExit**, **ServerInit**, **ServerTerm**, **Service**, and **WLMClassify**. The directories named in these directives are subject to the permission bit requirements documented in *Section G.3.3, Web Server HFS Objects*.

### G.3.10.4 Java Servlet

Java servlet support is not a unique interface, but rather an implementation of the GWAPI. However, since it functions as a gateway to other applications instead of an application itself, it deserves a brief comment here.

Java servlet support for the IHS for OS/390 is provided through IBM's WebSphere Application Server (WAS) product and through other third-party products. Although a discussion of these products is beyond the scope of this appendix, a couple of considerations worth noting for WAS are identified here.

Implementation of the WAS includes the addition of **ServerInit**, **Service**, and **ServerTerm** directives to the web server configuration file. The programs indicated by those directives are subject to the controlled-program requirement discussed earlier. The directories named in these directives are subject to the permission bit requirements documented in *Section G.3.3, Web Server HFS Objects*.

The WAS reads the **was.conf** configuration file to establish operating parameters. This file should be explicitly named on the **ServerInit** directive and be protected with appropriate permission bit settings.

### G.3.11 Environment Variables

The IBM HTTP Server for OS/390 uses environment variables to obtain some option values and to pass data to other programs. At initialization, the server reads an environment variables file to retrieve variables and their assigned values. Because some of these variables significantly affect server processing, update access to the file must be protected and certain variable values must be set.

The default name of the environment variables file is **/etc/httpd.envvars**. This name is compiled into the IHS for OS/390 program and is used unless explicitly overridden. In consideration of the discussion in *Section G.3.3, Web Server HFS Objects,* about web server configuration files, a different path to the file is recommended.

An environment variables file should be placed in a directory unique to the associated web server. A name such as **/etc/*websrv1*/httpd.envvars**, where *websrv1* is the identity of the web server, should be used.

See *Section G.3.1.3, Web Server Startup Configuration*, for the requirement to explicitly specify the name of the environment variables file in the started task JCL for each web server.

The following table describes some of the variables that can appear in the environment variables file and the security considerations related to those variables.

**Table G-7. IHS HTTPD.ENVVARS Environment Variables**

| ***IHS HTTPD.ENVVARS ENVIRONMENT VARIABLES*** | | |
|---|---|---|
| ***VARIABLE*** | ***DESCRIPTION*** | ***CONSIDERATIONS*** |
| _BPX_SHAREAS | Specifies if spawned programs should run in the web server address space | CGIs and web server logging and reporting should run in a separate address space |
| CLASSPATH | Specifies the search path for the OS/390 Java class libraries and CGI programs | Specified directories require appropriate permission bits |
| GSK_SSL_HW_DETECT_ MESSAGE | Specifies if a message should be written to **stderr** to indicate if hardware encryption is working | Confirms that hardware encryption and the SSL environment are working |
| JAVA_HOME | Specifies the path to the install root for Java | Specified directory requires appropriate permission bits |
| LIBPATH | Specifies the search path for DLL files and the OS/390 Java run-time | Specified directories require appropriate permission bits |
| NLSPATH | Specifies the search path for the message catalog | Specified directories require appropriate permission bits |
| PATH | Specifies the search path for CGI programs in an HFS directory | Specified directories require appropriate permission bits |
| STEPLIB | Specifies the search path for CGI programs in MVS data sets | Specified data sets require appropriate access rules |

- *The following variables will be set to the specified values in the environment variables file:*

  - ***_BPX_SHAREAS=NO** will be specified.*
  - ***GSK_SSL_HW_DETECT_MESSAGE=1** will be specified.*

- *Unless specified differently by requirements in the* DISA's *OS/390 STIG*, ***update** and **alter** access to data sets specified in the **STEPLIB** variable will be restricted to systems programming personnel and web server administrators.*

Permission bit settings for the environment variables file and the directories and files specified in environment variables are discussed in *Section G.3.3, Web Server HFS Objects*.

### G.3.12 MVSDS DLL Service

The MVSDS DLL Service is an optional extension to the IBM HTTP Server for OS/390 that allows content in OS/390 data sets to be displayed by the web server. Configuration statements can specify data sets to be preloaded to enhance performance.

The MVSDS DLL Service is an application that uses the Go Webserver Application Programming Interface (GWAPI). A **Service** directive in the **httpd.conf** file is used to enable the service. If preloading of data sets is desired, **ServerInit** and **ServerTerm** directives must be coded and a configuration file, normally **mvsds.conf**, must be created.

In consideration of the discussion in *Section G.3.3, Web Server HFS Objects***,** about web server configuration files, a different path to the **mvsds.conf** configuration file is recommended.

If used, the **mvsds.conf** configuration file should be placed in a directory unique to the associated web server. A name such as **/etc/*websrv1*/mvsds.conf**, where *websrv1* is the identity of the web server, should be used.

Example directives for enabling the MVSDS DLL Service are as follows:

```
ServerInit   /usr/lpp/internet/bin/mvsds.so:mvsdsInit
   /etc/websrv1/mvsds.conf
Service   /MVSDS*
   /usr/lpp/internet/bin/mvsds.so:mvsdsGet*
ServerTerm   /usr/lpp/internet/bin/mvsds.so:mvsdsTerm
```

The following notes apply to these examples:

- The example **ServerInit** directive explicitly specifies the location of the configuration file.

- In the example **Service** directive, the string **/MVSDS*** is the request template. The request template can be any value that does not conflict with or override other directives. Multiple Service directives, with unique request template values, can invoke the MVSDS DLL Service.

- *If the **ServerInit** directive for the MVSDS DLL Service is specified, the path to the **mvsds.conf** file will be explicitly specified.*

The concept of global data access is implemented in the RACF ACP through the universal access authority (UACC) permission and the global access-checking table and in the TOP SECRET ACP through the ALL Record. In the past these global access features have been used to allow access to data sets under the assumption that all users of the system would have been identified and authenticated to the ACP before access to data is allowed. Because using the MVSDS DLL Service under a surrogate ID could allow data access without individual authentication by the ACP, access to data cannot be allowed to a surrogate ID under the global data access permission. Access to data under explicit access permission is acceptable.

- *For each **Service** directive that invokes the MVSDS DLL Service, access control directives will be coded to restrict access.*

- *For **Private** web servers, access via the MVSDS DLL Service will require an ID and password or a surrogate ID that does not have access to data under the global data access features of the ACP.*

The following **Protection**, **Protect,** and **Service** directives illustrate to how to meet this requirement using an ID and password:

```
Protection  IMW_MVSDS {
   ServerID      IMWEBSRV_MVSDS
   UserID         %%CLIENT%%
   AuthType     Basic
   PasswdFile  %%SAF%%
}
Protect /MVSDS*   IMW_MVSDS
Protect /MVSDS*   IMW_MVSDS
Service /MVSDS*  /usr/lpp/internet/bin/mvsds.so:mvsdsGet*
```

Permission bit settings for the **mvsds.conf** configuration file are discussed in *Section G.3.3, Web Server HFS Objects*.

