Report Number: I33-004R-2005

BEA WebLogic Platform Security Guide

Network Applications Team
of the
Systems and Network Attack Center (SNAC)

Publication Date: 4 April 2005
Version Number: 1.0

National Security Agency
ATTN: I33
9800 Savage Road
Ft. Meade, Maryland 20755-6704

410-854-6191 Commercial
410-859-6510 Fax

Acknowledgment

We thank the MITRE Corporation for its collaborative effort in the development of this guide. Working closely with our NSA representatives, the MITRE team—Ellen Laderman (task leader), Ron Couture, Perry Engle, Dan Scholten, Len LaPadula (task product manager) and Mark Metea (Security Guides Project Oversight)—generated most of the security recommendations in this guide and produced the first draft.

Warnings

Do not attempt to implement any of the settings in this guide without first testing in a non-operational environment.

This document is only a guide containing recommended security settings. It is not meant to replace well-structured policy or sound judgment. Furthermore, this guide does not address site-specific configuration issues. Care must be taken when implementing this guide to address local operational and policy concerns.

The security configuration described in this document has been tested on a Solaris system. Extra care should be taken when applying the configuration in other environments.

SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED. IN NO EVENT SHALL THE CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This document is current as of September 30, 2004.

This Page Intentionally Left Blank

Trademark Information

Table of Contents

Table of Tables

Table of Figures

This Page Intentionally Left Blank

Introduction

The purpose of this guide is to provide the reader with security configuration guidance for the BEA WebLogic Platform[1] in an operational environment, focused on the WebLogic Server portion of the product. This document is intended for knowledgeable system administrators involved with or interested in installing and configuring WebLogic Platform for an operational environment. A knowledgeable system administrator can create and manage accounts and groups, understands how operating systems perform access control, understands how to set account policies and user rights, is familiar with how to set-up auditing and read audit logs, and so on.

WARNING: This guide does not address security issues for the Sun Microsystems Solaris system and the Microsoft Windows systems that are not specifically related to WebLogic Platform.

Getting the Most from this Guide

The following list contains suggestions to ensure successful use of this guide:

WARNING: This list does not address site-specific issues and every setting or suggestion in this guide should be tested on a non-operational network.

- Read the guide in its entirety. Omitting or deleting steps can potentially lead to an unstable system and/or network that will require reconfiguration and reinstallation of software.
- Perform pre-configuration recommendations:
  - Perform a complete backup of your system before implementing any of the recommendations in this guide.
  - Ensure that the latest bug fixes have been installed.
  - Use security settings that are appropriate for your environment.

About this Guide

This document consists of the following chapters and appendixes:

Chapter 1 - Overview of WebLogic Platform provides an overview of WebLogic Platform to provide a context for discussing security guidance.

Chapter 2 – Overall Security Architecture discusses the major security elements of the WebLogic Platform.

Chapter 3 – Installation contains guidance for securely installing WebLogic Platform.

[1] For the convenience of the reader, the prefix “BEA” is not used throughout the remainder of this document. The prefix “WebLogic” is retained to avoid the ambiguities of referring simply to the server or the portal and so on.

Chapter 4 – Post-Installation Configuration contains guidance for securely configuring an operational WebLogic Platform. This section also gives guidance for using the Administration Console of WebLogic Server.

Chapter 5 - Summary recaps the security features of WebLogic Platform and the important security points made throughout the document.

Chapter

1

Overview of WebLogic Platform

This chapter describes the components of WebLogic Platform. This overview is not intended to describe the full functionality of WebLogic Platform but rather to set an adequate context for the security guidance in the remaining chapters of this guide. For a more detailed discussion of WebLogic Platform's functionality please read the BEA WebLogic documentation.[2]

The BEA WebLogic Platform is BEA's attempt to integrate application, integration, and portal technologies with the goal of reducing complexity and improving productivity. Client interaction with WebLogic Platform is either user-oriented or application-oriented. Users interact with WebLogic Platform resources via web browsers over the HTTP or HTTPS protocols. Applications interact with WebLogic Platform via host protocols.

Components of WebLogic Platform

WebLogic Platform consists of an integrated suite of extensible components for developing and hosting web-based e-commerce-oriented services.

- WebLogic Server—the central component of WebLogic Platform; it provides an application infrastructure of reusable J2EE services upon which J2EE-compliant applications can be hosted.
- WebLogic Portal—the framework for development and deployment of portal-oriented web applications
- WebLogic Integration—the component that provides support for application integration
- WebLogic Workshop—the BEA-provided development environment
- JRockit—an optimized Java Virtual Machine (JVM) for Windows

WebLogic Server

WebLogic Server, the central component of WebLogic Platform, is an implementation of the Java 2 Enterprise Edition (J2EE) standard. When properly configured, it is an application server, with web server functionality, providing services such as security, transaction management, database connectivity, legacy integration, and resource pooling. Its service-oriented architecture supports the deployment of component-based applications, namely Enterprise JavaBeans (EJB). WebLogic Server provides an application infrastructure of reusable J2EE services upon which J2EE-compliant applications can be hosted. Some of these J2EE services are:

- Java Database Connectivity (JDBC)
- Java Messaging Service (JMS)

[2] http://edocs.bea.com/platform/docs81/index.html

- Java Naming and Directory Interface (JNDI)
- Java Transaction API (JTA)
- Remote Method Invocation (RMI)

This document refers to an installed and configured WebLogic Server product as a WebLogic application server.

Figure 1 shows WebLogic application server in the context of front- and back-end components in a multi-tiered architecture.

