APPLICATION SERVICES

SECURITY TECHNICAL IMPLEMENTATION GUIDE

Version 1, Release 1

17 January 2006

Developed by DISA for the DOD

This page is intentionally left blank.

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

This page is intentionally left blank.

1. INTRODUCTION

1.1 Background

This Application Services Security Technical Implementation Guide (STIG) provides security configuration and implementation guidance for application server products designed to the Java™ 2 Platform, Enterprise Edition (J2EE™). J2EE defines a standard security framework of configuration and implementation for the protection of application servers.

The J2EE platform is a superset of the Java 2 platform. It is specification that provides enhanced security mechanisms for authentication, authorization, and auditing.

Section 2, Application Services Overview, provides a generic description of the elements characteristic of most application server products. *Section 3, Application Services Security*, provides general guidance for all application server products. Specific commercial and open source application server products are addressed in separate appendices. This STIG is intended for use in conjunction with other STIGs developed by the Defense Information Systems Agency (DISA). The operating system (OS) STIGs provide crucial guidance for securing the platforms on which application servers run. Security requirements for Database Management Systems (DBMS) and web servers utilized by application servers are addressed in the *Database STIG* and *Web Server STIG*.

1.2 Authority

Department of Defense (DOD) Directive 8500.1 requires that "all Information Assurance (IA) and IA-enabled Information Technology (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, National Security Agency (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 Mission Assurance Category (MAC) II Sensitive level except where explicitly specified.

1.3 Scope

This STIG describes security requirements to be applied to application server products and their hosted web applications and services used in DOD environments. This STIG assigns responsibilities to the Application Server Administrator (ASA). The ASA is assigned the privileges, access, and responsibility to configure and maintain the security and operation of an application server. The information in this STIG is designed to assist ASAs with the creation of a more secure application server configuration and implementation. Application of these requirements is intended to provide a minimum level of assurance. This STIG provides both general and product-specific security guidance. Vendor implementation of application server functions varies, as most open source and commercial products provide only subsets of all the

available security-related functions and configurations defined in the J2EE framework specification.

J2EE application servers require the use of a Java Virtual Machine (JVM) for execution. The JVM provides a layer of abstraction between the application server and the underlying hardware platform and OS. The JVM is central to an application server's portability, because application servers run on the JVM, independent of whatever may be underneath a particular JVM implementation. This STIG provides guidance for securing application servers as they relate to the JVM. Security configuration specific to JVMs is addressed in the *Web Server STIG* and the *Desktop Application STIG*.

This STIG does not address security of applications developed to use the services of the application server. Compliance with security requirements for applications may be measured by use of the *Application Security Checklist*.

Specific guidance is provided for the following application servers and virtual machines:

1.3.1 Apache Jakarta Tomcat

Discussion and requirements for Apache Jakarta Tomcat included in Appendix B refer to the versions and release levels of Tomcat and the corresponding Servlet and JSP Specifications listed in the following table:

Apache Jakarta Tomcat	Servlet/ JSP Specification
5.5.9	2.4/ 2.0
5.0.28	
4.1.31	2.3/ 1.2

Table 1-1. Apache Jakarta Tomcat

1.3.2 BEA WebLogic Server

Appendix C, BEA's WebLogic Server refers to the versions and release levels with service packs (SP) listed in the following table:

BEA WebLogic Server
8.1 SP4
7.0 SP6

Table 1-2. BEA WebLogic Server

1.3.3 Sun Microsystems JVM

Discussion and requirements of the JVM refer to the versions and release levels of the Sun Microsystems JVM for the Java Runtime Environment (JRE) and the Java Development Kit (JDK) or the Software Development Kit (SDK) using the Java 2 Platform, Standard Edition (J2SE) specified in the following table:

JRE	JDK
JRE 5.0 Update 4	JDK 5.0 Update 4
J2SE v1.4.2_09 JRE	J2SE v1.4.2_09 SDK

Table 1-3. Sun Microsystems JVM

1.3.4 Microsoft .NET Framework

The Microsoft .NET framework is the framework specification for the serving of applications on Microsoft Windows OS platforms. This STIG does not address security of the Microsoft .NET framework. To secure the Microsoft .NET framework and supported applications, refer to the NSA Guide, *NSA Guide to Microsoft .NET Framework Security, Version 1.4, dated 22 September 2004.*

1.3.5 Other Application Servers

There are many application servers available both commercially and open source. This STIG does not address all the varieties of application servers that exist (i.e. JBoss, IBM WebSphere, Sun Java System Application Server, Oracle Application Server, etc.) It is applicable to use the general section of this STIG, section 3, to help secure those application servers not specifically covered in the appendices. The general section of this STIG includes security guidance that applies to any J2EE compliant application server.

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**," indicate 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 makes all "**will**" statements easier to locate and interpret from the context of the topic. The Information Assurance Officer (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**" indicates 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.

Table 1-4. Vulnerability Severity Code Definitions

1.6 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 Non-classified Internet Protocol Router Network (NIPRNet) Uniform Resource Locator (URL) for the IASE site is http://iase.disa.mil/.

1.8 Document Revisions

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

2. APPLICATION SERVICES OVERVIEW

2.1 General Overview

The terms *Application Server* and *Application Service* are often used interchangeably. This STIG uses the term application service to refer to the complete mission or function of an Enterprise Information System's (EIS). One or more application servers may support a single application service. An application server is a component-based product that resides in the application tier of a server-centric architecture. It provides middleware services for security and state maintenance, along with data access and persistence. The term *application server* refers to a single computer that may serve in conjunction with other computers in the network as part of an EIS that provides a specific application service.

Figure 2-1 depicts the architecture of a typical EIS.

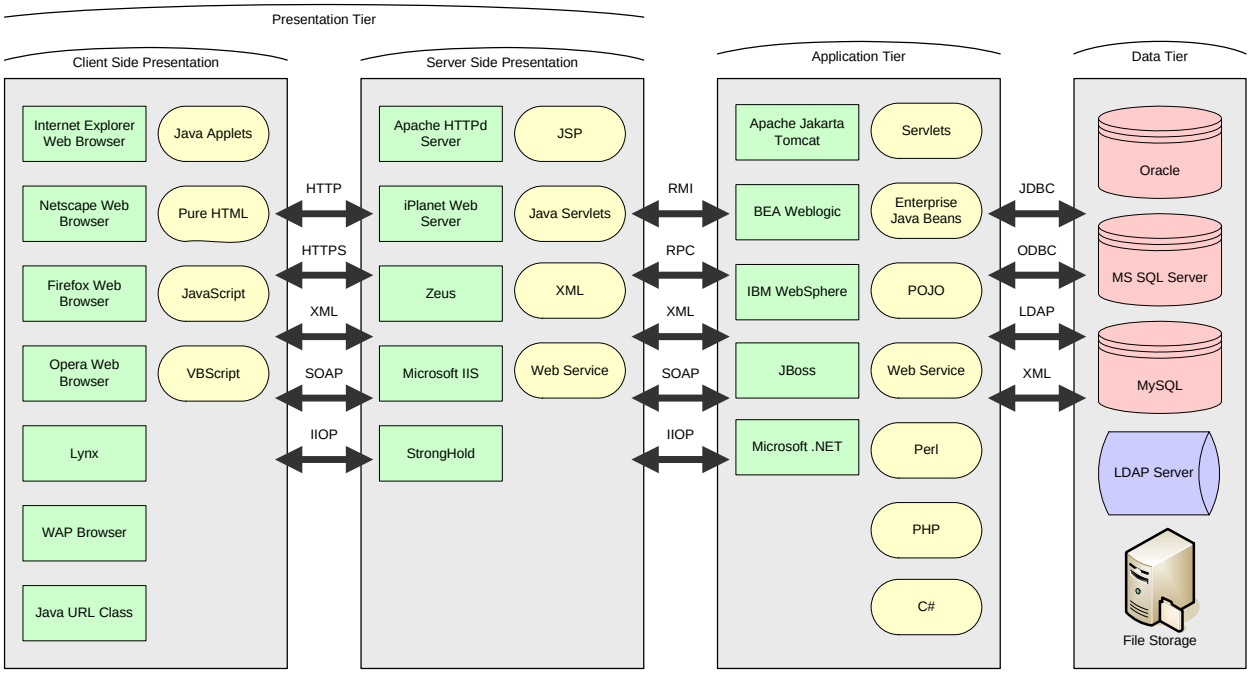

Figure 2-1. Enterprise Information System

Two presentation tiers are responsible for presenting the data to an end user or system. The server side presentation tier includes the web server that delivers the data to the web browser in the client side presentation tier. The web browser renders the data into a common format that is displayed to the user. The web browser also allows the user to interact with the application tier by accepting input from the user that is returned to the web server and then passed along to the application server. The server side presentation tier includes web servers like Apache and Internet Information Server (IIS). The client side presentation tier includes web browsers like Internet Explorer and Netscape Navigator. The client side presentation tier may also include application components that create the page layout.

The application tier is known as the engine of a web application. It executes business logic in response to user instructions and obtains data for display in the presentation tier. The application tier houses the application servers. The application tier also supports applications written in or utilizing technologies such as the Common Gateway Interface (CGI), Java, Microsoft's .NET services, Hypertext Preprocessor (PHP), Cold Fusion, etc.

The data tier provides storage for both temporary and permanent data accessed by the application server and the applications it serves. Some application servers store Extensible Markup Language (XML) formatted data in local or remote flat files, but most systems utilize a DBMS such as Oracle and MS SQL Server for data storage.

An application server allows a single set of security functions to be defined and used by several different applications. The centralization and sharing of a single set of security functions decreases the complexity and potential inconsistency of managing a multitude of security functions for each individual application. The application server design should be used when security, scalability, and cost are major considerations.