### G.3.13 Fast Response Cache Accelerator (FRCA)

The Fast Response Cache Accelerator (FRCA) is a dynamic cache manager for the IBM HTTP Server. The FRCA can improve performance when serving text and image files. When a file is first requested, it is loaded from disk. If the FRCA is enabled and the file is eligible for management by the FRCA, a copy is kept in storage to respond to future requests. Files can be automatically loaded, re-loaded, or deleted without the need to maintain a specific list of files to cache.

Certain types of content are not eligible for management by the FRCA. Ineligible content includes dynamically generated files, objects served over SSL connections, files subject to Protect directives without certain mask subdirectives, and files specifically excluded through FRCA directives. Please see the chapter on FRCA in IBM's *OS/390 HTTP Server Planning, Installing, and Using* document for details on eligibility.

Use of the FRCA can have a security implication in terms of file access. After an eligible file is loaded into the FRCA, future requests for the file are satisfied without requiring authentication even when the initial load required authentication. While the eligibility rules eliminate many types of content that might be sensitive, there can be situations in which sensitive information might unintentionally be loaded into the FRCA.

If the **EnableFRCA** directive is coded with a value of **on**, sites should code the **FRCACacheOnly** directive with appropriate operands to explicitly specify the directories or files to be considered for caching.

By default, access to files in the FRCA is logged in the server access log along with non-FRCA access. The **FRCAAccessLog** directive can be used to separate FRCA access logging. Please see *Section G.3.14, Access and Error Logging*, for a discussion of the use of the **FRCAAccessLog** directive.

### G.3.14 Access and Error Logging

Access and error logging are necessary to meet audit trail requirements specified in the DISA's *Computing Services Security Handbook*. IBM HTTP Server for OS/390 provides access and error logs to capture information that is not recorded in sufficient detail by other system components.

This section discusses the configuration file directives related to access and error logs. In addition to some overall directives, those for the server access log, FRCA access log, server error log, and CGI error log are specifically discussed. Although IHS for OS/390 can produce other logs, the site is not required to maintain those logs. Permission bit settings for the directories in which the log files reside are discussed in *Section G.3.3, Web Server HFS Objects*.

There are some general and miscellaneous configuration directives that control logging operations. Values for these directives determine whether any logging is active and if server configuration information is recorded.

- *The **DoReporting** directive will be specified as **DoReporting On** so that the appropriate **AccessLogArchive** or **ErrorLogArchive** directives are applied.*

- *The **NoLog** directive will not be specified.*

- *The **SMF** directive will be specified as **SMF config** or **SMF all**.*

The server access log can contain an entry for each access request. A log entry includes what was requested, who requested it, the return code that indicates if the request was honored, and other information. Because the access log provides the means to track individual accountability, it represents potentially crucial audit information that must be generated and retained.

- *Access logs will be configured as follows:*

  - *Each web server will have its own unique server access log file specified by the **AccessLog** directive.*

  - *Active access log files will reside in a file system that is separate from that which contains the web server software or documents. This file system will not be part of **/tmp** or any temporary file system (TFS).*

  - *Access log files will be retained in online or offline storage for one year for **Private** web servers and for no longer than three months for **Public** web servers. In support of this requirement, the following directive specifications apply:*

    - ***AccessLogArchive** will be specified as **AccessLogArchive none** or **AccessLogArchive userexit** with a site-coded routine to perform archival tasks.*

    - *If the value of **AccessLogArchive** is **none**, **AccessLogExpire** will be specified as **AccessLogExpire 0**.*

    - *If the value of **AccessLogArchive** is **none**, **AccessLogSizeLimit** will be specified as **AccessLogSizeLimit 0**.*

To filter out redundant information and to reduce processing and storage requirements, it is possible to exclude some types of files from access logging. These exclusions can be based on file or directory names (URL), the type of HTTP method in the request, the MIME type of the file requested, and the HTTP status code generated by the server for the request. The **AccessLogExcludeURL**, **AccessLogExcludeMethod**, **AccessLogExcludeMimeType**, and **AccessLogExcludeReturnCode** directives specify access log exclusions.

Because the exclusion of access log information results in the permanent loss of that information, great caution must be taken in using this capability. The primary use of these exclusions should be for graphics (e.g., **.ico**, **.gif**, and **.jpg**) files and HTML formatting (e.g., **.css**) files that accompany content. Excluding files on the basis of file type can be done with the **AccessLogExcludeURL** directive.

- *The **AccessLogExcludeURL** will not specify any of the following:*

  - *Directory names without file qualifiers*

  - *File qualifiers for content files including any type of text file such as plain text or html, any type of audio files, or any type of video files*

  - *File qualifiers for program files including any type of program such as Java server pages, Java archive files, or CGI programs*

  - *File types for any kind of archive such as Zip or cabinet files that contain compressed or reformatted versions of other files.*

- *The use of any **AccessLogExcludeMethod**, **AccessLogExcludeMimeType**, and **AccessLogExcludeReturnCode** directives will be documented and justified to the IAO.*

A separate access log can be created for requests served by the Fast Response Cache Accelerator (FRCA). The **FRCAAccessLog** directive is used to specify the path and file name for the log. If the **FRCAAccessLog** directive is not specified, requests served by the FRCA are logged in the server access log.

Because it separates access data and potentially complicates auditing, the **FRCAAccessLog** directive should not be specified unless an essential need is identified.

- *If the **FRCAAccessLog** directive is specified with a value other than **none**, the log files must be maintained according to the requirements noted earlier for server access logs.*

The server error log includes timeout and access errors encountered by clients. The CGI error log includes standard error output (stderr) from CGI programs. While these errors are not always related to security issues, there are times when the content or pattern of these errors is indicative of a security problem. For these reasons error logs need to be generated and to be retained for a period long enough to investigate security incidents.

- *Server error and CGI error logs will be configured as follows:*

  - *Each web server will have its own unique server error log file and CGI error log file specified by the **ErrorLog** and **CGIErrorLog** directives.*

  - *Active error log files will reside in a file system that is separate from that which contains the web server software or documents. This file system will not be part of **/tmp** or any temporary file system (TFS).*

  - *Error log files will be retained for a minimum of seven days. In support of this requirement, the following directive specifications apply:*

    - *If the value of **ErrorLogArchive** is **purge**, **ErrorLogExpire** will be specified as a number greater than seven.*

    - *If the value of **ErrorLogArchive** is **purge**, **ErrorLogSizeLimit** will be specified as **ErrorLogSizeLimit 0**.*

### G.3.15 IBM Communications Server Considerations

IHS for OS/390 uses TCP/IP services provided by the IBM Communications Server product. In addition, other applications packaged with the Communications Server product can impact the host security environment required for IHS for OS/390. This section discusses the requirements related to those TCP/IP services and other applications.

Some of the configuration parameters for the TCP/IP stack component of the Communications Server are defined in a file known as the **PROFILE.TCPIP** file. Within this file the **PORT**

statement can be used to reserve a TCP/IP port and to define some other operational characteristics. The **PORT** statement operands of interest are as follows:

- ***jobname*** – The ***jobname*** operand is used to reserve the port for a specific job or for certain OS/390 UNIX processes.

- **NOAUTOLOG** – The **NOAUTOLOG** operand specifies that if the TCP/IP address space detects that the server that was using the port is no longer running, it should not automatically restart the server.

- **SAF *resname*** – The **SAF *resname*** operand specifies the name of a site-selected SAF resource that a user of the port must have access to under ACP rules.