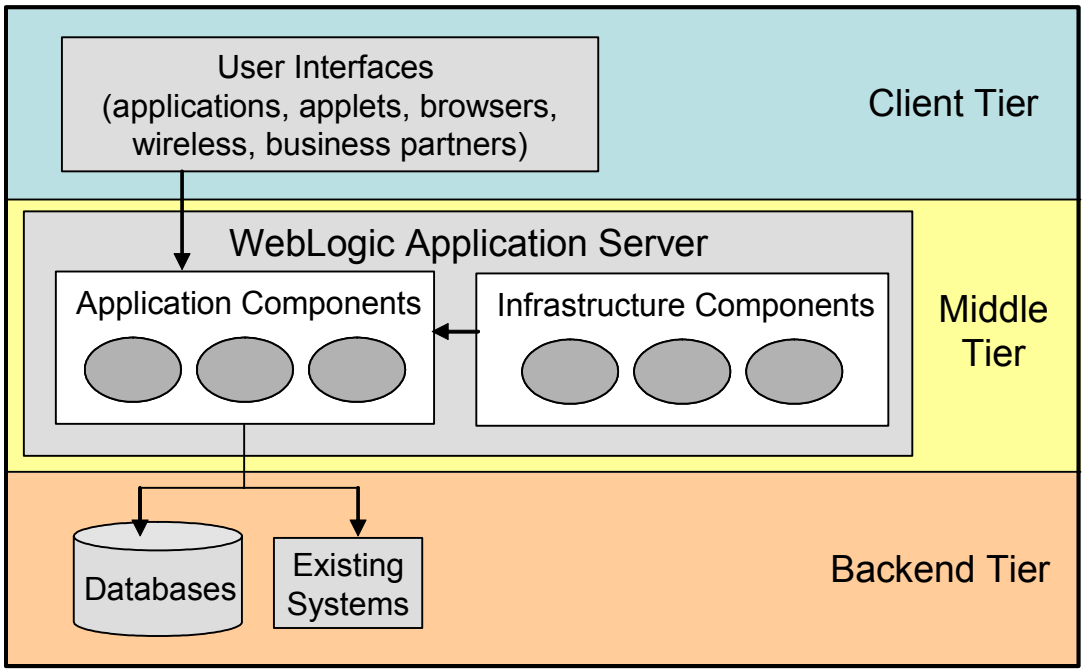

Figure 1. Typical Environment for a WebLogic Application Server

WebLogic Portal

WebLogic Portal is the framework for development and deployment of portal-oriented web applications. It supports content and appearance customization and personalization. Users can be given the ability to define the elements of functionality, called portlets, that appear on a portal page. Users can also specify where and how portlets appear on a page.

WebLogic Portal administration is conducted through **<portalname>Admin**, a browser-based tool. **<portalname>** is the name of the portal being administered. Administrators control specific portal content to specific users through user/group management and through visitor entitlement mechanisms.

WebLogic Integration

WebLogic Integration supports application integration. It creates appropriate interfaces among various preexisting or new applications so that their services are accessible to a WebLogic application server or portal. This component enables modeling, analysis, and

execution of business processes that may involve multiple internal and/or external systems or resources.

WebLogic Integration uses J2EE Connector Architecture-compliant adapters to integrate these systems and resources. J2EE Connector Architecture is a technology for connecting application servers with heterogeneous enterprise information systems (EIS). These EIS systems may include mainframe applications, enterprise resource planning (ERP) systems, database management systems (DBMSs), or legacy systems that were not developed using the Java programming language.

WebLogic Workshop

WebLogic Workshop consists of a visual development tool and a runtime framework. The visual development tool is a complete integrated development environment (IDE) that presents the developer a programming model based on events, properties, and Java controls. The runtime framework compiles the code created in the IDE and implements the appropriate J2EE components that are required to build and run the application specified in the IDE. The runtime framework incorporates much of the lower-level Java infrastructure used by web applications and web services.

In WebLogic Platform 8.1, the portal development tools formerly bundled with WebLogic Portal component have been incorporated into WebLogic Workshop.

JRockit

JRockit is BEA's optimized implementation of the Java Virtual Machine (JVM) for Microsoft Windows. WebLogic Platform comes bundled with JDK1.4, which provides a JVM for other operating systems. WebLogic Platform can also be configured to use other versions of Java.

Domains and Realms

A domain encapsulates an infrastructure of hardware and software services, encompassing physical and logical servers, groups of servers called clusters, J2EE services such as JDBC and JMS, and security services. Applications hosted within this environment are also part of the domain specification. In a WebLogic Platform implementation, a domain contains a single *administration server*, which acts as the controller for other *managed servers* in the domain.

A realm is a grouping of security relevant information and services. Within a realm, one manages information about users, groups, and roles. Realms contain security service implementations for functions such as authentication, authorization, and role mapping. A domain may contain one or more realms; however, only one realm can be active at a time.

Shared Information

WebLogic Server and WebLogic Portal depend on shared configuration data and shared dynamic data for their operation. This section identifies information that they share. The discussion is divided into two subsections: configuration data and dynamic data. The configuration data subsection contains information regarding configuration files such as fileRealm.properties and config.xml; the dynamic data section discusses dynamic information such as certificates and passwords.

Configuration Data

There are five configuration files that contain information required by WebLogic Server and WebLogic Portal. These are:

- boot.properties (or other boot identity file)
- config.xml
- fileRealm.properties
- db_settings.properties
- web.xml

There are two ways of configuring a WebLogic application server startup process. In the first way, the WebLogic application server uses a prompt to acquire the WebLogic administrator username and password required for startup. This is the recommended method for starting the server. In the other way, the WebLogic application server acquires administrator logon information from the boot.properties file. There is also an option at startup to take the username and password information from an alternative, administrator specified file called the boot identity file. This file contains the username and password of an account with privilege to start the WebLogic application server and the name of the server to start. The second method should only be considered if physical access restrictions to the server are in place. However, for the most sensitive environments, this method is not recommended. Domain properties are stored in the config.xml file. Config.xml is a hierarchical XML document containing elements and attributes associated with objects in the domain. These objects make up the domain such as servers, clusters, and applications. Each object has configurable attributes such as usernames, passwords, and listen ports.