2.2 J2EE Overview

The J2EE framework specifications are developed and maintained under the Java Community Process (JCP). The JCP is composed of representatives from various companies and organizations that support the continued evolution of the J2EE framework specification in cooperation with the entire Java developer community. Sun Microsystems, the creator of the Java programming language, is the maintainer of the J2EE specification upon which Java is based.

2.3 JVM

This STIG will focus on examples and specific product technologies related to Sun Microsystems JVM, but most of these also apply to any J2EE-compliant JVM.

The JVM is an abstract computer that executes compiled Java programs. The JVM is a software implementation executing on top of a hardware platform and OS. All Java programs are compiled for the JVM. The JVM is, in turn, implemented on a specific platform before the compiled Java programs can run on that platform.

The JVM is what makes J2EE application servers portable. It provides the abstraction between a compiled Java program and the underlying hardware platform and OS. There are many vendors that provide their own JVM implementations that adhere to the J2EE framework specification. J2EE application servers run on a JVM, thereby making them platform independent.

3. APPLICATION SERVICES SECURITY

This section is broken into subsections that align with the IA Controls subject areas defined in DOD Instruction 8500.2, Information Assurance (IA) Implementation. These subject areas are as follows:

- Continuity
- Security Design and Configuration
- Enclave Boundary Defense
- Enclave and Computing Environment
- Identification and Authentication

Some of these areas are further divided to provide a more cohesive presentation. The Personnel subject area and the Physical and Environmental subject area are not included as there are no controls in those subject areas that are addressed in an application services product security readiness review.

3.1 Continuity

3.1.1 Data Backup Procedures

Backups of the application server data and its hosted web applications are critical in order to recover from ordinary hardware problems, unexpected software errors, or a major computing facility event.

Backups will be made on the application server data, software, configuration files, audit logs and the critical application components. In default application server installations this entails backing up the entire directory tree of the application server. Since some of these items are frequently modified, backups are critical to quickly and efficiently restore systems in case of failure.

Backup procedures will encompass the necessary requirements for availability based on the MAC level of the system. For MAC III systems it is necessary to ensure that backups are performed weekly. For MAC II systems backups are performed daily and the recovery media is stored off-site in a protected facility in accordance with its mission assurance category and confidentiality level. In MAC I systems backups are maintained through a redundant secondary system, not collocated, and can be activated without loss of data or disruption to the operation.

- *(N/A: CAT II) The ASA will ensure the application server backups are performed in accordance with the assigned MAC level.*

3.1.2 Disaster and Recovery Planning

Procedures for disaster recovery should include business recovery plans, system contingency plans, facility recovery plans, and the plan acceptance as well. Similar to the backup plans, disaster recovery criticality is dependant on the mission assurance category of the system. For MAC III systems a disaster recovery plan provides for the resumption of mission or business

essential functions within five days of activation and within 24 hours for MAC II systems. In MAC I systems a plan must provide for the smooth transfer of all mission or business essential functions to an alternate site for the duration of an event with little or no loss of operational continuity. MAC I application servers will require fault tolerant, redundant servers utilizing some form of clustering and/ or failover configurations. This also must include the support components that comprise the complete application service, to include backend systems such as databases, Lightweight Directory Access Protocol (LDAP) servers, and other supporting technologies.

- *(APS0730: CAT II) The IAO will ensure a disaster recovery plan for an application server exists and includes appropriate provisioning for the continuity and contingency measures to maintain operational status for the complete application server.*

  - *For applications servers serving MAC I systems, transfer to an alternate site for the duration of an event with little or no loss of operational continuity.*
  - *For applications servers serving MAC II systems, resumption within 24 hours of activation.*
  - *For applications servers serving MAC III systems, resumption within five days of activation.*

3.2 Security Design and Configuration

3.2.1 Application Server Administration

The security of an application server depends upon the restriction of administrative privileges to authorized individuals. Indiscriminate assignment of administrative privileges leads to a lack of accountability and configuration control. The IAO will authorize all users assigned application server administrative privileges and monitor these privilege assignments to maintain their proper assignment. Additionally, the IAO will ensure that an administrator is assigned for each application server.

- *(N/A: CAT III) The IAO will ensure an ASA is appointed for each application server.*

- *(N/A: CAT III) The Information Assurance Manager (IAM) will ensure all accounts with assigned ASA privileges are authorized and documented.*

- *(N/A: CAT II) The IAO will ensure an ASA role is defined with least privilege.*

3.2.2 Application Server Content

The organization or activity that sponsors an application server will have content responsibility. These persons will ensure all information is kept current and all content placed on the application server is reviewed and approved by a Configuration Management (CM) authority.

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

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

NOTE: This document may be accessed via http://iase.disa.mil

3.2.3 Functional Architecture for Automated Information System (AIS) Applications

Unidentified application interfaces to external systems may lead to misassigned liability for untrusted information or disclosure of sensitive information to untrusted sources. Unprotected interfaces may lead to the transfer of sensitive information transmitted on un-trusted networks with un-trusted sources.

All interfaces will be documented, labeled, and users notified of any data transmitted by the application server to or from external or un-trusted systems. Maintain documentation, justification, and approval for all application server interfaces to external systems, or systems under the administration and control of others. Identify which application server interfaces connect to trusted DOD systems (certified and installed on a DOD network), un-trusted DOD systems (certification unknown, but installed on a DOD network), un-trusted non-DOD systems, and trusted non-DOD systems (outsourced DOD services where the vendor/provider has provided some level of assurance).

Externally connected non-DOD resources contained within DOD application servers should be reviewed periodically to ensure their continued suitability. If the content of an externally connected non-DOD resource is no longer required or approved, the reference or content must be removed. When externally connected resources to non-government sources are included, the head of the DOD Component, or the subordinate organization, is responsible for ensuring that a disclaimer is made that neither the DOD nor the organization endorses the product or organization at the destination, nor does the DOD exercise any responsibility over the content at the destination.

- *(N/A: CAT II) The ASA will ensure all application interfaces to external systems are identified and protected.*

- *(N/A: CAT II) The ASA will ensure all web hyperlinks that connect to external resources are approved.*

- *(N/A: CAT II) The ASA will ensure all web hyperlinks that connect to external resources identify the source of origin.*

Although technically possible, the use of applications servers on systems operating at different classification levels is not permitted. Application servers serving data of differing sensitivity to audiences with differing need-to-know requirements may be used. Such installations should be

configured in accordance with guidance provided in the *Web Content Access Control White Paper*.

- *(N/A: CAT II) The IAO will ensure application servers serving data of different classification levels are not installed on a shared host*

- *.(N/A: CAT II) The IAO will ensure application servers on a shared host serving data to audiences with different access control requirements are configured to support the greatest separation of access control possible.*

3.2.4 Mobile Code Technologies

Mobile code served by J2EE application servers is designated as Category 2 mobile code technology in the DOD Memorandum, Policy Guidance for Use of Mobile Code Technologies in DOD Information Systems. According to the Memorandum, Category 2 mobile code is allowed within DOD if it is signed with a DOD-approved PKI code-signing certificate or it meets the requirements for use of unsigned code.

- *(N/A: CAT I) The ASA will ensure mobile code delivered by the application server is signed with a DOD-approved PKI code-signing certificate or it meets the requirements for use of unsigned code.*

3.2.5 Partitioning the Application

Many application servers use external data repositories to store I&A data. Application servers support the separation of repository data from the application server software libraries and content. At a minimum, the storage of data must be located in a separate file directory from the application server software libraries and content. However, it is recommended that the data reside on a separate host for maximum security. Separation provides a more controlled access path to the data and reduces the risk of resource contention.
The *Database STIG* recommends that any DBMS be installed on a host system dedicated to its support. By separating the DBMS server, access to that platform can be more finely controlled, resulting in reduced exposure to vulnerabilities in the DBMS software. DODI 8500.2 requires a physical or logical separation of user interface services from data storage and management services.

- *(N/A: CAT II) The ASA will separate the repository data from the application server data and its content.*

- *(N/A: CAT II) At a minimum, the ASA will ensure the repository data is in a separate directory hierarchy from the application server data and its content.*

3.2.6 Ports, Protocols, and Services

The requirements in this section are intended to ensure that the combination of ports, protocols, and services (PPS) used by an application server and its hosted web applications are consistent with secure practices identified in DOD network security guidance, including *DOD Instruction*

8551.1, Ports, Protocols, and Services Management (PPSM) and the associated *Ports, Protocols, and Services (PPS) Assurance Category Assignments List* (CAL*).* The PPS Assurance CAL, provides detailed guidance for specific ports, protocols, and services. The following are extracts of the PPS usage principles from the PPSM instruction:

- PPS are assessed for vulnerabilities and assigned to one of three assurance categories: RED, YELLOW, or GREEN.

- PPS designated as RED has a low level of assurance. These PPS implemented in applications expose DOD networks to an unacceptable level of risk for routine use. A RED PPS will only be allowed when approved by the Defense Information Systems Network (DISN) DAAs for a specific DOD information system under defined conditions and restrictions and if no suitable alternative exists.

- PPS designated as YELLOW have a medium level of assurance. These PPS expose DOD networks to an acceptable level of risk for routine use only when implemented with the required mitigation strategy and approved by the DISN DAAs for a specific DOD information system.
- PPS designated as GREEN have a high level of assurance. These PPS are considered best security practices and are recommended for use when implemented with the required mitigation strategy and approved by the DISN DAAs for a specific DOD information system.