Reserving the TCP/IP ports adds a degree of security by preventing other processes from taking control of the port before the web server starts.

- *The **PROFILE.TCPIP PORT** statement will be coded for each port used by a web server. The statement will specify the name of the web server's started task or the OS/390 UNIX kernel (e.g., **OMVS**).*

Preventing automatic restarts can help to ensure that security-related problems are investigated before web servers are restarted after a problem. Sites should consider using the **NOAUTOLOG** operand if its use would be consistent with the availability requirements for the server.

The following illustrates to how to code the **PROFILE.TCPIP** statements to reserve ports 80 and 443 and to prevent automatic restarts of a server listening on port 443:

```
…
80    TCP    OMVS
…
443   TCP    OMVS    NOAUTOLOG
…
```

In this example, ports 80 and 443 are reserved for OS/390 UNIX servers (such as IHS for OS/390) that use the bind() socket API. The server listening on port 443 is not automatically restarted. OMVS is the name of the address space running the OS/390 UNIX kernel.

Sites should consider the use of the **PORT** statement **SAF** operand to further restrict access to web server ports. A **Public** web server environment is an appropriate candidate for using this control.

Certain TCP/IP applications have been specifically associated with vulnerabilities that negatively impact system security. E-mail servers are one type of application that has had this issue. IBM Communications Server provides two e-mail servers—the SMTP server and the OS/390 UNIX sendmail server. Within a secure enclave, the risk is substantially reduced; but platforms that

host **Public** web servers may be exposed to a higher threat level. As a result, restricting the implementation of e-mail servers in higher risk environments enhances security.

- *For hosts running **Public** web servers, the SMTP server and the OS/390 UNIX sendmail server components of the IBM Communications Server will not be enabled.*

### G.3.16 Open Source Software

The IBM HTTP Server for OS/390 provides a standards-compliant environment for web server functions. This environment makes it somewhat easier to customize and extend the functions of the server by adding software. Open source software packages are often targeted at this area. Implementing open source software for web server environments can create security issues that require evaluation.

The DISA's *Computing Services Security Handbook* requires Vendor Integrity Statements to ensure that adding software to an environment does not compromise the integrity of a system. The Handbook states, "An integrity statement is required for all software (operating system, utility, or application) running on a system." The requirement is that software "…not negatively affect the security of the operating system and other applications running on the system."

Additional policy is found in the *Chairman of the Joint Chiefs of Staff Manual (CJCSM) 6510.01, "Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)*". That document restricts usage to "… software, shareware, or public domain software only with the expressed permission or approval of the DAA."

While the use of open source software can represent significant value, the risk of a compromise of system integrity must be minimized. In accordance with the requirements in the DISA's *Computing Services Security Handbook* and *CJCSM 6510.01*, and the discussion of system integrity in the DISA's *OS/390 STIG*, the use of open source software requires evaluation.

- *Open source software will not be enabled to a production web server until the product has been reviewed and approved by* DISA *Field Security Operations or by the applicable designated approving authority (DAA). Documentation of approval by the DAA will be required.*

At this time DISA Field Security Operations has not evaluated the OS/390 ports of popular open source software such as PERL, PHP, OpenSSH, OpenSSL, and MySQL.

## G.4 ACF2 Implementation

This section describes the commands needed to implement the security requirements for IHS for OS/390 under the ACF2 ACP. The following task categories are described:

- Started task and other userid definitions
- Data set protection.

### G.4.1 Started Tasks and Other Logonids

Web server IDs, web server administrator IDs, and optionally surrogate client IDs are defined to enable IHS for OS/390. This section describes commands to define these IDs.

#### G.4.1.1 Web Server IDs

In *Section G.3.1.1, Web Server Identification*, there is a description of the requirements for web server IDs. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websrv1** is the web server ID; it is defined for use as a started task.
- Group **websrvg1** is a local server group for server **websrv1**.
- Group **IMWEB** is defined to use as the group ID in the IBM HFS objects.
- The web server ID is a member of groups **websrvg1** and **IMWEB**.
- Because **websrv1** is not an OS/390 UNIX superuser, the site may wish to update the **ASSIZE**, **CPUTIME**, and **FILEPROC** fields to allow the server to exceed the normal defaults.

The following commands can be used to create the OS/390 UNIX groups and logonid that are required for the server:

**SET PROFILE(GROUP) DIVISION(OMVS)**
**INSERT IMWEB GID(*205*)**
**INSERT** *websrvg1* **GID(*nnn*)**

**SET LID**
**INSERT** *websrv1* **NAME(WEB SERVER 1) GROUP(*websrvg1*) STC MUSASS**
**SET PROFILE(USER) DIVISION(OMVS)**
**INSERT** *websrv1* **UID(*n*) HOME(/usr/lpp/internet) OMVSPGM(/bin/sh)**
**CHANGE** *websrv1* **ASSIZE(*n*) CPUTIME(*n*) FILEPROC(*n*)**

The following addition (in bold) to the indicated rule set can be used to add the IMWEB group access that is required for the web server:

$KEY(IMWEB) TYPE(TGR)
**…**
**- UID(*websrv1-uid*) SERVICE(READ) ALLOW**
**…**

The following additions (in bold) to the indicated rule set can be used to assign the privileges that are required for the web server:

$KEY(BPX) TYPE(FAC)
**…**
**DAEMON UID(*websrv1-uid*) SERVICE(READ) ALLOW**
**…**

**SERVER UID(***websrv1-uid***) SERVICE(UPDATE) ALLOW**
**…**
**SMF UID(***websrv1-uid***) SERVICE(READ) ALLOW**
**…**

The following operator commands are required to complete the updates:

**F ACF2,REBUILD(GRP),CLASS(P)**
**F ACF2,REBUILD(USR),CLASS(P)**
**F ACF2,REBUILD(TGR)**
**F ACF2,REBUILD(FAC)**

For IHS for OS/390 to process SSL connections, the ID associated with the web server must have a digital certificate and key ring. In addition, to be able to authenticate client certificates, the certificates of Certificate Authorities must be available. To accomplish this for systems running IHS for OS/390 Version 5.3 and above, ACF2 is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the server's certificate and those of the Certificate Authorities to the server's key ring.

**SET PROFILE(USER) DIVISION(CERTDATA)**
**INSERT** *websrv1***.CERT01 DSN(***certificate-dataset***) LABEL(***websrv1***-Cert01)**
**TRUST**

**SET PROFILE(USER) DIVISION(KEYRING)**
**INSERT** *websrv1***.RING01 RINGNAME(***websrv1***-Ring01)**

**CONNECT CERTDATA(***websrv1***.CERT01)**
**KEYRING(***websrv1***.RING01)**
**DEFAULT**
**CONNECT CERTDATA(***CERTDATA-of-DOD-CLASS-3-Root-CA-Certificate***)**
**KEYRING(***websrv1***.RING01)**
**CONNECT CERTDATA(***CERTDATA-of-DOD-PKI-Med-Root-CA-Certificate***)**
**KEYRING(***websrv1***.RING01)**

The following operator commands are required to complete the updates:

**F ACF2,REBUILD(USR),CLASS(P)**
**F ACF2,OMVS**

The following considerations apply:

- The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to ACF2.