The fileRealm.properties file was used in prior versions of WebLogic Server (6.x), to store users, groups, encrypted passwords, and ACLs. The fileRealm.properties file should be used only when managing a WebLogic application server that is based on version 6.x of the WebLogic Server product. This document addresses version 8.1 only.

The db_settings.properties file contains two types of data. The first is database properties such as DBMS product, driver, user name, and password. The second is connection properties and information needed for accessing data files for the component-level tables. The db_settings.properties file controls what SQL files to process from *WL_HOME*\portal\db.

The web.xml file is a deployment descriptor unique to each application. It defines how the WebLogic application server deploys the application. This file contains information such as how roles are mapped to resources, how applications acquire credentials from a user, and how the application maps URL requests.

Immediately after installation, these files are not read/write protected. Thus, anyone can see and modify them. Moreover, the passwords contained in these files are not encrypted, thus open to anyone who reads the files. All five files contain sensitive information that must be protected from tampering to prevent unauthorized access, loss of integrity, and denial of service. Protect these files with the operating system's capabilities to grant and deny security rights.

Dynamic Data

Some data communicated between WebLogic Portal and WebLogic Server is too dynamic to store in a configuration file. This information includes the following such data:

- Certificates – used in Secure Sockets Layer (SSL) protocol.
- Username – used to identify the entity seeking access.

- Passwords – used to authenticate users seeking access.
- Groups – sets of users with similar privileges.
- Roles - used to manage the relationship between groups and resources.

Groups, Roles and other static data are stored in the database provided with the WebLogic Platform. The dynamic data listed above may also be stored in this database. Most organizations will opt to use databases that are already part of their enterprise infrastructure. This data is sensitive and therefore should be protected from unauthorized access. The use of the above data is explored more fully in Chapter 2.

External Services

This section discusses external services, both required and optional, used by WebLogic Server and WebLogic Portal. External services are those services not intrinsically provided by WebLogic Platform—that is, services not provided by one of the five components of WebLogic Platform. Required services are those services without which WebLogic Server and WebLogic Portal will not function. Optional services are those that an application running on WebLogic application server may use, such as other web services, directory services, and databases.

WARNING: Ideally, external services will be locked down by the owners/operators of the services. However, the WebLogic administrator should not assume that services under others' control have been locked down. They may, in fact, be unsecured.

It is recommended that WebLogic administrators lock down services under their control and take measures to counter the threat posed by other, potentially unsecured services.

Following are descriptions of required and optional services. For each service a description is provided along with details for how the service interacts with the WebLogic Server and WebLogic Portal. Where practical, advice or a reference on how to lock down the service is also provided.

Required External Services

Services required for the operation of WebLogic Platform components are

- An underlying operating system
- Java libraries and utilities
- A web browser
- A DBMS

Underlying Operating System

The operating system provides fundamental services such as file system management, communications, and some basic security services (e.g., authentication, logging, and access control). WebLogic Platform is supported on a number of UNIX platforms including Solaris, HP-UX, and AIX, as well as on Linux and Windows platforms. The operating system used must be locked down using the appropriate guide. Security guidance for several of these operating systems is available at http://www.cisecurity.org and http://www.nsa.gov.

Java Libraries and Utilities

WebLogic Platform 8.1 requires Java libraries and utilities because it implements J2EE. WebLogic Platform comes bundled with JDK1.4; however, it can be configured to use any other version of Java. JDK1.4 provides a Java Virtual Machine so that a variety of operating systems can be supported, including Windows platforms. Additionally, for the Windows platform, BEA has developed the optimized Java Virtual Machine called JRockit.

Web Browser

The WebLogic Platform administrator uses a web browser to manage the configuration of WebLogic Platform components. For WebLogic Portal, the administrator uses a browser to access **<portalname>Admin***,* the browser-based tool provided with WebLogic Platform for enabling configuration, including security configuration, of deployed portal web applications. For WebLogic Server, the administrator uses a browser to access the Administration Console, which offers the administrator a wide range of management capabilities such as server configuration, deployed application configuration, J2EE services configuration, and security settings (including management of security service provider[3] components). WebLogic Integration and WebLogic Workshop support use of a web browser for configuration purposes. A web browser is also the method by which external users interact with portal applications. WebLogic Platform supports Microsoft Internet Explorer, Mozilla, and Netscape Navigator browsers.

Database Management System (DBMS)

WebLogic Server requires a DBMS to store data, for example product catalog items, information relating to content management, and data resulting from event and behavior tracking. WebLogic Platform ships with an embedded all-Java PointBase DBMS; however, this database is provided strictly for evaluation and tutorial purposes. Thus, for production deployments, an alternate secure DBMS is strongly recommended.

WebLogic Platform has native support for the following DBMSs:

- Oracle
- Sybase
- SQL Server
- DB2

Security guidance for Oracle 9i [CHRISHAY-2] and SQL Server [CHRISHAY-1] is available at http://www.cisecurity.org and http://www.nsa.gov

Optional External Services

A wide variety of optional external services are available. The services can be grouped into three categories

- Web Services—applications distributed on the web that cooperatively provide a service through the use of standard technologies
- Directory Services—hierarchically organized directory information storage
- Databases—data information storage

[3] Security providers are modules that interface with a WebLogic security realm to provide security services to applications.

Web Services

A web service provides functionality through a URI that is accessed by an application. In general, a web service is a combination of application and infrastructure elements spread across multiple computers on a network. The web service makes available one or more applications running on one or more computers attached to the network. This is made possible by web services technology, which uses the following recognized standards:

- Web Service Description Language (WSDL)—a standard format for describing web services
- eXtensible Markup Language (XML)—a common language by which different applications may communicate with one another over a network
- Simple Object Access Protocol (SOAP)—a standard format for applications to call each other's methods and pass data to one another
- Network protocols such as Hyper Text Transfer Protocol (HTTP) and Java Messaging Service (JMS)—network protocols used to communicate requests and responses between two network entities

The components of a web service, residing on various computers, can be implemented in different programming languages and technologies. Thus, a program written in C# .NET operating on a Windows platform and a program written in Java operating on a Solaris platform can cooperate to provide a web service to a client.