- *(N/A: CAT II) The IAO will ensure any PPSM-designated RED or YELLOW ports and designation boundaries utilized by the application server are approved by the DISN DAAs.*

- *(EN810: CAT III) The ASA will ensure only ports and protocols approved by the DOD Ports, Protocols, and Services Assurance Category Assignments List are used.*

- *(EN820: CAT III) The ASA will ensure the use of new protocols or ports by the application server and its hosted web applications are submitted to the appropriate approving authority for that organization, which in turn are submitted through the Ports, Protocols, and Services Management (PPSMP).*

- *(EN830: CAT III) The ASA will ensure protocols do not use random ports or non-fixed port numbers by the application server and its hosted web applications. Instead, static port allocation should be used to avoid proliferation of possible vulnerabilities.*

- *(EN840: CAT III) The ASA will not modify any IP services and will ensure the application server and its hosted web applications are compliant to the relevant Request For Comments (RFC) standard.*

3.2.7 Supported Software and Patch Requirements

To protect the integrity of the application sever and its environment, the IAO will ensure that the application server version is a supported product version. Supported product versions are those

that continue to receive security updates and are actively maintained by the manufacturer in response to the discovery of vulnerabilities and defects. The ASA will ensure that the application server patch level, security upgrades, and service packs are current.

- *(N/A: CAT I) The IAO will ensure unsupported application server software is removed or upgraded prior to a manufacturer dropping support.*

- *(N/A: CAT II) The IAO will ensure the site has a formal migration plan for removing or upgrading an application server prior to the date the manufacturer drops security patch support.*

- *(N/A: CAT I) The IAO will ensure the application server version has all patches, security upgrades, and SPs applied.*

3.2.8 Software Baseline

A baseline is an inventory of all application server software and component software. Baseline control consists of comparing a current application server software snapshot with a previously determined valid snapshot of the application server software. The purpose of maintaining and checking a baseline is to detect unauthorized, undocumented changes. Unauthorized changes may indicate compromise and, if detected early, could prevent serious damage. A baseline consists of a file list to include size, access permissions, modification times, checksums, etc. of the application server's software directories. The ASA should maintain three weeks of baseline reports and be able to provide them upon request. The ASA should ensure that all baseline backups are maintained on write-protected media.

- *(N/A: CAT II) The IAM will ensure the Configuration Control Board (CCB) maintains a current and comprehensive baseline inventory that includes all application server software.*

- *(N/A: CAT II) The IAM will ensure a current and comprehensive baseline inventory that includes all application server software is maintained as part of the Certification and Accreditation (C&A) documentation.*

- *(N/A: CAT II) The IAO will ensure a copy of the application server software inventory is stored in a fire-rated container or otherwise not collocated with the original.*

3.3 Enclave Boundary Defense

These requirements address remote access to application servers and their hosted web applications. For the purpose of this STIG, remote access is described as any access to an application server or its hosted web applications from a host outside of the enclave in which the host resides.

Enclave boundary defenses protect inside data and services from outside dangers. All remote access to sensitive or classified data on DOD information systems, including both privileged and unprivileged access, requires a restricted access path that includes encryption and strong

authentication. The following requirements address those security measures for any remote application server access.

- *(N/A: CAT II) The IAO will ensure remote access to the application server and its hosted web applications is secured with the following:*

  - *Use of a managed access control point such as a remote access server in a Demilitarized Zone (DMZ).*
  - *Sensitive data uses National Institute of Standards and Technology (NIST) FIPS 140-2 validated cryptography.*
  - *Classified data uses NSA-approved cryptography.*

A discussion of privileged user remote access can be found in the *Enclave STIG*. Because of the potential for serious impact from the compromise of privileged access, additional security for sessions and special attention to auditing to those sessions is necessary.

The following requirements address the need to protect remote ASA access and to review use of that access because it represents remote user privileged access.

- *(N/A: CAT II) The IAO will ensure that remote access for application server administration uses:*

  - *Session security measures such as a Virtual Private Network (VPN) configured in blocking mode to discard all but authorized traffic.*
  - *A process that creates an audit log for each remote session.*

- *(N/A: CAT II) The IAM/IAO will review the audit log for every remote session of an ASA.*

While VPN implementations do provide desirable session protection, they can also be used to conceal malicious traffic. An intrusion detection system (IDS) is a tool to address this risk. A discussion of various IDS types and functions can be found in the *Enclave STIG*. The requirement in this document enforces the specific need for remote application server administrative traffic that uses a VPN to be examined for intrusive behavior.

- *(N/A: CAT II) The IAO will ensure that VPN traffic for remote ASA sessions is visible to an IDS.*

3.4 Enclave and Computing Environment

This section describes the specific considerations and requirements for application servers that are related to the IA Controls in the Enclave and Computing Environment subject area. Because of the extent of the information in this area, it is organized in the following subsections:

- Data protection
- Account management
- Auditing

- Remote Access

3.4.1 Data Protection

Application servers, their hosted web applications, and associated repository data need to be protected at the same confidentiality level as the systems on which that data is disseminated. This reflects the fact that unauthorized access to this data might have several negative effects:

- Invalid data, used as the basis for configuration changes or corrective action, could result in a loss of availability.

- Specific configuration data might expose a subject system to specific attacks based on known vulnerabilities.

- The loss of auditing data for application server events could make it impossible to identify the source of malicious activity.

The following requirements ensure access to an application server's data is appropriately restricted and the access attempt is recorded for subsequent review if needed.

- *(N/A: CAT II) The IAO will ensure that access to application server configuration, audit, and operational data is limited so that only authorized application server processes and users can read or update it.*

- *(N/A: CAT II) The ASA will ensure that application server actions that access or change application server configuration, audit, and operational data are logged and that the logs are reviewed periodically or immediately upon suspicious and suspect security events.*

It should be noted that the requirement for data access and change logging might be satisfied using OS facilities as well as the application server functions.

The application server can itself be sensitive or classified if it serves information that is not for general public consumption.

As data used in conjunction with the application server and its hosted web applications traverses networks, it must be protected from disclosure according to its confidentiality level and in accordance with need-to-know requirements. In this case protection is mandated in the form of data encryption.

- *(N/A: CAT II) The ASA will ensure that the application server is configured to support or require at a minimum NIST-validated cryptography when serving data that includes unclassified, sensitive information (including system hardware or software configuration data) that traverses a commercial or wireless network.*

It is necessary to implement cryptographic protection for certain types of application servers. Encryption, key exchange, digital signature, and hash algorithms are used in various cryptographic services to provide this protection. Proprietary or less robust commercial

algorithms cannot be used because their level of protection may be too weak. Product implementations that have not been appropriately evaluated and validated might not provide the intended protection. The use of NIST-validated implementations ensures the appropriate strength and correct implementation of a cryptographic service.

Ensure that NIST-validated or NSA approved cryptography is used to implement encryptions, key exchanges, digital signatures, and hashes applying newer, more robust standards as they become available.

- *(APS0320: CAT II) The IAO will ensure encryption use, according to the data classification, is NIST-validated or an NSA approved cryptography.*

3.4.2 Account Management

User accounts defined for access to application servers and their hosted web applications allow various interactions with all the associated application services, underlying OS, and external supporting systems. These interactions can be relatively benign, such as access to availability information, or quite powerful, such as reconfiguration, redeployment of web applications, or even restart and stop of any of the interfaced systems. Accounts must be defined carefully so that privileges between an application server and its hosted web applications or external supporting systems are not granted unintentionally. It is important that privileged accounts are used only when required in order to closely document and control the use of privileges.

Access controls with supporting programs and data are directly related to privileges granted in the application server account definitions. The controls provide little protection if the practices used for account management are weak. In large organizations, careful account management can be a significant administrative burden. One mechanism for dealing with that burden is the use of role based access controls (RBAC). This helps to control privileges according to a user's functional need for them and should simplify account maintenance.

The following requirements enforce the principles of least privilege and separation of duties for application server accounts. This means that privileges are not granted unless necessary and not used unless intended. The requirement to use role based access control helps to reduce the complexity and potential errors associated with privileged account maintenance.

- *(N/A: CAT II) The IAO will ensure ASA accounts are assigned the minimum privileges required for the user's job function.*

- *(N/A: CAT III) The IAO will ensure ASA accounts are not used for non-privileged functions.*

- *(APS0510: CAT II) The IAO will ensure ASA accounts are established and administered in accordance with a role-based access scheme to the maximum extent technically feasible within the application server software.*

Errors or omissions in account management may be the result of process problems. Processes that are documented and implemented are helpful in avoiding problems that could weaken

security over application server accounts. Formal documentation of privilege assignment is an essential part of the management process.

Some vendors provide installation procedures that include or specify the use of particular User IDs (UIDs) and passwords. These are referred to as the factory set, default, or standard values. The account management process must incorporate manual or automated steps to handle situations that would otherwise allow continued privileged access after it is no longer authorized. Default accounts left over from a newly installed system must be changed or removed. The following requirements provide for the removal or modification of accounts installed by default or that are no longer needed.

- *(N/A: CAT II) The ASA will ensure all factory set, default, or standard user accounts that are not needed are removed.*

- *(APS0210: CAT I) The ASA will ensure all factory set, default, or standard user passwords for the application server are changed.*

3.4.3 Auditing

Auditing for information systems involves the collection and retention of data so that it is possible to assess the adequacy of system controls and the degree of compliance with policies and procedures. Audit data provides information needed to evaluate the source, scope, and impact of a security incident.

Auditing will be configured and implemented on an application server. Auditing will be capable of capturing all application server and hosted web application operations. This includes both events that occur by the application server and within its hosted web applications that affect modification to any application server parameters and resources and events performed on or by the host system such as the application server startup and shutdown. Audit data will be maintained for one year. The audit data is not required to be local to the application server for a year, but will be available for historical analysis if needed. Audit data will only be readable by personnel authorized by the IAO.