- The values in **RINGNAME** and **CERTDATA** operands may include lower case characters.

For IHS for OS/390 to authenticate clients with digital certificates, the ID associated with the server must have access to read the clients' key rings and certificates. Resources in the FACILITY SAF class control this access.

The following additions (in bold) to the indicated rule set can be used to assign the privileges that are required for the server:

```
$KEY(IRR) TYPE(FAC)
…
DIGTCERT.LIST UID(websrv1-uid) SERVICE(UPDATE) ALLOW
DIGTCERT.LISTRING UID(websrv1-uid) SERVICE(UPDATE) ALLOW
…
```

The following operator command is required to complete the updates:

```
F ACF2,REBUILD(FAC)
```

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the ICSF component have been defined. The following rule set additions are required to allow the web server and users of the web server to access the ICSF resources. Refer to *Section G.3.9, Secure Sockets Layer (SSL) Configuration*, for more information.

```
$KEY(CSFCK*) TYPE(CSF)
- UID(-) SERVICE(READ) ALLOW
$KEY(CSFPK*) TYPE(CSF)
- UID(-) SERVICE(READ) ALLOW
$KEY(CSF**C) TYPE(CSF)
- UID(-) SERVICE(READ) ALLOW
```

The following operator command is required to complete the updates:

```
F ACF2,REBUILD(CSF)
```

These commands and definitions assume that the default type code for CSFSERV resources is CSF.

Note that these rules allow all authenticated users to access the ICSF resources. This is not recommended. Sites should consider more restrictive rules using their specific ID and group structure.

### G.4.1.2 Web Server Administrator IDs

In *Section G.3.1.2, Web Server Administrator Identification*, there is a description of the requirements for web server administrator IDs. As noted there, an administrator ID is a normal userid, supplemented with the special privileges required. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websys1** is a web server administrator's ID.
- Group **webadmg1** is a local server group for server administrators.
- The web server administrator's ID is a member of groups **webadmg1**, **websrvg1**, and **IMWEB**.

The following commands can be used to create the OS/390 UNIX group and logonid that are required for a web server administrator:

**SET PROFILE(GROUP) DIVISION(OMVS)**
**INSERT** *webadmg1* **GID(***nnn***)**

**SET LID**
**INSERT** *websys1* **NAME(***WEB SYS ADMIN 1***) GROUP(***webadmg1***) -**
**PASSWORD(***password***)**
**SET PROFILE(USER) DIVISION(OMVS)**
**INSERT** *websys1* **UID(***n***) HOME(/u/***websys1***) OMVSPGM(/bin/sh)**

The following additions (in bold) to the indicated rule sets can be used to add the IMWEB and local server group access that is required for the web server administrator:

$KEY(IMWEB) TYPE(TGR)
**…**
**- UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**

$KEY(*websrvg1*) TYPE(TGR)
**…**
**- UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**

The following additions (in bold) to the indicated rule sets can be used to assign the privileges that are required for the web server administrator:

$KEY(BPX) TYPE(FAC)
**…**
**FILEATTR.APF UID(***websys1-uid***) SERVICE(READ) ALLOW**
**FILEATTR.PROGCTL UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**
**SUPERUSER UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**

$KEY(SUPERUSER) TYPE(UNI)
**…**
**FILESYS UID(***websys1-uid***) SERVICE(UPDATE) ALLOW**
**…**
**FILESYS.CHOWN UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**
**PROCESS.GETPSENT UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**
**PROCESS.KILL UID(***websys1-uid***) SERVICE(READ) ALLOW**
**…**

The following operator commands are required to complete the updates:

**F ACF2,REBUILD(GRP),CLASS(P)**
**F ACF2,REBUILD(USR),CLASS(P)**
**F ACF2,REBUILD(TGR)**
**F ACF2,REBUILD(FAC)**
**F ACF2,REBUILD(UNI)**

### G.4.1.3 Surrogate Client IDs

In *Section G.3.4, Client Identification and Authentication*, there is a description of the use and requirements of surrogate IDs. As noted there, surrogate IDs must have limited access authority. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websur1** is a web server surrogate ID.
- Group **websrvg9** is a local server group for surrogate web users.
- ID **websrv1** is the ID of the web server that can act as a surrogate of **websur1**.

The following commands can be used to create the OS/390 UNIX group and logonid that are required for a web server surrogate ID:

**SET PROFILE(GROUP) DIVISION(OMVS)**
**INSERT** *websrvg9* **GID(***nnn***)**

**SET LID**
**INSERT** *websur1* **NAME(WEB SURROGATE 1) GROUP(***websrvg9***) -**
**RESTRICT**
**SET PROFILE(USER) DIVISION(OMVS)**
**INSERT** *websur1* **UID(***n***) HOME(/) OMVSPGM(/bin/sh)**

The following additions (in bold) to the indicated rule set can be used to assign the privileges that are required for the web server to use the surrogate logonid:

$KEY(BPX) TYPE(SUR)
**…**
**SRV.***websur1* **UID(***websrv1-uid***) SERVICE(READ) ALLOW**
**…**

The following operator command is required to complete the updates:

**F ACF2,REBUILD(GRP),CLASS(P)**
**F ACF2,REBUILD(USR),CLASS(P)**
**F ACF2,REBUILD(SUR)**

### G.4.2 Data Sets

As described in *Section G.3.2, Web Server Data Sets*, there are two groups of vendor product data sets associated with IHS for OS/390 that require protection:

- Distribution data sets named with the prefix SYS1.IMW.AIMW
- Target data sets named with the prefix SYS1.IMW.SIMW.

The following additions (in bold) to the SYS1 rule set can be used as a base to secure the MVS data sets:

$KEY(SYS1)
…
**IMW.AIMW- UID(***sysprog-UID***) READ(A) WRITE(A) ALLOC(A) EXEC(A)**
**IMW.AIMW- UID(-) PREVENT**
**IMW.SIMW- UID(***sysprog-UID***) READ(A) WRITE(A) ALLOC(A) EXEC(A)**
**IMW.SIMW- UID(-) PREVENT**
…

## G.5 RACF Implementation

This section describes the commands needed to implement the security requirements for IHS for OS/390 under the RACF ACP. The following task categories are described:

- Started task and other userid definitions
- Data set protection.

### G.5.1 Started Tasks and Other Userids

Web server IDs, web server administrator IDs, and optionally surrogate client IDs are defined to enable IHS for OS/390. This section describes commands to define these IDs.

### G.5.1.1 Web Server IDs

In *Section G.3.1.1, Web Server Identification*, there is a description of the requirements for web server IDs. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websrv1** is the web server ID; it is defined for use as a started task.
- Group **websrvg1** is a local server group for server **websrv1**.
- Group **IMWEB** is defined to use as the group ID in the IBM HFS objects.
- The web server ID is a member of groups **websrvg1** and **IMWEB**.
- Because **websrv1** is not an OS/390 UNIX superuser, the site may wish to update the **ASSIZEMAX**, **CPUTIMEMAX**, and **FILEPROCMAX** fields to allow the server to exceed the normal defaults.