As shown in Figure 2 below, the components of a web service communicate with clients and resources using SOAP-formatted XML via HTTP. A WebLogic application server is responsible for routing incoming XML messages to the web service code. The component of the web service residing on the WebLogic application server exports a WSDL file describing its interface. Developers can use the WSDL file to write components that access the service on the WebLogic application server.

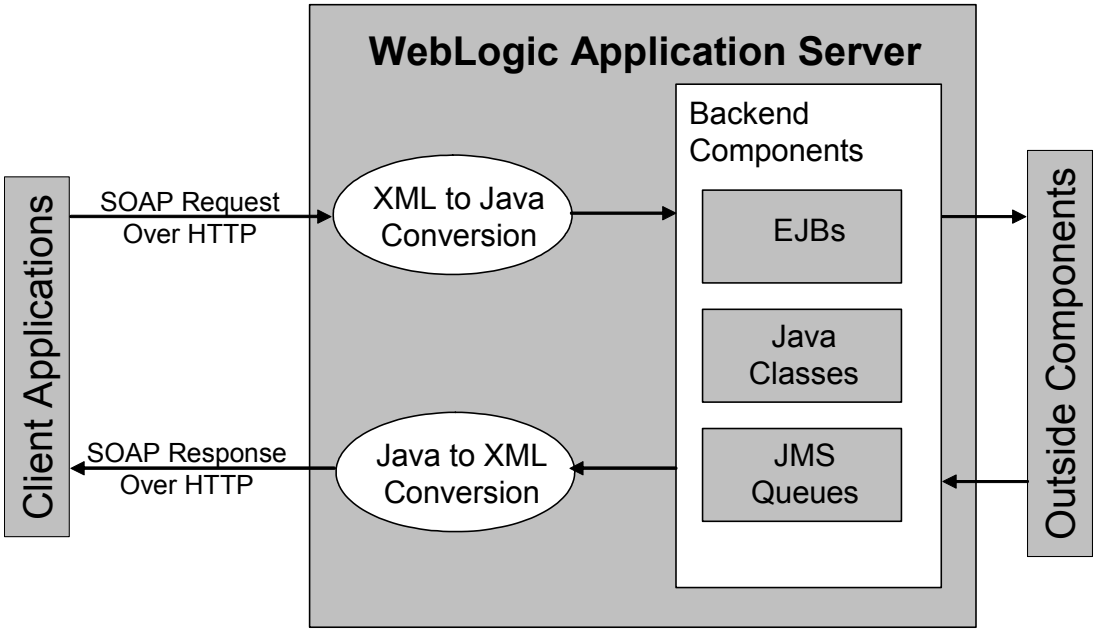

Figure 2. Web Service Architecture

Directory Services

WebLogic Platform provides an embedded LDAP Directory Service as well as the ability to interface with external directory services. The embedded LDAP Directory Service is the default database for authentication, authorization, credential mapping, and role mapping.

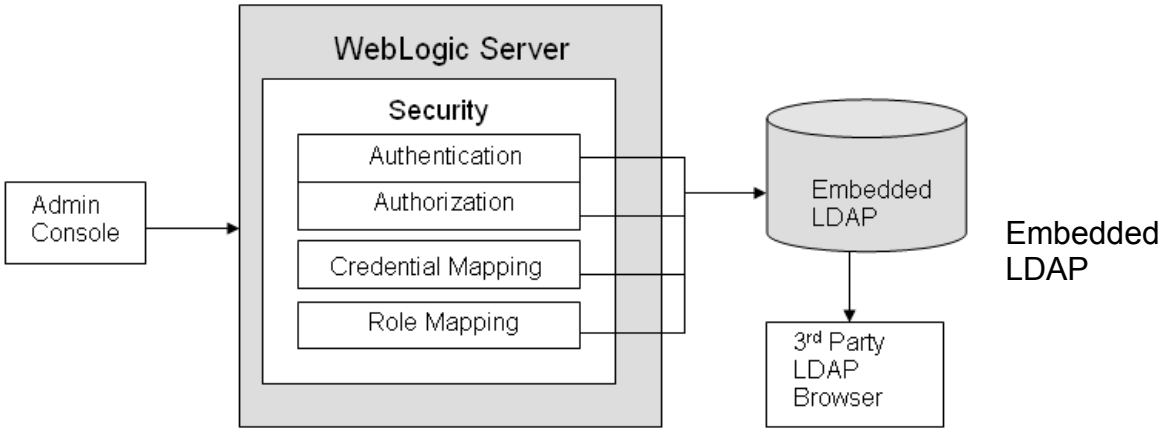

Figure 3. WebLogic Platform Interaction with the Directory Service

Directory services store the following types of information:

- Users
- Groups
- Group membership
- Security roles
- Security policy
- Credential map

The WebLogic Server administrator is able to set the properties for the directory service, via the Administration Console. This is the case for both the embedded LDAP Directory Service and any external directory services. The property information is stored in a WebLogic application server's *config.xml* file. Each domain has a different *config.xml* file with properties particular to that domain. Any third party directory service browser is able to connect to the embedded LDAP server to view and/or modify its contents. Sensitive directory data, such as passwords, stored in a *config.xml* file is encrypted using 3DES. Access to the sensitive data requires the use of credentials established by the WebLogic Server administrator.

NOTE: Consider changing the LDAP port number so that how to access the LDAP server is not obvious to outsiders.

Databases

Most web applications hosted by WebLogic Platform use a DBMS to store their application-level data. The DBMS can be the same one used by WebLogic Platform or any database that supports JDBC.