As with other areas of concern, auditing for application servers has additional significance because the application server may execute as privileged processes. Certain user actions, performed within a limited time period and in certain patterns, can be signs of preparation or attempts to exploit system vulnerabilities that involve privileged access. These actions include attempts to access security files and attempts to access an interactive application on the host system. Certain actions taken by the application server, in response to a perceived threat, are also potential signs of an attack. These actions include denying access due to successive invalid password entries; disabling user accounts, network ports, or other access mechanisms; or otherwise flagging actions that appear to be malicious.

Taken individually, these events are not absolute indicators and any response to them could be premature. However, if the execution of the actions is not recorded, it becomes impossible to recognize later the pattern that confirms the occurrence of an attack. Therefore, it is necessary to capture this information as the events occur.

- *(N/A: CAT II) The ASA will ensure auditing is enabled.*

- *(APS0410: CAT II) The ASA will ensure audit data from an application server containing UID, date and time of event, type of event, and success or failure of event are written for the following.*

  - *Successful and unsuccessful attempts to access security (e.g., account or permission) files*
  - *Successful and unsuccessful logon to (attempt to access) the application*
  - *Denial of access resulting from excessive number of logon attempts*
  - *Blocking or blacklisting a UID, terminal, or access port, and the reason*
  - *Activities that might modify, bypass, or negate safeguards controlled by the application*

As mentioned, it may take a collection of data over time to recognize an attack. Because the time span of an attack can be lengthy, some period of data retention has to be selected. Also, because investigations can take extended periods of time, it is necessary to be sure that important evidence is not inadvertently lost.

- *(APS0750: CAT II) The IAO will ensure audit data generated by an application server is retained for at least one year.*

Some application servers offer built-in facilities for collecting and reporting of audit data, or even for generating warnings to the information owners based on the data captured. While such features may not appear to be immediately important from an operational view, waiting until malicious activity is suspected to deploy the features can mean that the data is lost or is not processed in time to detect an attack in progress. An application server might also have the ability to take defensive action such as disabling user access or IP blocking, based on the malicious activity. Deploying these product capabilities early ensures that they will be available when needed.

- *(N/A: CAT II) To the extent technically feasible for an application server, the IAO will ensure an automated, continuous on-line monitoring and audit trail creation capability is deployed and configured to immediately alert personnel of any unusual or inappropriate activity with potential IA implications and that it provides the capability to configure automatic disabling of the application if serious IA violations are detected.*

A primary benefit of collecting audit data is lost if the data is only reviewed when a security incident has been confirmed. A proactive approach to reviewing the data on a regular basis is important for early detection and for prevention of attacks.

- *(N/A: CAT II) The IAO will ensure audit data generated by application servers are regularly reviewed for indications of inappropriate or unusual activity and that suspected IA policy violations are analyzed and reported.*

Protecting audit data requires safe physical storage and appropriate access control. To accomplish this, the data is backed up and access control definitions are implemented to limit updates to the original and backup copies. The destruction of audit data due to media failures or

the absence of update access controls represents a double loss. The ability to recognize and possibly recover application server audit data that was tampered with is lost and the system resources expended to collect, process, and store the data are effectively lost.

- *(APS0740: CAT II) The IAO will ensure an application servers audit data is backed at least weekly onto a different system or media than the system on which the application server executes.*

- *(N/A: CAT II) The IAO will ensure that access to application server audit data, including backup copies, is limited so that only authorized application server processes, and ASAs can read, update, or delete it.*

- *(N/A: CAT II) Where technically feasible, the IAO will ensure that an auditor role is created in the application server to access and manage audit data. The least privileges required to perform the audit function will be assigned to this role.*

- *(N/A: CAT II) The IAO will ensure that ASAs are not granted permissions to the audit data in the application sever where technically feasible.*

- *(N/A: CAT II) The ASA will ensure that access and changes to audit data are audited.*

According to the DODI 8500.2, IA Controls for changes to data, logging access, and changes to data is required for some environments. Determine the scope and depth of auditing capabilities that has been implemented for a given application server. All J2EE compliant application servers provide default or standard audit and logging facilities. Other mechanisms for auditing may be employed but are beyond the scope of this STIG. Logging can be accomplished through two facilities:

- Current Windows OSs have the capability for file and directory auditing. The generated data specifically identifies the data access attempted. *Windows 2003/2000/XP Addendum* security guidance has requirements that address file and directory access logging. Implementation of that guidance is assumed.

- Application server components record process information in individual log files. Although this information is not intended to be a data access log, elements of it can be used for that purpose.

3.4.4 Application Server State Change

Access to application server availability controls including privileges to startup, shutdown, or stop the application server can be used to render the application server and its hosted web applications inaccessible to users. To secure these controls, restrict access to the associated privileges to authorized ASAs.

- *(APS0580: CAT II) The IAO will ensure functions that include system initialization, shutdown, or features that may compromise the application server's availability is restricted to authorized ASAs.*

3.4.5 Warning Banner

The requirement for a warning banner is essential because it preserves the Government's right to monitor, record, and audit session activities. If this requirement is not fulfilled, data that is collected may have to be excluded from any prosecution of an intruder.

- *(APS0810: CAT II) The ASA will ensure the hosted web applications on the application server system are configured to present an initial warning banner advising the user that:*

  - *The system is a DOD system.*
  - *The system is subject to monitoring.*
  - *Monitoring is authorized in accordance with applicable laws and regulations and conducted for purposes of systems management and protection, protection against improper or unauthorized use or access, and verification of applicable security features or procedures.*
  - *Use of the system constitutes consent to monitoring.*
  - *The system is for authorized U.S. Government use only.*

An example of an acceptable warning banner can be found in the *Chairman of the Joint Chiefs of Staff (CJCS) Manual 6510.01, Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)*. Questions concerning the legal validity of specific text should be addressed to the DOD Component's General Counsel.

3.5 I&A

This section describes application server security requirements based on applicable DODI 8500.2, IA Controls in the I&A subject area. These requirements address items that are used to identify and authenticate a user and as input to encryption processes. These items include account IDs, PKI certificates, and symmetric and asymmetric keys.

An I&A service establishes a user's identity and verifies that the requesting user is correctly associated with that identity. Because of the sensitive nature of this service, some of the requirements stated here apply specifically to application servers that provide that service. A final general note on I&A services involves the choice where multiple options exist. Some application servers allow a choice between using host or application-based I&A services. A single point of security control, host authentication is generally better because it simplifies account maintenance and auditing, and it provides an integrated point for assigning privileges. Therefore, when the option exists and there are no significant security advantages to the contrary, host I&A services should be implemented instead of the equivalent application server implementation.

Integrity checking at the message level and host authentication is frequently related to network sessions. One common implementation is the use of the Secure Sockets Layer (SSL) or a

follow-on protocol. Session protocols like SSL offer hash checking as messages are exchanged to validate the contents of each message. Host authentication occurs at session startup and periodically thereafter. By requiring both the server and client to authenticate each other, the data source is established with higher assurance than simple network address validation. The use of session protocols requires some integration depending on the flavor of application server used, usually offering a standards-based solution that can be implemented using elements of the DOD Public Key Infrastructure (PKI).

The DOD established the External Certification Authority (ECA) program to support the issuance of DOD-approved certificates to industry partners and other external entities and organizations. The ECA program is designed to provide the mechanism for these entities to securely communicate with the DOD and authenticate to DOD Information Systems. The application server and its hosted web applications can use only DOD Interim External Certification Authority (IECA) or ECA approved certificates. The certificates used for authentication, access control, data integrity, and non-repudiation, must be used in accordance with the DODI 8520.2, *Public Key Infrastructure (PKI) and Public Key (PK) Enabling, dated 1 April 2004*.

PKI enabled systems depend on digital certificates, electronic credentials issued by a certificate authority (CA), to establish identity and trust. These may be referenced at http://iase.disa.mil/pki/eca/.

A PKI and PK enabling solution will be implemented for all DOD unclassified and classified information systems for components that support authentication, access control, confidentiality, data integrity, and non-repudiation in accordance with the DOD Instruction 8520.2, *Public Key Infrastructure (PKI) and Public Key (PK) Enabling, dated 1 April 2004.* The application server and its hosted web applications will be PK enabled.The use of Secure Socket Layer (SSL) alone does not constitute a PK enabled application.

- *(APS0110: CAT II) The IAO will ensure DOD PKI Class 3 or 4 certificates and hardware security tokens (when available) are used for I&A, encryption, digital signing, access control, and data integrity when supporting the application server and its web applications.*

- *(N/A: CAT II) The ASA will ensure the application server and its hosted web applications have implemented a PKI and PK enabling solution that uses NIST-validated or NSA approved cryptography for I&A services.*

I&A services provided by application servers must include their own protective controls better than or equivalent to other providers (such as OSs) of those services. These controls address repeated logon attempts, concurrent sessions, and notification of privacy and security conditions. If the controls are not implemented, the system is more vulnerable to password attacks, continuing use of compromised accounts, and the loss of the right to use audit data.

- *(N/A: CAT II) The ASA will ensure the application server provides functions to control the following:*

- *Access is denied after multiple unsuccessful logon attempts.*
- *The number of access attempts in a given period is limited.*
- *A time delay control system is employed.*

- *(APS0310: CAT I) If an application server provides internal I&A services, the IAO will ensure any passwords stored by the application server are encrypted using NIST- validated or NSA approved cryptography.*

- *(APS0350: CAT I) The IAO will ensure the transmission of authentication data for access to the application server and its hosted web applications is encrypted using NIST- validated or NSA approved cryptography.*

3.5.1 Password Guidelines

Passwords must be protected from being accessed by unauthorized users. When an account is created for a user, that user will be given a temporary password. The administrator will brief the user on DOD password policy (DODI 8500.2 and Chairman of the Joint Chiefs of Staff (CJCSM) 6510.01C) and implementation of password protection. All passwords will be stored in an encrypted format. The application server account name and password will not be visible to the host or client OS. Where available, application server account logons will be limited to three failed logons before they become locked. This requirement reduces the ability for password cracking programs to be used successfully. The ASA will set the duration of the lock time to a specific length as approved by the IAO for the application or site or require a manual reset. The duration should be set appropriately for the environment, keeping in mind that the longer the duration, the more protected the accounts will be from password cracking programs.