The following commands can be used to create the groups and a userid and assign the privileges that are required for a web server:

**ADDGROUP IMWEB OMVS(GID(*205*)) OWNER(ADMIN)**
**ADDGROUP** *websrvg1* **OMVS(GID(*nnn*)) OWNER(ADMIN)**

**ADDUSER** *websrv1* **DFLTGRP(*websrvg1*) OWNER(ADMIN) -**
**NOPASSWORD NOOIDCARD -**
**OMVS(UID(*n*) HOME('/usr/lpp/internet') PROGRAM('/bin/sh')) -**
**ASSIZEMAX(*n*) CPUTIMEMAX(*n*) FILEPROCMAX(*n*)**
**CONNECT** *websrv1* **GROUP(IMWEB) OWNER(ADMIN)**
**RDEFINE STARTED** *websrv1*.* **UACC(NONE) OWNER(ADMIN) -**
**STDATA(USER(*websrv1*) GROUP(*websrvg1*) TRUSTED(NO))**

**PERMIT BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(*websrv1*)**
**PERMIT BPX.SERVER CLASS(FACILITY) ACCESS(UPDATE) ID(*websrv1*)**
**PERMIT BPX.SMF CLASS (FACILITY) ACCESS(READ) ID(*websrv1*)**

For IHS for OS/390 to process SSL connections, the ID associated with the web server must have a digital certificate and key ring. In addition, to be able to authenticate client certificates, the certificates of Certificate Authorities must be available. To accomplish this for systems running IHS for OS/390 Version 5.3 and above, RACF is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the web server's certificate and those of the Certificate Authorities to the server's key ring.

**RACDCERT ID(*websrv1*) ADD('*certificate-dataset*') -**
**WITHLABEL('*websrv1*-Cert01') TRUST -**
**PASSWORD('*pkcs12-cert-pswd*') -** **/*** Used if PKCS #12 format only ***/**
**ICSF** /* Optional if hardware encryption active ***/**

**RACDCERT ID(*websrv1*) ADDRING(*websrv1*-Ring01)**

**RACDCERT ID(***websrv1***) CONNECT(ID(***websrv1***) -**
**LABEL('***websrv1***-Cert01') RING(***websrv1***-Ring01) -**
**DEFAULT)**
**RACDCERT ID(***websrv1***) CONNECT(CERTAUTH -**
**LABEL('***Label-of-DOD-CLASS-3-Root-CA-Certificate***') RING(***websrv1***-Ring01))**
**RACDCERT ID(***websrv1***) CONNECT(CERTAUTH -**
**LABEL('***Label-of-DOD-PKI-Med-Root-CA-Certificate***') RING(***websrv1***-Ring01))**

**SETROPTS RACLIST(DIGTCERT) REFRESH**
**SETROPTS RACLIST(DIGTRING) REFRESH**

The following considerations apply:

- The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to RACF.

- The values in **WITHLABEL**, **ADDRING**, **LABEL**, and **RING** operands may include lower case characters.

- If the certificate in the data set containing the server's certificate is in PKCS #12 format, the **PASSWORD** operand is required.

- If hardware encryption is enabled, the **ICSF** operand may be used.

For IHS for OS/390 to authenticate clients with digital certificates, the ID associated with the web server must have access to read the clients' key rings and certificates. Resources in the FACILITY SAF class control this access. Refer to *Section G.3.9, Secure Sockets Layer (SSL) Configuration*, for more information.

The following commands can be used to assign the privileges that are required for the web server:

**PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) -**
**ACCESS(READ) ID(***websrv1***)**
**PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) -**
**ACCESS(READ) ID(***websrv1***)**
**SETROPTS RACLIST(FACILITY) REFRESH**

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the ICSF component have been defined. The following PERMIT commands are required to allow the web server and users of the web server to access the ICSF resources. Refer to *Section G.3.9, Secure Sockets Layer (SSL) Configuration*, for more information.

**PERMIT CSF%%C CLASS(CSFSERV) ACCESS(READ) ID(*)**
**PERMIT CSFPK% CLASS(CSFSERV) ACCESS(READ) ID(*)**
**PERMIT CSFCK% CLASS(CSFSERV) ACCESS(READ) ID(*)**

Note that these rules allow **all** authenticated users to access the ICSF resources. This is not recommended. Sites should consider more restrictive rules using their specific ID and group structure.

### G.5.1.2 Web Server Administrator IDs

In *Section G.3.1.2, Web Server Administrator Identification*, there is a description of the requirements for web server administrator IDs. As noted there, an administrator ID is a normal userid, supplemented with the special privileges required. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websys1** is a web server administrator's ID.
- Group **webadmg1** is a local server group for server administrators.
- The web server administrator's ID is a member of groups **webadmg1**, **websrvg1**, and **IMWEB**.

The following commands can be used to create the group and a userid and assign the privileges that are required for a web server administrator:

**ADDGROUP** *webadmg1* **OMVS(GID(***nnn***)) OWNER(ADMIN)**

**ADDUSER** *websys1* **DFLTGRP(***webadmg1***) OWNER(ADMIN) -**
**PASSWORD(***password***) -**
**OMVS(UID(***n***) HOME('/u/***websys1***') PROGRAM('/bin/sh'))**

**PERMIT BPX.FILEATTR.APF CLASS(FACILITY) -**
**ACCESS(READ) ID(***websys1***)**
**PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) -**
**ACCESS(READ) ID(***websys1***)**
**PERMIT BPX.SUPERUSER CLASS(FACILITY) -**
**ACCESS(READ) ID(***websys1***)**

**PERMIT SUPERUSER.FILESYS CLASS(UNIXPRIV) -**
**ACCESS(UPDATE) ID(***websys1***)**
**PERMIT SUPERUSER.FILESYS.CHOWN CLASS(UNIXPRIV) -**
**ACCESS(READ) ID(***websys1***)**
**PERMIT SUPERUSER.PROCESS.GETPSENT CLASS(UNIXPRIV) -**
**ACCESS(READ) ID(***websys1***)**
**PERMIT SUPERUSER.PROCESS.KILL CLASS(UNIXPRIV) -**
**ACCESS(READ) ID(***websys1***)**

**CONNECT** *websys1* **GROUP(***websrvg1***) OWNER(ADMIN)**
**CONNECT** *websys1* **GROUP(IMWEB) OWNER(ADMIN)**

### G.5.1.3 Surrogate Client IDs

In *Section G.3.4, Client Identification and Authentication*, there is a description of the use and requirements of surrogate IDs. As noted there, surrogate IDs must have limited access authority. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websur1** is a web server surrogate ID.
- Group **websrvg9** is a local server group for surrogate web users.
- ID **websrv1** is the ID of the web server that can act as a surrogate of **websur1**.

The following commands can be used to create the group and a userid and assign the privileges that are required to use a surrogate ID:

**ADDGROUP** *websrvg9* **OMVS(GID(***nnn***)) OWNER(ADMIN)**

**ADDUSER** *websur1* **DFLTGRP(***websrvg9***) OWNER(ADMIN) -**
**NOPASSWORD NOOIDCARD -**
**OMVS(UID(***n***) HOME('/') PROGRAM('/bin/sh'))**

**PERMIT BPX.SRV.***websur1* **CLASS(SURROGAT) ACCESS(READ) ID(***websrv1***)**

### G.5.2 Data Sets

As described in *Section G.3.2, Web Server Data Sets*, there are two groups of vendor product data sets associated with IHS for OS/390 that require protection:

- Distribution data sets named with the prefix SYS1.IMW.AIMW
- Target data sets named with the prefix SYS1.IMW.SIMW.