Important Security Points

- Configure the WebLogic application server's startup process to use a prompt to acquire username and password and, if possible, provide physical security protection for the server.
- Use the fileRealm.properties file only when managing a WebLogic application server that is based on version 6.x of the WebLogic Server product.
- Protect the five configuration files required by WebLogic Server and WebLogic Portal from tampering.
- Protect the data of the database used with WebLogic Platform from unauthorized access.
- Lock down external services used by WebLogic under your control and take measures to counter the threat posed by other, potentially unsecured external services.
- Lock down the underlying operating system used with WebLogic Platform; use the guide or benchmark appropriate for the operating system.
- Use a robust DBMS. NOTE: The PointBase DBMS provided with the WebLogic Platform is strictly for evaluation and tutorial purposes. Non-evaluation use of the PointBase Server requires a separate license be obtained from PointBase.
- Secure the DBMS that is used with WebLogic Platform in accordance with an appropriate guide or benchmark, such as the NSA Guide to the Secure Configuration and Administration of Oracle 9i Database Server. [CHRISHAY-2]

Chapter

2

Overall Security Architecture

This chapter has two parts: the first part describes the security architecture of WebLogic Platform; the second part describes how IA capabilities[4] are implemented using this architecture.

Most of the security functionality is located within a WebLogic application server and it mediates security related interactions for the rest of the WebLogic Platform. Although there are some security features in a portal, most of the security functionality accessible through a portal is implemented in a WebLogic application server. Thus, most of the discussion in this guide deals with WebLogic application servers.

Platform Security Architecture

Historically, it was left to middleware developers to integrate each application to each type of security component. This approach made IA implementation more difficult in three ways. First, each application might need different security components for application specific forms of authentication, access controls, and identity stores. The result is multiple combinations of application-to-security functionality mappings. This increases development and maintenance costs and unnecessarily increases the risk of security vulnerabilities. Second, security policy was often embedded within business logic. This created a situation where the policy was developed and maintained within the application, duplicating effort and increasing maintenance. Finally, because security policy was implemented in both infrastructure and application components, security expertise was not used consistently. BEA addressed these issues by creating a framework for consistently integrating security with applications while avoiding duplication of effort.

The WebLogic Platform is a component in a BEA WebLogic Enterprise. A typical BEA WebLogic Enterprise can be composed of clusters of WebLogic application servers, databases, directory servers, legacy system interfaces and remote application interfaces. The security architecture of the WebLogic Platform is integral to the security architecture of the WebLogic Enterprise. WebLogic Enterprise security is beyond the scope of this document. For more information please refer to *BEA WebLogic Enterprise Security: An Introduction to BEA WebLogic Enterprise Security*. [BEA-2]

Figure 4 is a simplified view of security in the BEA WebLogic Enterprise. Only one domain, one realm, and one WebLogic application server are shown. In a production enterprise, there may be multiple WebLogic application servers installed on a variety of host operating systems each with its own set or sets of security service providers (SSPs). WebLogic Server provides the Administration Console to administer all WebLogic application servers in an enterprise using a unifying security policy.

Security in WebLogic Server is based on the Secure Sockets Layer (SSL) protocol, certificates, internal SSPs, and external SSPs. When a user, via a web browser, requests access to one of the applications for which a WebLogic application server provides an

[4] The capabilities are those listed in DoD Directive 8500.1. [1]

interface, the WebLogic application server uses SSL and a digital certificate, signed by a trusted certificate authority (CA), to authenticate itself to the browser. If the authentication is two-way, an option in SSL, the browser uses SSL and its certificate to authenticate itself to the server. Next the user must be authenticated to verify the user's identity. Then a role mapper is invoked to determine a set of valid roles for the user, such as "administrator" or "guest." After this, authorization providers are executed to ascertain if the user is authorized to access that resource. If there is more than one authorization provider, an adjudication provider will determine which authorization provider's decision should be used. If the user wants to access a remote system through the WebLogic application server, the credential mapping allows the WebLogic application server to log in to that remote system on the user's behalf. The WebLogic application server uses its own SSPs or external SSPs to perform functions such as credential mapping, authentication, authorization, and role mapping. If an external SSP is used, that SSP must also authenticate to the WebLogic application server using SSL. Services, such as Java Database Connectivity (JDBC), Web Services (WS), Remote Method Invocation (RMI), or Java Messaging Services (JMS), can be configured to invoke the SSPs and SSL via the WebLogic application server. The Identity Assertion SSP is a special type of Authentication SSP and the Principal Validation SSP is a helper SSP to the Authentication SSP. Both of these SSPs will be discussed in the subsection that discusses the Authentication SSP. More details about the function of the individual SSPs and SSL are given below. Details relating to installation and configuration are given in chapters 3 and 4.

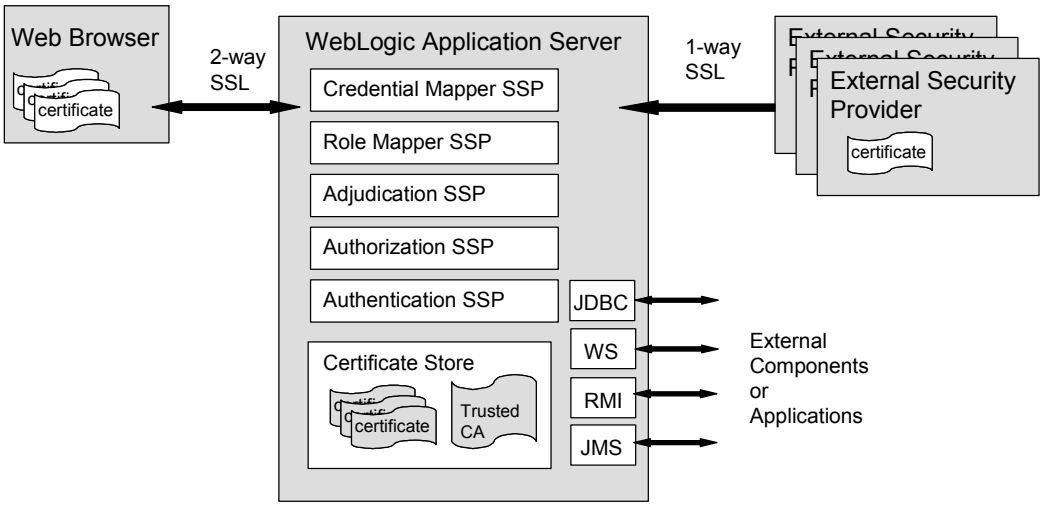

Figure 4. Enterprise Security

The following subsections discuss the two main aspects of WebLogic Platform security—SSPs and SSL.