- *(N/A: CAT II) The ASA will ensure application server account passwords conform to DOD password policy and each user is briefed on the policy on receiving a temporary password.*

- *(N/A: CAT II) The ASA will ensure application server account passwords are stored in an encrypted format.*

- *(N/A: CAT II) The ASA will ensure application server account passwords are a minimum of eight alphanumeric characters in length and do contain a mix of upper case letters, lower case letters, numbers, and special characters.*

- *(N/A: CAT II) The ASA will ensure application server account passwords do not contain personal information such as names, telephone numbers, account names, dictionary words, etc.*

- *(N/A: CAT II) The ASA will ensure application server account passwords do not contain consecutively repeating characters.*

- *(N/A: CAT II) Where possible, the ASA will ensure that new application server account passwords differ from the previous password by at least four characters when a password is changed.*

- *(N/A: CAT II) The ASA will ensure application server account passwords are changed every 90 days or more frequently.*

- *(N/A: CAT II) The ASA will ensure application server account passwords are not reused within ten password changes.*

- *(N/A: CAT II) Where available, the ASA will ensure application server account passwords are not reused for a period of one year or longer.*

- *(N/A: CAT II) The ASA will ensure application server account passwords are changed at least once a year and anytime an ASA is reassigned.*

- *(N/A: CAT II) The ASA will ensure users are not allowed to change their passwords more than once every 24 hours without IAO approval.*

3.5.2 Sessions

The most common method of tracking a user through a web application is by assigning a unique session ID. The session ID information is then transmitted back to the application server with every request. Unfortunately, should an attacker guess or steal this session ID information, it is normally a trivial exercise to hijack and manipulate another user's active session.

An important aspect of correctly managing state information through session IDs involves the authentication processes. Session IDs are not only used to follow clients throughout the web application experience, they are also used to uniquely identify the authenticated users. Thereby regulating access to an application server's content and information. It is important to ensure session limits exist and are configured appropriately to determine how long a particular web application will maintain a session on the server (on behalf of the client.) Another benefit to controlling state information through regulated session ID's is the reduction of risk for Denial Of Service (DOS) attacks.

- *(APS0530: CAT II) If an application server provides internal I&A services, the ASA will ensure a session limit is defined and does not exceed 24 hours.*

- *(N/A: CAT II) The ASA will ensure session ID's are created utilizing NIST- validated or NSA approved cryptography.*

APPENDIX A: RELATED PUBLICATIONS

A.1 Government Publications

Chairman of the Joint Chiefs of Staff (CJCS) Manual 6510.01, “Defense-in-Depth: Information Assurance (IA) and Computer Network Defense (CND)”, dated 10 August 2004.

Defense Information Systems Agency, Database Security Technical Implementation Guide.

Defense Information Systems Agency, Desktop Application Security Technical Implementation Guide.

Defense Information Systems Agency, Enclave Security Technical Implementation Guide.

Defense Information Systems Agency, Web Content Access Control White Paper.

Defense Information Systems Agency, Web Server Security Technical Implementation Guide.

DOD Instruction 5200.40, DoD Information Technology Security Certification and Accreditation Process (DITSCAP), dated 30 December 1997.

DOD Directive 8500.1, "Information Assurance (IA)," dated 24 October 2002.

DOD Instruction 8500.2, Information Assurance (IA) Implementation, dated 6 February 2003.

DOD Instruction 8520.2, Public Key Infrastructure (PKI) and Public Key (PK) Enabling, dated 1 April 2004.

DOD Instruction 8551.1, Ports, Protocols, and Services Management (PPSM), dated 13 August 2004.

DOD Ports, Protocols, and Services (PPS) Assurance Category Assignments List.

DOD Ports, Protocol, and Services, ASD/C31 Memorandum "Increasing Security at the Internet/DISN Boundary," dated 28 January 2003.

DOD Policy Guidance for use of Mobile Code, Policy Guidance for use of Mobile Code Technologies in Department of Defense (DOD) Information Systems Memorandum.

DOD Web Policy, Web Site Administration Policies and Procedures, dated 25 November 1998.

NSA Guide, BEA WebLogic Platform Security Guide, Version 1.0, dated 4 April 2005.

NSA Guide, Guide to Microsoft .NET Framework Security, Version 1.4, dated 22 September 2004.

A.2 Web Sites

ECA PKI Program	http://iase.disa.mil/pki/eca/
IASE	http://iase.disa.mil
JTF-GNO	http://www.cert.mil
NSA	http://www.nsa.gov
NIST Computer Security Resource Center (CSRC)	http://csrc.nist.gov/pcig/cig.html
The Apache Jakarta Project, Tomcat	http://jakarta.apache.org/tomcat/index.html
United States Department of Defense	http://www.defenselink.mil/
WebLogic Platform 8.1 Online Documentation, Security	http://e-docs.bea.com/wls/docs81/security.html

APPENDIX B: APACHE JAKARTA TOMCAT

Tomcat is an open source application server that is a subproject of the Jakarta Project. It is designed to adhere in its entirety to the specifications of Sun's JSPs and Java Servlets. Java Management Extensions (JMX) provides increased granularity in management of web applications and are implemented in Tomcat version 5 and later. The added granularity provided by JMX provides a basis for the assignment of security controls to more finely defined security contexts.

The following terms, when used to refer to Tomcat elements, are defined as follows:

- Server - A Server represents the complete container or application server instance. It is known as the implementation of the Server interface.

- Service - A Service is known as an intermediate component, residing inside the application server using one or more Connectors for one Engine. This is not to be confused with the term application service used throughout this STIG.

- Engine - An Engine is known as the request-processing pipeline of a specific Service. The Engine receives and processes all requests from active Connectors, handing the response back to the appropriate Connector for transmission to the client.

- Host - A Host is an implementation of the network name (e.g., www. somecompany.com) to the Tomcat application server. An Engine can contain one-to-many Hosts, and can use network aliases such as somecompany.com and ww2.somecompany.com.

- Connector - A Connector provides communications with the client. Several Connectors are defined for Tomcat using the Connector interface. They are the Coyote Connector, used for HyperText Transfer Protocol (HTTP) traffic, the web serving component, and the JK/JK2 connectors, also known as the Apache JServe Protocol (AJP) protocol for connecting Tomcat to an external web server.

- Context - A Context represents the hosted web application. Hosts have one-to-many Contexts.

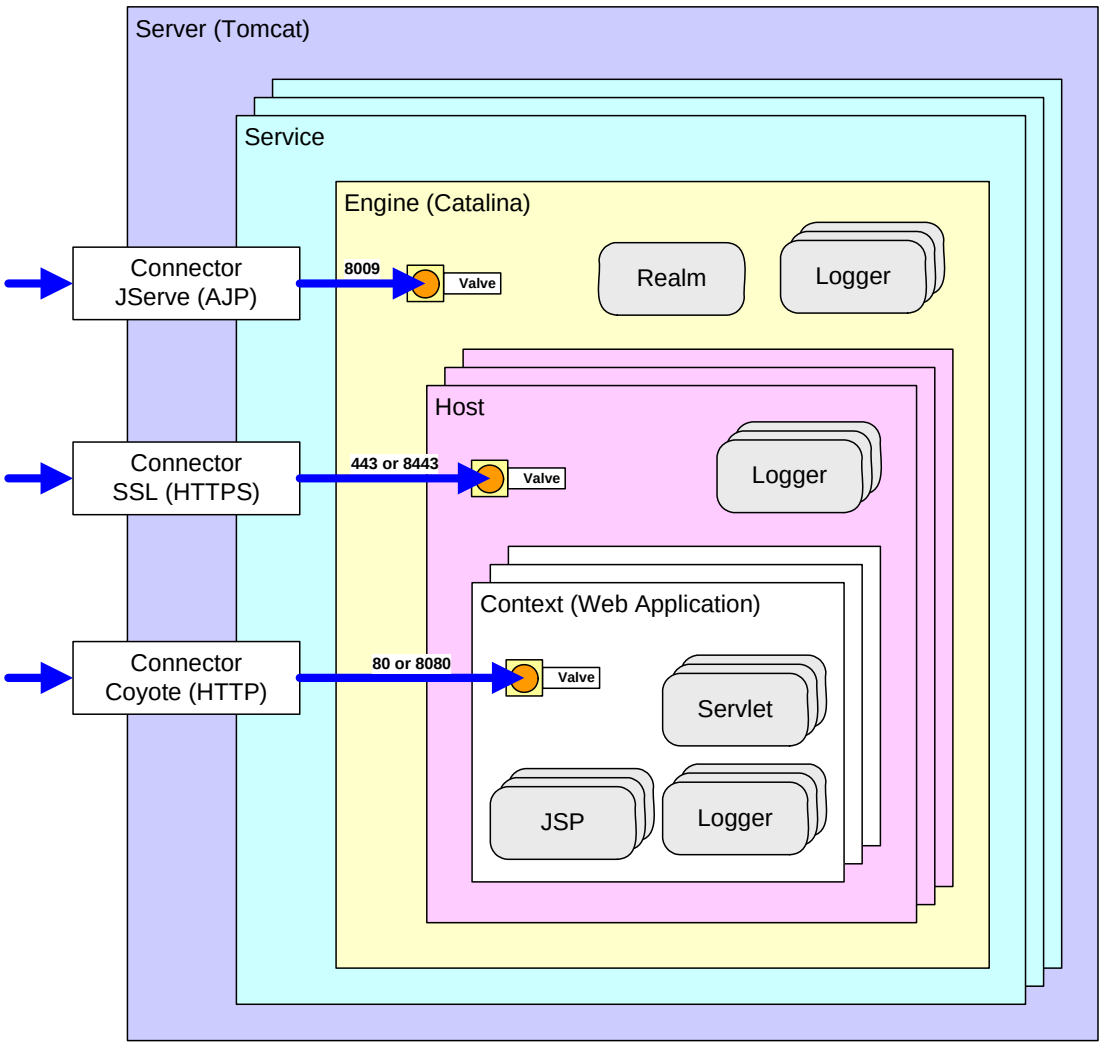

Figure B-1. Tomcat Component Architecture

B.1 Current Tomcat Version

The information contained in this appendix is specific to the Apache Jakarta Tomcat Versions 4.1.x, 5.0.x, and 5.5.x. When version-specific information is presented, it will be labeled with the version to which it specifically applies. The installed Tomcat software will have all security patches applied.