The following commands can be used to provide the required access control for the MVS data sets:

**ADDSD 'SYS1.IMW.AIMW*' OWNER(SYS1) UACC(NONE)**
**PERMIT 'SYS1.IMW.AIMW*' ACCESS(ALTER) ID(***sysprog-group***)**
**ADDSD 'SYS1.IMW.SIMW*' OWNER(SYS1) UACC(NONE)**
**PERMIT 'SYS1.IMW.SIMW*' ACCESS(ALTER) ID(***sysprog-group***)**

## G.6 TOP SECRET Implementation

This section describes the commands needed to implement the security requirements for IHS for OS/390 under the TOP SECRET ACP. The following task categories are described:

- Started task and other userid definitions
- Data set protection.

### G.6.1 Started Tasks and Other ACIDs

Web server IDs, web server administrator IDs, and optionally surrogate client IDs are defined to enable IHS for OS/390. This section describes commands to define these IDs.

It is possible to create a TOP SECRET User Facility to add an additional level of access control for web servers. Once the Facility is created, it must be added to each user ACID that is allowed access to a web server.

The following command can be used to create the User Facility:

**TSS MODIFY FACILITY(USER*x*=NAME=IMWEB)**

The following consideration applies:

- **IMWEB** is used as a general name to indicate all web servers. The site may choose individual names to allow control of individual servers.

#### G.6.1.1 Web Server IDs

In *Section G.3.1.1, Web Server Identification*, there is a description of the requirements for web server IDs. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websrv1** is the web server ID; it is defined for use as a started task.
- Group **websrvg1** is a local server group for server **websrv1**.
- Group **IMWEB** is defined to use as the group ID in the IBM HFS objects.
- The web server ID is a member of groups **websrvg1** and **IMWEB**.
- Because **websrv1** is not an OS/390 UNIX superuser, the site may wish to update the **ASSIZE**, **OECPUTM**, and **OEFILEP** fields to allow the server to exceed the normal defaults.

The following commands can be used to create the groups and a userid and assign the privileges that are required for a web server:

**TSS CREATE(IMWEB) TYPE(GROUP) NAME('WEB SERVER SOFTWARE')**
**DEPT(***existing-dept***)**
**TSS ADD(IMWEB) GID(205)**
**TSS CREATE(***websrvg1***) TYPE(GROUP) NAME('WEB SERVER 1')**
**DEPT(***existing-dept***)**
**TSS ADD(***websrvg1***) GID(***nnn***)**

**TSS CREATE(***websrv1***) TYPE(USER) NAME('WEB SERVER 1')**
**DEPT(***existing-dept***) FACILITY(STC) PASSWORD(NOPW,0)**
**TSS ADD(***websrv1***) DFLTGRP(***websrvg1***) GROUP(***websrvg1***)**
**TSS ADD(***websrv1***) SOURCE(INTRDR)**
**TSS ADD(***websrv1***) UID(***n***) HOME(/) OMVSPGM(/bin/sh)**
**TSS ADD(***websrv1***) ASSIZE(***n***) OECPUTM(***n***) OEFILEP(***n***)**
**TSS ADD(***websrv1***) GROUP(IMWEB)**
**TSS ADD(***websrv1***) MASTFAC(IMWEB)**
**TSS ADD(STC) PROCNAME(***websrv1***) ACID(***websrv1***)**

**TSS PERMIT(***websrv1***) IBMFAC(BPX.DAEMON) ACCESS(READ)**
**TSS PERMIT(***websrv1***) IBMFAC(BPX.SERVER) ACCESS(UPDATE)**
**TSS PERMIT(***websrv1***) IBMFAC(BPX.SMF) ACCESS(READ)**

For IHS for OS/390 to process SSL connections, the ID associated with the web server must have a digital certificate and key ring. In addition, to be able to authenticate client certificates, the certificates of Certificate Authorities must be available. To accomplish this for systems running IHS for OS/390 Version 5.3 and above, Top Secret is used as the certificate store.

The following commands can be used to insert the certificate, define a key ring, and connect the web server's certificate and those of the Certificate Authorities to the server's key ring.

**TSS ADD(***websrv1***) DIGICERT(CERT01)**
**DCDSN(***certificate-dataset***)**
**LABLCERT('***websrv1***-Cert01') TRUST**
**ICSF** /* Optional if hardware encryption active */

**TSS ADD(***websrv1***) KEYRING(RING01)**
**LABLRING('***websrv1***-Ring01')**

**TSS ADD(***websrv1***) KEYRING(RING01)**
**RINGDATA(***websrv1***.CERT01)**
**DEFAULT**
**TSS ADD(***websrv1***) KEYRING(RING01)**
**RINGDATA(CERTAUTH.***digicert-of-DOD-CLASS-3-Root-CA-Certificate***)**
**TSS ADD(***websrv1***) KEYRING(RING01)**
**RINGDATA(CERTAUTH.***digicert-of-DOD-PKI-Med-Root-CA-Certificate***)**

The following considerations apply:

- The commands that connect the Certificate Authority certificates assume that these certificates have already been defined to TOP SECRET.

- The values in **LABLCERT**, **LABLRING** and **CERTAUTH** operands may include lower case characters.

- If hardware encryption is enabled, the **ICSF** operand may be used.

For IHS for OS/390 to authenticate clients with digital certificates, the ID associated with the server must have access to read the clients' key rings and certificates. Resources in the IBMFAC SAF class control this access.

The following commands can be used to assign the privileges that are required for the web server:

**TSS PERMIT(***websrv1***) IBMFAC(IRR.DIGTCERT.LIST)**
**ACCESS(UPDATE)**
**TSS PERMIT(***websrv1***) IBMFAC(IRR.DIGTCERT.LISTRING)**
**ACCESS(UPDATE)**

If the OS/390 host machine has hardware encryption installed and enabled, resources owned by the ICSF component have been defined. The following PERMIT commands are required to allow the web server and users of the web server to access the ICSF resources. Refer to *Section G.3.9, Secure Sockets Layer (SSL) Configuration*, for more information.

**TSS PERMIT(ALL) CSFSERV(CSFCKI) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFCKM) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFDEC) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFENC) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFPKB) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFPKX) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFPKE) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFPKD) ACCESS(READ)**
**TSS PERMIT(ALL) CSFSERV(CSFPKI) ACCESS(READ)**

Note that these rules allow **all** authenticated users to access the ICSF resources. This is not recommended. Sites should consider more restrictive rules using their specific ID and group structure.

### G.6.1.2 Web Server Administrator IDs

In *Section G.3.1.2, Web Server Administrator Identification*, there is a description of the requirements for web server administrator IDs. As noted there, an administrator ID is a normal userid, supplemented with the special privileges required. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websys1** is a web server administrator's ID.
- Group **webadmg1** is a local server group for server administrators.
- The web server administrator's ID is a member of groups **webadmg1**, **websrvg1**, and **IMWEB**.