Security Service Providers

Security service providers (SSPs) in the BEA WebLogic architecture are modules that provide information assurance functionality. They interface with a WebLogic security realm to provide security services to applications by calling into the WebLogic security framework on behalf of applications.

BEA provides a default set of these modules with WebLogic Platform. These default SSPs can be supplemented with or replaced by providers that are custom created or those that come from third parties. WebLogic Platform provides the following SSPs:

- Authentication Provider
  - Identity Assertion Provider, a special type of Authentication Provider
  - Principal Validation Provider, a helper to the Authentication Provider
- Role Mapping Provider
- Credential Mapping Provider
- Authorization Provider
- Adjudication Provider
- Auditing Provider
- KeyStore Provider (for backward compatibility with WebLogic 6.x)
- Realm Adapter Authentication Provider (for backward compatibility with WebLogic 6.x)

Authentication

The authentication provider is responsible for verifying the identity of users and other system entities such as processes or remote computers. The authentication provider makes identity information available to other components on the platform as needed. Authentication providers within the WebLogic Platform Security Architecture can perform authentication in three different ways

- Username/password authentication
- Certificate-based authentication directly with the WebLogic application server
- HTTP certificate-based authentication via an external web server

Authentication performed via web servers, a form of perimeter authentication, is supported in the WebLogic Platform Security Architecture through a specific form of authentication called identity assertion, which is a means of checking that a user is listed as a trusted user. Multiple identity assertion providers can be configured in a security realm. If no perimeter authentication is required, no identity assertion provider is required.

When one or more applications within the WebLogic enterprise enclave require that a user or group be authenticated, the principal validation provider can vouch for that user or group; the user does not need to repeatedly authenticate to every application within the enclave. The principal validation provider, in conjunction with the identity assertion provider, essentially is the means by which single sign-on functionality is provided within the enterprise. Note that, unlike the principal validation provider, the identity assertion provider does not validate the user; it does not sign and/or verify user credentials. Instead, it merely determines that the user is in fact listed as a trusted user. It is the token that provides proof that a user is trusted.

Role Mapping

The role-mapping provider is sent information regarding the user and the resource requested by the user. It then dynamically determines the set of roles that are valid for that user with respect to the targeted resource. A role-mapping provider dynamically obtains a set of security roles granted to a requestor for a given WebLogic resource. These security roles are obtained from the J2EE and WebLogic descriptor files and from business logic and current operation parameters.[5] If security policy specifies that the

[5] *Types of Security Providers* www.bea.com

requester is permitted to assume a certain role, then that role is dynamically added to the list of roles that are valid for that user. Each security realm requires at least one role-mapping provider. A security realm may have more than one role-mapping provider (for example, one for each LDAP server). If more than one role-mapping provider is configured, the security role names from all role-mapping providers are merged into a single list with duplicates removed. This process continues until all security policies that apply to the resource are evaluated. The authorization provider uses this information by checking the role against policy to determine authorization.

Credential Mapping

Credential mapping allows a WebLogic application server to log into remote systems on behalf of a user or computer that has already been authenticated. The credential-mapping provider provides a means to map the WebLogic-approved credentials to the set of credentials required by the remote system. Credential mappings can be set in deployment descriptors or by using the Administration Console. Credential mapping providers can handle several different credentials such as username/password combinations, Kerberos tickets, and public key certificates. Each security realm must have at least one credential-mapping provider. If multiple credential mapping providers are configured, a list of all credentials in all the credential-mapping providers is returned when a request for credentials is made.

Authorization

The authorization provider determines whether a resource can be accessed, based on the role of the user. Authorization can be granted based on individual user credentials as well. BEA claims that performance can be greatly increased if permissions are based on the user's role rather than the user's credentials because the users do not need to be re-authenticated to access additional resources. Groups of users can be assigned to a given role if their business rules are the same, making role-based authorization highly scaleable and thereby making this aspect of security simpler to manage.

Each security realm must have one authorization provider. However, for a more modular design, more than one authorization provider can be configured. For example, you would use more than one authorization provider if you wanted to have separate authorization providers for different groups of users. The authorization providers return one of three answers: **permit**, **deny** or **abstain**. If there is only one authorization provider, **abstain** is treated as **deny**.

Adjudication

When multiple authorization providers are configured within a realm, an adjudication provider is required. Each authorization provider may return a different answer to the question "is access allowed?" Each provider can return PERMIT, DENY, or ABSTAIN. The adjudication provider's function is to resolve the conflict.

The default adjudication provider bundled with WebLogic provides a **Require Unanimous Permit** attribute that can be set to TRUE or FALSE. The WebLogic Adjudication provider behaves as follows:

- If all authorization providers return PERMIT, then PERMIT
- If any authorization providers return DENY, then DENY
- If some authorization providers return ABSTAIN and others return PERMIT, then PERMIT if **Require Unanimous Permit** is FALSE
- DENY otherwise

Auditing

Auditing is the means by which system events are recorded. The decision to audit an event is made by the auditing provider based on administrator configuration parameters. It can be enabled to trigger before or after events such as authentication and authorization operations. Additionally, it can be configured based on specific criteria and severity levels.

The auditing provider supports the Apache Open Source log4j framework. Data can be written to an LDAP store, database table, or plain file. BEA or one of a variety of third-party report generating software may be used to generate reports from the stored audit data.