B.2 Setup and Startup

Tomcat may be configured to run as either a single-user application or as a shared system service or process. A Tomcat service or process does not need host platform administrator or root privileges to operate. To limit the risk of exploits of the Tomcat server, a custom host account dedicated to running the Tomcat service or process will be created and assigned minimal host system privileges.

- *(N/A: CAT II) The ASA will ensure the Tomcat application server is run using a dedicated, non-privileged account.*

B.3 Realms and Access Control

In Tomcat, the web application security model is built around the concept of users and roles. Users are assigned to a role, which determines the resources that the user is allowed to access.

Configured Realms is defined as a collection of users, passwords, and their associated roles. Tomcat's hosted web applications utilize Container Managed Security defined as a standard mechanism used for storing account credentials to authenticate clients. Realms enforce which resources are accessible by which groups of users defined in the *web.xml* deployment descriptor. This allows the administrator to configure Tomcat to retrieve user, password, and role information using one or more of the available Realm configurations.

For each Realm defined, password storage and transmission must be protected using message digest encryption to prevent unauthorized use and tampering.

Several types of frameworks for realms exist for implementation depending on the need and infrastructure requirements of the Tomcat server and applications. The *UserDatabase*, Java Database Connectivity (JDBC) *JDBCRealm*, the Java Naming and Directory Interface (JNDI) *JNDIRealm*, and the Java Authentication Architecture Service (JAAS) *JAASRealm* can be specified for use by including their definition in the *server.xml* file using the *className* attribute to provide their configuration information.

B.3.1 UserDatabaseRealm

The UserDatabaseRealm is loaded into memory from a static file and kept in memory until Tomcat is shut down. In fact, the representation of the users, passwords, and roles that Tomcat uses resides only in memory, and is reloaded when Tomcat is restarted. The default file for assigning permissions in a UserDatabaseRealm is the *tomcat-users.xml* file, located in the *$CATALINA_HOME/conf* directory.

The *tomcat-users.xml* file is where the group, user, and password credentials are stored. It contains a list of users who are allowed to access Tomcat and its hosted web applications. It is an XML file, with a root element of *tomcat-users* and sub-elements of *role* and *user*. Each role element has the single attribute *rolename*. Each user element has three attributes: *username*, *password*, and *roles*. All passwords stored in the *tomcat-users.xml* file must be digested using a one-way hash. To further protect this file, access to this file will be restricted to ASAs, the Tomcat service account, and authorized web applications.

- *(N/A: CAT II) The ASA will ensure access to the tomcat-users.xml file is restricted to ASAs and the Tomcat service account.*

- *(APS0140: CAT II) The ASA will ensure client authentication process uses strong user authentication that resists spoofing, such as a two-factor system.*

The Java Cryptography Architecture (JCA) defines several message digest algorithms that can be used to store passwords for various available Realms. Message digest algorithms supported by the java.security.MessageDigest can be specified when generating an instance of MessageDigest.

- *(APS0320: CAT II) The ASA will ensure the client authentication process uses NIST-validated or NSA approved cryptography.*

B.3.2 JDBCRealm

The JDBCRealm provides more flexibility than a UserDatabaseRealm, and accesses data dynamically. It is a realm connected to a relational database where users, passwords, and roles are stored. A JDBCRealm accesses these data credentials as often as needed. As accounts are manipulated in the relational database, the JDBCRealm is able to access it immediately. Protection of passwords stored for each JDBCRealm is configured by assigning the *digest* attribute with a setting of *SHA* within the server.xml file.

- *(N/A: CAT II) The ASA will ensure the JDBCRealm digest attribute is set to SHA.*

B.3.3 JNDIRealm

Tomcat can be configured to retrieve usernames, passwords, and roles from an LDAP directory, called JNDIRealm. JNDIRealm is an alternative Realm implementation, where authentication of users is queried from an LDAP directory of usernames, passwords, and roles. JNDIRealm recursively searches an LDAP hierarchy of entries until it finds the information it needs. JNDIRealm is configured to search a specific location in the directory server for the requested information.

Attribute	Meaning	Required Setting
connectionName	The username used to authenticate a read-only LDAP connection. If left unset, an anonymous connection will be made.	**Will be set**
connectionPassword	The password used to establish a read-only LDAP connection.	**Will be set**
Digest	Digest algorithm (SHA only). The default is "cleartext".	**SHA**
userPassword	The name of the attribute in the user's directory entry containing the user's password. Specifying this value, the JNDIRealm will bind to the directory using the values specified by the connectionName and connectionPassword attributes, and retrieve the corresponding password attribute from the directory server for comparison to the value specified by the user being authenticated. If the digest attribute is set, the specified digest algorithm is applied to the password offered by the user before comparing it with the value retrieved from the directory server. If left unset, JNDIRealm will attempt a simple bind to the directory using the DN of the user's directory entry and password specified by the user, with a successful bind being interpreted as a successful user authentication.	**Will be set**

Table 5. JNDIRealm Attributes

- *(N/A: CAT II) The ASA will ensure the JNDIRealm digest attribute is set to SHA.*

- *(N/A: CAT II) The ASA will ensure the JNDIRealm connectionName, connectionPassword, and userPassword attributes are defined.*

B.3.4 JAASRealm

JAASRealm is an experimental Realm implementation that authenticates users via JAAS. JAAS implements a version of the standard Pluggable Authentication Module (PAM) framework that allows applications to remain isolated from the authentication process. New and updated authentication implementations may then be introduced to Tomcat without requiring modifications to the application itself. Because JAAS represents a highly customized type of authentication logic that Tomcat does not implement on its own, this Realm is beyond the scope of this STIG and should be reviewed independently. The type and method of authentication implemented should be thoroughly inspected and carefully documented in order to determine it meets current DOD standards.

B.3.5 Security Manager

To support additional security control over the behavior of Tomcat web applications, the configuration and use of the Java Security Manager (JSM) is required. The JSM allows for an application server to be configured with fine-grained security policies. This allows Tomcat to accept or reject a program's attempt to shut down the JVM, access local disk files, or connect to arbitrary network locations.

- *(APS0560: CAT I) The ASA will ensure the Application Security Manager is enabled.*

The configuration file for security decisions in Tomcat is the *catalina.policy* file, written in the standard Java security policy file format. The JVM reads this file when Tomcat is invoked with the *-security* option. The file contains a series of permissions, each granted to a particular codebase or set of Java classes. An example control for a Java security policy, assuming that the hosted web application's root directory is */home/somecompany/webapps/*, is shown in the following example:

```
grant codeBase "file:/home/somecompany/webapps/-" {
        permission java.net.SocketPermission "dbhost.somecompany.com:5432",
        "connect";
        }
```

A list of permission names and their allowed usage is given in *Table 6*.

Permission name	Meaning	Allowed Usage
java.io.FilePermission	Controls read/write/execute access to files and directories.	**Limited**
java.security.AllPermission	Grants *all* permissions.	**No**

Table B-2. Security Manager Attributes

- *(N/A: CAT II) The ASA will ensure the java.io.FilePermission is set to Limited.*

- *(N/A: CAT II) The ASA will ensure the java.security.AllPermission is set to No.*

Tomcat hosted web applications may make use of the file system to save and load data. If Tomcat is executed with the SecurityManager enabled, it will not allow the hosted web applications to read and write their own data files. To enable required hosted web applications operations under the SecurityManager, grant each hosted web application the minimal permissions to accomplish the required operation.
The following lines are an example of file permission grants to a web application defined in the *catalina.policy* file:

```
grant codeBase "file:${catalina.home}/webapps/ROOT/-" {
      permission java.io.FilePermission "${catalina.home}/webapps/ROOT/test.txt",
      "read, write, delete";
      };
```

- This grants the ROOT web application permissions to read, write, and delete only its own test.txt file.

- *(N/A: CAT II) The ASA will ensure the least permissions to perform required operations of web applications are assigned in the catalina.policy file.*

Critical Tomcat functions are available for access via listening network ports. The following is a template of example rules to be configured on a firewall to effectively secure the critical ports from unauthorized access. To further protect Tomcat' control and connector messages, configure the firewall to disallow these message types that originate from outside the enclave.