The following commands can be used to create the group and a userid and assign the privileges that are required for a web server administrator:

**TSS CREATE(***webadmg1***) TYPE(GROUP) NAME(‘***WEB SYS ADMIN 1***’)**
**DEPT(***existing-dept***)**
**TSS ADD(***webadmg1***) GID(***nnn***)**

**TSS CREATE(***websys1***) TYPE(USER) NAME(‘***WEB SYS ADMIN 1***’)**
**DEPT(***existing-dept***) PASSWORD(***password***,90,EXP)**
**TSS ADD(***websys1***) DFLTGRP(***webadmg1***) GROUP(***webadmg1***)**
**TSS ADD(***websys1***) UID(***n***) HOME(/u/***websys1***) OMVSPGM(/bin/sh)**
**TSS ADD(***websys1***) GROUP(***websrvg1***)**
**TSS ADD(***websys1***) GROUP(IMWEB)**
**TSS ADD(***websys1***) FAC(IMWEB)**

**TSS PERMIT(***websys1***) IBMFAC(BPX.FILEATTR.APF) ACCESS(READ)**
**TSS PERMIT(***websys1***) IBMFAC(BPX.FILEATTR.PROGCTL) ACCESS(READ)**
**TSS PERMIT(***websys1***) IBMFAC(BPX.SUPERUSER) ACCESS(READ)**
**TSS PERMIT(***websys1***) UNIXPRIV(SUPERUSER.FILESYS)**
**ACCESS(CONTROL)**
**TSS PERMIT(***websys1***) UNIXPRIV(SUPERUSER.FILESYS.CHOWN)**
**ACCESS(READ)**
**TSS PERMIT(***websys1***) UNIXPRIV(SUPERUSER.PROCESS.GETPSENT)**
**ACCESS(READ)**
**TSS PERMIT(***websys1***) UNIXPRIV(SUPERUSER.PROCESS.KILL)**
**ACCESS(READ)**

### G.6.1.3 Surrogate Client IDs

In *Section G.3.4, Client Identification and Authentication*, there is a description of the use and requirements of surrogate IDs. As noted there, surrogate IDs must have limited access authority. This section describes the ACP-specific implementation of those requirements. The following conventions are used:

- ID **websur1** is a web server surrogate ID.
- Group **websrvg9** is a local server group for surrogate web users.
- ID **websrv1** is the ID of the web server that can act as a surrogate of **websur1**.

The following commands can be used to create the group and a userid and assign the privileges that are required to use a surrogate ID:

**TSS CREATE(***websrvg9***) TYPE(GROUP) NAME(‘WEB SERVER 9’)**
**DEPT(***existing-dept***)**
**TSS ADD(***websrvg9***) GID(***nnn***)**

**TSS CREATE(***websur1***) TYPE(USER) NAME('WEB SURROGATE 1')**
**DEPT(***existing-dept***) PASSWORD(NOPW,0)**
**TSS ADD(***websur1***) DFLTGRP(***websrvg9***) GROUP(***websrvg9***)**
**TSS ADD(***websur1***) UID(***n***) HOME(/u/***websur1***) OMVSPGM(/bin/sh)**
**TSS ADD(***websur1***) FAC(IMWEB)**

**TSS PERMIT(***websrv1***) SURROGAT(BPX.SRV.***websur1***) ACCESS(READ)**

### G.6.2 Data Sets

As described in *Section G.3.2, Web Server Data Sets*, there are two groups of vendor product data sets associated with IHS for OS/390 that require protection:

- Distribution data sets named with the prefix SYS1.IMW.AIMW
- Target data sets named with the prefix SYS1.IMW.SIMW.

The following commands can be used to provide the required access control for the MVS data sets:

**TSS ADD(SYS1) DSN(SYS1.IMW.AIMW-)**
**TSS PERMIT(***sysprog-group***) DSN(SYS1.IMW.AIMW-) ACCESS(ALL)**
**TSS ADD(SYS1) DSN(SYS1.IMW.SIMW-)**
**TSS PERMIT(***sysprog-group***) DSN(SYS1.IMW.SIMW-) ACCESS(ALL)**

# APPENDIX H. GUIDELINES FOR SOFTWARE REVIEW OF VENDOR-PROVIDED PROGRAMS AND SCRIPTS

## H.1 Purpose

DOD faces multiple threats to the systems it develops and hosts. A primary threat exists in the software running on its distributed servers within a complex environment. Security can be compromised through unintended weaknesses in software or intentionally programmed malicious code. Within the DOD, Web environment applications are examined prior to deployment to mitigate the risk introduced by internally developed code, open source code, or public domain code. This appendix describes a recommended review process, which may be adapted by local CCBs.

## H.2 Major Risk Areas

1. Changes in content on web pages

2. Release of system information that may increase the opportunity of a malicious individual to compromise the system

3. Denial of Access through exhausting system resources, overflowing buffers, disk space, RAM, or bandwidth

4. Inhibiting associated system resources and applications through disruption of those elements

5. Commandeering the system to be used for other purposes (e.g., the use of the system as a "trusted" system to access other systems)

6. Collection and release of personal and/or privacy information

7. Transmission of malicious code to clients

8. Backdoors

## H.3 Overview

The above risk areas can be exposed through various means. A security reviewer must skeptically approach all software in an attempt to determine the risk to which the software might expose systems and services. Software written by trusted individuals or organizations is no less likely to contain security risks than open source or public domain code.

A section of code may have malicious code embedded within an application. CGI applications invoked by the web server, generally inherit the environment and permissions of the web server user. That means if a file can be written by the server user, a CGI application can also modify files. A CGI application will have access to all local data areas without issue of access controls that might be established. Thus a malicious programmer could embed code into a CGI to scan all web data areas on a site regardless of access control. For this reason, all input/output statements (opens, reads, and writes) should be documented and verified.

A CGI application may have the ability to write files to disk, log files, data files, etc. Again, a malicious programmer could cripple parts of a system by generating and writing gigabytes of meaningless data to disk, and exhaust system partitions. For this reason, it is necessary to ensure that all log files, and other files are written to specified areas, and never to the root partition, or system application areas.

A CGI application generally accepts input from user forms and URL strings. A malicious programmer can use an unexpected input strings to trigger malicious code. All input strings will be checked to ensure that the input string matches the expected data. For example a date should match a preset string "**date=00/00/00**" when entered to execute a function.

A non-CGI applications running via a CRON could do a simple system time check and execute a function each even day of the month between the times of 11 p.m. to 5 a.m. when no one might be monitoring traffic. A CRON application run by root, by default runs with wide-open permissions, having access to all system resources. A CRON application run by the web server user has access to web server resources. This holds true for the full text search user or the Oracle users. Thus, each application must be evaluated to ensure that it is using only the resources required to accomplish its tasks. A random unexplained socket call, the opening of a PIPE, or sending an e-mail to an unknown destination all must be examined and evaluated during a security review.

A backdoor is a way for a programmer to easily obtain access to a program to allow debugging. It might be that a program has a form of access control enabled, and the programmer hardcodes the username and password "admin/debug" to simplify maintenance. This procedure is very high risk and such code should not pass security review. COTS products often do this very thing. Backdoors can also be used for malicious purposes. A programmer might place code with a backdoor into the public domain, and later use that backdoor to compromise the systems of adapters. Applications should be examined for the presence of backdoors, and in those cases the facts must be recorded, the risks estimated, and management must determine the acceptability of the risk. For instance, until a recent patch to the Netscape Apple application, WebObjects, the software that enabled a password on application statistics was broken, and thus no password could be set. This meant that anyone could obtain application statistics.