KeyStore

The KeyStore provider can be used to obtain secure private keys from keystores. The KeyStore provider is deprecated and is supported in WebLogic Server 8.1 only for compatibility with WebLogic Server 7.0. Custom KeyStore provider development is not supported. The functionality provided by the KeyStore provider is available through Java KeyStores. For more information please see e-docs bea.com/wls/docs81/secintro/realm_chap.html.

Realm Adapter

The realm adapter authentication provider maps the realm API used in WebLogic Server 6.x to the API used in WebLogic Server 8.1, allowing the use of existing WebLogic 6.x security realms with the features in WebLogic Server 8.1. In particular, it enables version 8.1 to use users and groups from version 6.x security realms.

SSL

Secure Sockets Layer (SSL) provides technology to implement IA characteristics such as authentication, integrity, and confidentiality. It does this by using digital certificates, hashing, and encryption.

WebLogic Platform supports SSL to protect confidentiality and integrity and to increase confidence in user and system authentication for a number of different protocols.

Since WebLogic Server is primarily a web-based technology, its primary protocol is Hypertext Transport Protocol (HTTP), the protocol used by the World Wide Web for communication between the web browsers and the web servers. HTTP is a plain text protocol, which can be eavesdropped by anyone with network monitoring equipment and access to the network between the two systems. Network monitoring equipment can view all plaintext data passing in both directions between a browser and a server, including usernames and passwords. This data could be collected for use by adversaries, or diverted, changed, and reinserted into the system network traffic[6] to disrupt operations.

SSL was created to establish privacy and increase trust for commercial web transactions over HTTP, called HTTPS (for HTTP-Secure). It does this by encrypting data using cryptographically signed digital certificates containing public keys. The certificates are passed during the initial SSL handshake between the server and the client. Digital certificates signed by a trusted third party, called a certificate authority (CA), can be trusted by the receiver of the certificate to be authentic. When the signed certificate is passed from the server to the client only, the usual occurrence in HTTPS, the trust relationship established by the certificate is called a “one-way trust”. In a one-way trust relationship, the client trusts the server, but not vice versa. If, in addition, the client sends

[6] This type of attack is called a “man in the middle attack”.

a signed certificate to the server during handshake, a “two-way trust” relationship is established, in which the client and the server trust each other.

WebLogic Server supports SSL on several different protocols including HTTP, Lightweight Directory Access Protocol (LDAP), and Java Authentication and Authorization Service (JAAS).

IA Characteristics and the Enterprise Security Architecture

Information Assurance (IA) is a set of measures intended to protect and defend information and information systems. They can be divided into five areas: **authentication**, **integrity**, **confidentiality**, **non-repudiation, and availability**.[7]

In order to effectively apply IA measures, one must have a clear understanding of what IA is, where IA measures are applied, and the risk environment for the information system under consideration. The risk environment is defined by the sensitivity of the information or information system, the threat, and risk management decisions.

This subsection provides information about IA and discusses what WebLogic provides and in what areas it relies on external resources. As with most other applications today, WebLogic provides some IA measures internally and relies on external resources for the rest.

Authentication

As discussed in the first part of chapter 2, the particular authentication method used depends on the configuration of the WebLogic application server and the way in which access to it is requested. Users can be authenticated via username and password, certificates, or perimeter authentication. SSL and HTTPS can be used with username and password authentication, in their one-way authentication mode, to provide additional security. In certificate authentication, SSL and HTTPS are used in their two-way authentication mode. Perimeter authentication is the process of authenticating a user that is outside the application domain. Under perimeter authentication, the user authenticates to some entity within the WebLogic environment, such as a firewall or enterprise authentication service, and this entity then generates a special token which is used to vouch for that user within the WebLogic environment.

Integrity

In the WebLogic Platform, integrity of data is preserved through encryption provided by SSL.

Confidentiality

WebLogic implements confidentiality via encryption and access control technologies.

Encryption

There are various technologies available to perform encryption, including symmetrical and asymmetrical keys, and several algorithms of varying strengths. WebLogic offers encryption technologies to achieve confidentiality when passing sensitive data such as passwords.

[7] http://www.nsa.gov/about/about00019.cfm, July 2004.

Access Control

WebLogic implements access control using the authorization provider and the role-mapping provider. Additionally, when multiple authorization providers are deployed, the adjudication provider is also required.

Non-Repudiation

Digital Certificates

Asymmetric keys support non-repudiation since senders and receivers use private keys rather than a shared key. WebLogic provides non-repudiation assurance through the support of digital certificates.

Auditing

WebLogic provides the capability to electronically record a large variety of system events. Recording events in a log supports non-repudiation by making it difficult for senders and receivers to disavow transmission and receipt of messages. BEA WebLogic auditing, including support for log4j[8], or third-party auditing functionality can be employed for this purpose. Logs can be stored in repositories such as flat file, databases, or LDAP servers.

Availability

WebLogic relies on a defense-in-depth approach to availability. That is, by hiding services behind protection devices such as firewalls, distributing functions across platforms, and using the backup functionality of the operating system, an administrator can decrease the chances of catastrophic downtime.

[8] Log4j is an open source tool developed for putting log statements into applications. It was developed at Apache's Jakarta Project. Its speed and flexibility allows log statements to remain in shipped code while giving the user the ability to enable logging at run-time without modifying any of the application binary. It accomplishes this without incurring a high performance cost. From http://www.jguru.com/faq/Log4j as of July 2004.

This Page Intentionally Left Blank

Chapter

3

Installation

Securing a WebLogic Platform installation involves not only securing the product itself, but all other components that comprise the environment in which WebLogic Platform operates, including underlying operating systems, databases, LDAP servers, and so on.

This chapter discusses how to install the WebLogic Platform so that it is securely configured. This discussion is not intended to replace lockdown procedures for host machines and supporting applications but rather to provide security guidance on risks and mitigations associated with installing WebLogic Platform.