- *(APS0710: CAT I) The IAO or IAM will ensure the perimeter firewall includes the following rules to protect sensitive Tomcat ports from unauthorized access:*

1. *block [inbound] on [external_interface] protocol [tcp] from [any] port [8005] to [any]*
2. *block [inbound] on [external_interface] protocol [tcp] from [any] port [8009] to [any]*
3. *allow [inbound] on [external_interface] protocol [tcp] from [web_server] port [8009] to [this_machine]*

***NOTE**: The last rule (rule #3) only applies if Tomcat is used in conjunction with a web server, such as Apache HTTPd Server, Microsoft IIS, etc.*

Each hosted web application restricted to read and write files within its own directory structure. There should be a *grant* instance for each hosted web application that assigns the minimum file and directory access permissions necessary for operation. If read, write, and/or execute privileges are necessary for proper operation, they must be individually assigned to authorized trusted code. Under no circumstance should a hosted web application grant access to its *WEB-INF/* directory where sensitive **.xml* files containing deployment descriptors are located. Under no circumstances should any hosted web application grant the *<<ALL FILES>>* attribute, as it gives the hosted web applications full access to the local host file system.

- *(N/A: CAT II) The ASA will restrict access to hosted web application files and directories to the assigned web application.*

- *(APS0330: CAT II) The ASA will ensure the <<ALL FILES>> attribute is not granted to any hosted web application.*

NOTE: Use the following to help troubleshoot the SecurityManager if the *catalina.policy* file does not work as expected. To debug security problems, add this Java invocation when starting Tomcat:

-Djava.security.debug="access,failure"

Then check audit logs for any security debug lines with the word “denied” in them. Any security failures will leave a stack trace and a pointer to the ProtectionDomain that failed.

B.4 Authentication

Container-managed authentication methods control how user credentials are verified when a protected resource is accessed. There are four types of container-managed security that Tomcat supports, and each obtains credentials differently.

In order to implement any of the container-managed authentication methods that Tomcat provides, a more robust third party or custom developed solution must be implemented. Tomcat does not include I&A security mechanisms to meet DOD account and password requirements, that include measures such as checking for password strength, login successes and failures, and account lockout, etc. The following sections describe some guidelines when using Tomcat in conjunction with alternative solutions to meet DOD requirements. See section 3.5 for I&A details and section 3.5.1 for password guidelines.

- *(N/A: CAT II) The IAO will ensure the default Tomcat I&A services are supplemented with an authentication and account management solution that meets DOD account and password requirements.*

B.4.1 Basic Authentication

Tomcat uses the container-managed authentication BASIC to query a client’s web browser for a username and password whenever the browser requests a protected Tomcat hosted web application resource. Basic authentication is defined by a value of BASIC in the *web.xml* file’s *authmethod* element. Basic authentication transmits a user’s password via HTTP as base64-encoded text. The password encoding used by basic authentication is considered insecure, easily broken, and insufficient for protection of passwords in transit across the network.

- *(N/A: CAT II) The ASA will ensure basic authentication is not used to access Tomcat or its hosted web applications in production environments.*

B.4.2 Digest authentication

Digest authentication is defined by a value of DIGEST in the *web.xml* file's *authmethod* element. When digest authentication is used with a more robust supplemented authentication service, it is a suitable alternative to basic authentication because it sends passwords across the network and on disk in a more strongly encoded form using hashes. The password is requested via HTTP authentication in the form of a digest-encoded string. The main disadvantage to using digest authentication is that some legacy web browsers do not support it. There is not a definitive list of client web browser versions that are known to support or not support digest authentication. Therefore, before deciding to use digest authentication for a hosted web application, it is suggested that each client browser brand and version is tested for support.

In addition to specifying Tomcat configuration for password storage, each user's password may be manually encoded using a specified hash format. This involves a two-step process that is repeated with each defined user's password.

- *(N/A: CAT II) The ASA will ensure when DIGEST authentication is used to access Tomcat applications that the stored passwords are protected by NIST- validated or NSA approved cryptography.*

B.4.3 Form Authentication

Form authentication displays a web page login form to the user when the user requests a protected resource from a hosted web application in Tomcat. Defining the auth-method element's value of FORM specifies form authentication.

To implement form-based authentication, define a custom developed login form page and an authentication failure error page for the hosted web application in Tomcat. To complete form authentication securely, define a *security-constraint* element and a *login-config* element in the *web.xml* file. To ensure passwords are not sent in the clear, activate SSL so that transmitted sensitive data is encrypted.

- *(N/A: CAT II) The ASA will ensure FORM authentication uses SSL configured in accordance with NIST- validated or NSA approved cryptography.*

B.4.4 Client-cert Authentication

To allow clients to authenticate without the use of a password, a browser can present a client-side X.509 digital certificate as the login credential. Each user is issued a unique digital certificate that the application server will recognize. How the certificates are generated and stored is up to the ASA and is a manual process. Once users import and store their digital certificates in their web browsers, the browsers present them to the server whenever the server requests them. The client-cert is defined as CLIENT-CERT in the *web.xml* file's *auth-method* element and only available when serving content via SSL (i.e., HTTPS).

- *(N/A: CAT II) The ASA will ensure CLIENT-CERT authentication is implement in conjunction with SSL configured in accordance with NIST- validated or NSA approved cryptography.*

- *(N/A: CAT II) The ASA will ensure CLIENT-CERT authentication is accomplished using a DOD PKI Class 3 or 4 certificate and hardware security token (when available), or an NSA-certified product.*

In order to log and audit client-cert authentication attempts, set the debug attribute of the Realm to 2 (or to a higher number) in the *server.xml* file.

- *(N/A: CAT II) The ASA will ensure the debug attribute for Tomcat realms is set to 2 or higher in the server.xml file.*

B.5 Connector Management

In Tomcat, a connector is configured to listen on a specific port with a definable set of parameters. Tomcat has two classifications of connectors, one that allows browsers to connect directly to Tomcat (the default HTTP connector on port 8080) and one that listens for and handles traffic from traditional web servers (the default AJP using the JK/JK2 connector on port 8009). Tomcat ships with some of these connectors enabled by default in its configuration file *server.xml*, and some that are disabled. It is important to disable the connectors not required for the operation of Tomcat or the supported web applications..

- *(N/A: CAT II) The ASA will ensure unused connectors are disabled.*

B.6 Default Tomcat Applications

Tomcat also has several web applications that ship with its default install. Depending on the version of Tomcat, various web applications may or may not be installed or enabled by default. The possible applications known at the time this STIG was written are the following:

- The Manager application
- The Admin application
- The examples applications
- The Web Distributed Authoring and Versioning (WebDAV) application
- The balancer application

Each application provides a particular service and set of functions that may or may not be necessary to support the application server's mission. Some of these web applications should never be used, while others will require some configuration tuning to avoid being susceptible to future vulnerabilities.

The default manager and admin web applications for Tomcat provide deployment and management of the application server itself and of its hosted web applications. Access to the

manager and admin applications require the assignment of a privileged role defined in the *tomcat-users.xml* located in the Tomcat's */conf* directory.

- *(N/A: CAT II) The IAO will ensure assignment of the Tomcat privileged role is restricted to authorized ASAs.*

Most installations of Tomcat include a set of JavaServer Pages (JSP) and servlet examples for demonstrating capability. The */examples* context directory and its contents should be completely removed as it has several well known vulnerabilities that an attacker can use as a means of discerning sensitive installation information about Tomcat.

Another part of a default installation of Tomcat, at least in earlier versions, the WebDAV application is a method of allowing remote authoring. Tomcat supports WebDAV and WebDAV 2 for development environments that require WebDAV services. Under no circumstances will this be used on production Tomcat applications servers. Remove the */webdav* Context directory and its subsequent contents. WebDAV is to be used solely in development environments and not suitable for production scenarios.

- *(APS0610: CAT II) The ASA will ensure the following Tomcat components are not installed:*

  - *The jsp-examples and servlet-example applications.*
  - *The WebDAV application.*

B.7 File Permissions

This section is intended to provide guidance on security measures that should be observed for the installation of Tomcat. Compliance will be the responsibility of the ASA. In order to ensure the security of Tomcat the following must occur:

- *(N/A: CAT II) The ASA will restrict access to Tomcat and its hosted web application files and directories to ASAs, the Tomcat service account, and authorized web applications.*

- *(N/A: CAT II) The ASA will ensure the assignment of authorization roles is restricted to authorized accounts.*

- *(N/A: CAT II) The ASA will create privileged system accounts for administrative functions and assign least host and Tomcat privileges required to administer the applications.*

To accomplish the goal of reducing Tomcat's permissions to the minimum required revoke all default permissions granted to the application server's system account. Tomcat need only have permissions to list directory contents, read, and execute the JRE files located where the systems JVM is installed. Tomcat also needs access to the directory tree where it is installed. Here, only read access is necessary for Tomcat to access anything within its directory tree, except for the *$CATALINA/conf* directory which may need write access, if the UserDatabase Realm is used. However, if the Tomcat manager web application is used to deploy new web applications (*.WAR files), then write access is necessary for it to function properly. In the case where a

hosted web application needs permissions other than read, it is up to the ASA to determine whether it is appropriate on a case-by-case basis.

This page is intentionally left blank.

APPENDIX C: BEA WEBLOGIC ADDENDUM

C.1 Background

This Addendum to the NSA Guide, *BEA WebLogic Platform Security Guide* was developed to enhance the confidentiality, integrity, and availability of sensitive DOD AISs and EISs using the BEA WebLogic Application Server.

This Addendum is coordinated with the following documents here after collectively known as the NSA WebLogic Guide:

- NSA Guide, BEA WebLogic Platform Security Guide, Version 1.0, dated 4 April 2005.

This Addendum is designed to supplement the security guidance provided by the NSA Guide, *BEA WebLogic Platform Security Guide* with DOD-specific requirements. The requirements listed in this STIG in addition to the NSA WebLogic Guide specify the minimum requirements for securing a BEA WebLogic application server installation.