A server side application has the ability to send JavaScript, Java applets, and other mobile code to client browsers. It is possible for a malicious programmer to output malicious Java applets, and/or JavaScript. While the JavaScript language and Java language have safeguards, it is possible that previously identified bugs and security holes in early browsers might be exploited through server software targeting users who have not upgraded. Developers can prevent their sites from being abused by ensuring that dynamically generated pages do not contain undesired tags. In addition, web pages should explicitly set a character set to an appropriate value in all dynamically generated pages. Finally, all programs (scripts) will have absolute paths set.

## H.4 General Guidance

When reviewing code, assume the worst. If something is not documented or is suspicious, take extra care to review the code. If the code is confusing or complicated, request additional help; do not approve something you do not understand. You should never conduct a code review on a program written in a language that is not familiar.

Security can be compromised through intentional and accidental events. Poor programming is more likely to cause a security issue than an intentional Trojan horse or backdoor. Third-party free or open source software must be assumed to contain back doors, Trojan horses, and poorly written code.

## H.5 Conclusion

It is critical to report the discovery of any malicious code to the security officer. Further, it is important to document findings where a programmer on staff has erroneously coded an application in a way that might cause a security risk. The issues should be reviewed with the programmer and with the development staff.

This page is intentionally left blank.

# APPENDIX I. LIST OF ACRONYMS

| | |
|---|---|
| ACF2 | Access Control Facility 2 |
| ACL | Access Control List |
| ACP | Access Control Program |
| AFSSI | Air Force System Security Instruction |
| AFSSM | Air Force System Security Memorandum |
| AIS | Automated Information Systems |
| APAR | Authorized Program Analysis Record |
| API | Application Program Interface |
| AR | Army Regulation |
| | |
| C&A | Certification and Accreditation |
| CA | Certification Authority |
| CCB | Configuration Control Board |
| CGI | Common Gateway Interface |
| CINC | Commander-in-Chief |
| CIO | Chief Information Officer |
| CM | Configuration Management |
| COAST | Computer Operations, Audit, and Security Technology |
| COE | Common Operating Environment |
| COMPUSEC | Computer Security |
| COOP | Continuity of Operations Plan |
| COPS | Computer Oracle and Password System |
| COTS | Commercial Off-The-Shelf |
| | |
| DCTF | DISA Continuity of Operations and Test Facility |
| DECC | Defense Enterprise Computing Center (was Defense Megacenter [DMC]) |
| DECC-D | Defense Enterprise Computing Center - Detachment |
| DES | Digital Encryption Standard |
| DII | Defense Information Infrastructure |
| DISA | Defense Information Systems Agency |
| DISAI | Defense Information Systems Agency Instruction |
| DISN | Defense Information System Network |
| DLAR | Defense Logistics Agency Regulation |
| DLL | Dynamic Link Library |
| DNS | Domain Name Service |
| DOD | Department of Defense |
| DOD-CERT | Department of Defense Computer Emergency Response Team (was ASSIST) |
| DSS | Digital Signature Standard |
| DTIC | Defense Technical Information Center |
| | |
| E-mail | Electronic Mail |
| ESM | Enterprise Security Manager |
| | |
| FRCA | Fast Response Cache Accelerator |
| FSO | Field Security Operations |
| FTP | File Transfer Protocol |
| | |
| GID | Group ID |
| GNOSC | Global Network Operations and Security Center |

| | |
|---|---|
| GWAPI | Go Webserver Application Programming Interface |
| HFS | Hierarchical File System |
| HTML | Hyper Text Markup Language |
| HTTP | Hyper Text Transport Protocol |
| I&A | Identification and Authentication |
| IAM | Information Assurance Manager |
| IAO | Information Assurance Officer |
| IAVA | Information Assurance Vulnerability Alert |
| IAVM | Information Assurance Vulnerability Management |
| IAW | In Accordance With |
| IHS | IBM HTTP Server |
| IIS | Internet Information Server |
| INFOSEC | Information Security |
| INFOWAR | Information Warfare |
| IP | Internet Protocol |
| IS | Information System |
| ISP | Internet Service Provider |
| ISSM | Information Systems Security Manager |
| ISSO | Information Systems Security Officer |
| ITA | Intruder Alert |
| JAVA | A programming language |
| JCL | Job Control Language |
| JDK | Java Development Kit |
| LAN | Local Area Network |
| LCC | Local Control Center |
| LDAP | Lightweight Directory Access Protocol |
| LPAR | Logical Partition |
| LRA | Limited Registration Authority |
| MIME | Multi-purpose Internet Mail Extension |
| MMC | Microsoft Management Console |
| MTA | Message Transfer Agent |
| MVS | Multiple Virtual Storage |
| NAVSO | Navy Staff Office Publication |
| NCSA | National Computer Security Agency |
| NCSC | National Computer Security Center |
| NFS | Network File System |
| NIC | Network Information Center |
| NIPRNet | Non-classified (but Sensitive) Internet Protocol Routing Network |
| NIS | Network Information Services |
| NIST | National Institute of Standards and Technology |
| NNTP | Network News Transfer Protocol |
| NOSC | Network Operations and Security Center |
| NSA | National Security Agency |
| NSAPI | Netscape Server Application Program Interface |
| NSO | Network Security Officer |

| | |
|---|---|
| NT | Microsoft Networking Operating System |
| OS | Operating System |
| PAO | Public Affairs Officer |
| PC | Personal Computer |
| PERL | Practical Extraction and Report Language |
| PHP | An HTML preprocessor scripting language |
| PKI | Private/Public Key Infrastructure |
| PM | Program Manager |
| POC | Point of Contact |
| RACF | Resource Access Control Facility |
| REXX | Restructured eXtended eXecutor |
| RISSC | Regional Information System Security Cell |
| RNOSC | Regional Network Operations and Security Center (formerly ROSC) |
| ROSC | Regional Operations Security Center |
| SA | System Administrator |
| SAF | System Authorization Facility |
| SBU | Sensitive but Unclassified |
| SECNAVINST | Secretary of the Navy Instruction |
| SIPRNet | Secret Internet Protocol Router Network |
| SLA | Service Level Agreement |
| SM | Security Manager |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| SOP | Standard Operating Procedure |
| SRR | Security Readiness Review |
| SSH | Secure Shell |
| SSL | Secure Socket Layer |
| SSO | Systems Support Office |
| STIG | Security Technical Implementation Guide |
| TASO | Terminal Area Security Officer |
| TCB | Trusted Computing Base |
| TCP | Transmission Control Protocol |
| TFS | Temporary File System |
| TFTP | Trivial File Transfer Protocol |
| UDDI | Universal Description, Discovery and Integration |
| UID | Userid |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| VAAP | Vulnerability Analysis and Assistance Program |
| VCTS | Vulnerability Compliance Tracking System |
| VIS | Vendor Integrity Statement |
| VMS | Vulnerability Management System |
| WebSphere | IBM Application Development Environment |

| | |
|---|---|
| WESTHEM | Western Hemisphere |
| WSDL | Web Service Description Language |
| WWW | World Wide Web |
| XML | eXtensible Markup Language |