C.2 General

A coordinated and planned setup of a WebLogic application server is important before and after the WebLogic application server installation. To ensure that the WebLogic application server's general configuration aspects are securely implemented, application of the NSA WebLogic Guide must be followed. The following guidelines outline recommended additions to the important security points listed in the NSA WebLogic Guide.

- *(N/A: CAT II) The ASA will restrict access to the following files to ASAs and the WebLogic service account:*

  - *boot.properties*
  - *config.xml*
  - *fileRealm.properties*
  - *db_settings.properties*
  - *web.xml*

A default installation of WebLogic includes the PointBase database that is provided for demonstration and evaluation purposes only. This database is unnecessary and not required by a production WebLogic application server.

- *(N/A: CAT II) The ASA will ensure the default PointBase Server included in a default WebLogic application server installation is removed.*

A boot identity file is a text file that contains user credentials for starting and stopping an instance of a WebLogic application server. A WebLogic Administration Server can refer to this file for user credentials instead of prompting a user to provide them. Because the credentials can be encrypted, using a boot identity file is easier and more secure than storing unencrypted credentials in a startup or shutdown script.

It is necessary to create a boot.properties file and locate it in the server's root directory. The server automatically uses this file during its subsequent startup cycles. The first time this file is used to start the WebLogic application server, it reads the file and then overwrites it with an encrypted version of the username and password.

- *(N/A: CAT II) The ASA will configure the WebLogic application server's startup process to either prompt for a username and password or ensure user credentials are encrypted in the boot.properties file using NIST- validated or an NSA approved cryptography.*

If the SerializedSystemIni.dat file is destroyed or corrupted, a complete reconfiguration of the WebLogic Server domain is necessary. Therefore, the following file protections are required to successfully restore the SerializedSystemIni.dat file.

- *(N/A: CAT II) The IAO will ensure the SerializedSystemIni.dat file is included in the application server backups.*

- *(N/A: CAT II) The IAO will ensure access to the SerializedSystemIni.dat file is restricted to the ASA and the WebLogic process account.*

C.3 Installation

To properly secure the WebLogic installation it is critical to remove any services, ports, and functions that are unnecessary for the operation of the WebLogic application server.

Development tools such as WebLogic Builder and jCOM are not required for production environments and will be removed.

- *(N/A: CAT III) The ASA will ensure development tools are removed from a production WebLogic application server.*

- *(N/A: CAT I) The IAO or IAM will ensure the perimeter firewall includes rules to protect sensitive ports from exposing critical WebLogic application server functions.*

- *(N/A: CAT II) The ASA will ensure all user lockout settings for default and custom realms have been enabled.*

- *(N/A: CAT II) The ASA will enable the audit provider and set the severity level to a minimum of INFO.*

When installing a WebLogic application server, use the Configuration Wizard and select the JRE option. This option eliminates the Java compiler and other development tools that can pose a potential risk.

- *(N/A: CAT III) The ASA will ensure only the JRE version of the JVM is installed on a WebLogic application server.*

An unprivileged OS account will be created and dedicated to run an instance of the WebLogic application server. This account requires read, write and execute permissions to the BEA Home Directory, the WebLogic application server product directory tree, and the WebLogic domain directories. Restrict access to these files and directories to the ASAs and the WebLogic account. On a Windows platform, use the unprivileged account to run the WebLogic application server service.

- *(N/A: CAT II) The ASA will ensure access to the BEA Home Directory, WebLogic application server product directory tree, and domain directories are restricted to the ASA and WebLogic account.*

- *(N/A: CAT II) The ASA will ensure the WebLogic application server is run using a dedicated, non-privileged account.*

The Administration Server is the WebLogic application server used to configure and manage all the WebLogic application servers in its domain. A domain may include multiple WebLogic application server clusters and independent WebLogic application server instances. If a domain contains only one WebLogic application server, then that server is the Administration Server.

- *(N/A: CAT II) The ASA will ensure Administrative traffic between WebLogic application servers employs SSL using the secure WebLogic Administration channel.*

C.4 Configuration

To simplify the configuration and management of security in a WebLogic application server, a default security configuration is provided. It is necessary to review and modify as appropriate the default security configuration in order to comply with security requirements.

- *(N/A: CAT II) The ASA will ensure only authorized ASAs belong to the WebLogic application server's Administrator's group.*

- *(N/A: CAT II) The ASA will ensure only authorized ASAs belong to the WebLogic application server's PortalSystemAdministrator's group.*

- *(N/A: CAT II) The ASA will configure the WebLogic application server to ensure I&A is accomplished using a DOD PKI Class 3 or 4 certificate and hardware security token (when available), or an NSA-certified product.*

- *(N/A: CAT II) If a WebLogic Server provides internal I&A services, the ASA will ensure successive logon attempts are controlled using one or more of the following:*

    - *Access is denied after multiple unsuccessful logon attempts.*
    - *The number of access attempts in a given period is limited.*
    - *A time delay control system is employed.*

Connection filters are particularly useful when using the Administration port. Depending on the network firewall configuration, it may be possible to use a connection filter to further restrict administration access. A typical use might be to restrict access to the Administration port to only the servers and machines in the domain. An attacker, who gains access to a machine inside the firewall, still cannot perform administration operations unless the attacker is on one of the permitted machines. It's important when using connection filtering to provide for hostname verification sometimes disabled during installation.

- *(N/A: CAT III) The ASA will ensure hostname verification is enabled.*

The WebLogic application server domain provides protection against loss or degraded service by restricting the message sizes and time out settings. Proper setting of the T3 and HTTP attributes will reduce the risk for possible denial of service mitigated by configuring and controlling these session limits.

- *(N/A: CAT II) The ASA will ensure T3 and HTTP message sizes are set no higher than 10000000 bytes (10 MB).*

- *(N/A: CAT II) The ASA will ensure T3 and HTTP message timeouts are set no higher that 60 seconds (1 Minute).*

A WebLogic application server has the ability to restrict the content of its response header information. This means that a WebLogic application server can be configured disable the sending of the server's name and version number as a response to a given request. This information can aid an attacker's ability to discern the specific type of attacks to employ.

- *(N/A: CAT III) The ASA will ensure the Send Server Header attribute is disabled.*

APPENDIX D: LIST OF ACRONYMS

ACL	Access Control List
AIS	Automated Information System
API	Application Programming Interface
AJP	Apache JServe Protocol
ASA	Application Server Administrator
C&A	Certification and Accreditation
CAL	Category Assignment List
CCB	Configuration Control Board
CGI	Common Gateway Interface
CIO	Chief Information Officer
CJCS	Chairman of the Joint Chiefs of Staff
CJCSM	Chairman of the Joint Chiefs of Staff Manual
CLI	Command Line Interface
CM	Configuration Management
CND	Computer Network Defense
CNSS	Committee on National Security Systems
COTS	Commercial-Off-the-Shelf
DAA	Designated Approving Authority
DBMS	Database Management System
DISA	Defense Information Systems Agency
DISN	Defense Information Systems Network
DMZ	Demilitarized Zone
DOD	Department of Defense
DODI	DOD Instruction
DODD	DOD Directive
ECA	External Certification Authority
EIS	Enterprise Information System
EJB	Enterprise Java Beans
ESM	Enterprise System Management
FIPS	Federal Information Processing Standard
FSO	Field Security Operations
GID	Group ID
GOTS	Government-Off-the-Shelf
GUI	Graphical User Interface
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I&A	Identification and Authentication

IA	Information Assurance
IAM	Information Assurance Manager
IAO	Information Assurance Officer
IASE	Information Assurance Support Environment
IAVA	Information Assurance Vulnerability Alert
IAVM	Information Assurance Vulnerability Management
IECA	Interim External Certification Authority
IETF	Internet Engineering Task Force
IIS	Microsoft Internet Information Server
IT	Information Technology
JAAS	Java Authentication Architecture Service
J2EE™	Java™ 2 Platform, Enterprise Edition
J2SE™	Java™ 2 Platform, Standard Edition
JCA	Java Cryptography Architecture
JCP	Java Community Process
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JTF-GNO	Joint Task Force - Global Network Operations
JMX	Java Management Extensions
JRE	Java Runtime Environment
JSDK	Java Software Development Kit
JSM	Java Security Manager
JSP	Java Server Pages
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
MAC	Mission Assurance Category
NAT	Network Address Translation
NIAP	National Information Assurance Partnership
NIDS	Network Intrusion Detection System
NIPRNET	Non-classified Internet Protocol Router Network
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSTISSP	National Security Telecommunications and Information Systems Security Policy
OID	Object Identifier
OS	Operating System
OSS	Open Source Software
PAM	Pluggable Authentication Module
PAO	Public Affairs Office

PDI	Potential Discrepancy Item
PHP	Hypertext Preprocessor
PKE	Public Key Enabling
PKI	Public Key Infrastructure
PM	Program Manager
POC	Point of Contact
PPS	Ports, Protocols, and Services
PPSM	Ports, Protocols, and Services Management
PPSMP	Ports, Protocols, and Services Management Policy
RBAC	Role Based Access Controls
RFC	Request for Comment
S/MIME	Secure Multipurpose Internet Mail Extensions
SDID	Short Description ID
SDK	Software Development Kit
SIPRNet	Secret Internet Protocol Router Network
SP	Service Pack
SSL	Secure Sockets Layer
STIG	Security Technical Implementation Guide
TCP/IP	Transmission Control Protocol/Internet Protocol
UI	User Interface
UID	User ID
URL	Uniform Resource Locator
VMS	Vulnerability Management System
VPN	Virtual Private Network
WebDAV	Web Distributed Authoring and Versioning
XML	Extensible Markup Language

This page is intentionally left blank.