There are four sections. The first section discusses lockdown of the host system and support applications so that the installation takes place on a secure base. The second section discusses installation of the WebLogic Platform software. The third section discusses the notion of Domains and Realms and how they are related to security concerns and makes security recommendations regarding them. The fourth section discusses security service providers and the installation issues related to installing them securely.

Base Lockdown

General Guidelines

The first consideration in securing an installation is one of physical security. Place the computing resources that will host WebLogic Platform in an area where they will be physically secure (e.g., behind a locked door where only authorized personnel can gain physical access to it).

Since a system administrator has no need to access information associated with the applications being provided on the WebLogic application server, install WebLogic Platform on its own dedicated host, not one that supports other services such as database service. Do not store development tools on the production system because of the damage that could be caused by careless use of those tools.

If Internet access to the WebLogic application server is required, protect the WebLogic Platform from incoming traffic during setup and configuration. Install it initially with all incoming traffic blocked in some manner. The method by which this can be accomplished depends on the networking environment and business constraints. Some ideas are

- Intranet Firewall can block traffic based on IP address or other parameters
- Host-based Firewall can block incoming traffic on specified ports
- Router access control can be modified to block traffic based on IP address
- Services can be turned off on the host system
- The network connection can be disabled
- A shunt application can be provided to accept incoming traffic and delete it

- An application proxy that can monitor layer 7 traffic

Once the configuration of the WebLogic application server is complete, you may allow incoming traffic. This is recommended to prevent connections to active services of the WebLogic application server before securely configuring it.

For intranet service in a Windows environment, install WebLogic application server on a domain member server, where possible. Do not install it on a domain controller. If WebLogic application server was to be installed on a domain controller and the server is attacked, the entire domain and sensitive domain information may be at risk.

If the domain[9] where WebLogic application server is to be installed requires trust links to other domains, ensure that the access granted to domain resources through this trust is what was intended. It is important to document domain trust relationships and the resource access/permissions associated with the trust.

If the WebLogic application server will be accessed from the Internet, consider configuring a DMZ for the WebLogic application server. If possible, only store information that is meant for public dissemination on the WebLogic application server. Also, do not configure trust links back to the internal domain from the DMZ. Several sources are available on the internet describing DMZ architectures. The “Microsoft Windows 2000 Architecture Guide” is a good resource for this information and can be found on the www.nsa.gov web site.

Implement best practice account and password management. Change account passwords every 90 days. Use complex passwords for accounts that will be used with certificate mapping; use a password complexity scheme, such as SNAC Seeder[10], to achieve this. Do not use default accounts for administering a WebLogic application server.

Operating System Guidelines

Windows

Before installing WebLogic Platform on a Microsoft Windows 2000 or 2003 server, apply the security guidance of the applicable NSA or Microsoft guide to the operating system (Windows Server 2000 or Windows Server 2003) used by your installation. [NSA-3, MS] If you are using 2000 server, apply NSA’s W2K Server.inf file (available for download, see NSA-3 reference) after modifying it appropriately to reflect the enterprise’s security policy.

Install Windows server on its own drive or partition and apply the latest service packs and hotfixes. Disable the following services on the Windows platform if feasible, but configure in a test environment to ensure that operational capability is not affected before disabling them:

- Alerter
- Clipbook Server
- Computer Browser
- DHCP Client
- Distributed File System
- Fax Service
- Internet Connection sharing

[9] In this context, the term “domain” is being used in its general sense, not restricted to a Windows domain.

[10] This tool is described in the Guide to the Secure Configuration of Oracle 9i Database Server available at http://nsa1.www.conxion.com/support/guides/sd-11.pdf

- IPSEC policy agent (disable unless IPSEC policies will be used)
- License Logging Service
- Logical Disk Manager Administrator
- Messenger
- NetMeeting Remote Desktop Sharing
- Network DDE
- Network DDE DSDM
- Print Spooler
- Remote Access Auto Connection Manager
- Remote Access Connection Manager
- Remote Registry Service (disable unless running HFNetCheck or other network management software requiring remote registry access)
- Removable Storage
- RunAS Service
- Smart Card
- Smart Card Helper
- Task Scheduler (disable unless batch jobs will be run from within the WebLogic application server or if scheduling tasks on the server is required)
- Telephony
- Telnet
- Windows Installer

Remove all unnecessary protocol stacks (do not remove TCP/IP). Rename the local computer's Administrator account. Remove the **Everyone** group from the access control list on the installation drive or partition. Give the system account and the administrators group full control over the installation drive or partition. Enable Screensaver with wait time of 15 minutes and password protection.

UNIX

Apply the NSA security guidance to the operating system. Although this subsection discusses securing the Sun Solaris operating system, the principles are applicable to any UNIX-based operating system. Information for securing the Sun Solaris 8 and 9 operating systems is available in *Guide to the Secure Configuration of Solaris 8*. [NSA-1]

WebLogic Platform Installation

Installation of WebLogic Platform is straightforward, involving only the specification of a home directory and selection of the components to be installed.

Isolate the WebLogic Platform software from other software. Install on a partition or disk separate from the partition or disk containing the operating system software. No other software should be installed on the machine that hosts WebLogic Platform.

The platform software distribution offers development tools and numerous example and tutorial components that are not appropriate for a production environment. Select Custom installation when queried during the install process. All components are selected by default. For WebLogic Workshop, development tools are selected by default—-Workshop runtime framework, Workshop application developer edition, Workshop integration extensions, and Workshop portal extensions. Deselect each of the Workshop development tools and all sample, example, tutorial, and tour entries.

Domains and Realms

BEA defines a domain to be a single administrative unit for WebLogic Platform applications. A domain can be configured to contain one or more servers, clusters, hosts, applications, and security realms. When creating a domain, WebLogic application server establishes a default security realm. A realm is used in the management of users, groups, roles, and security service providers. WebLogic application server administrators are free to create additional realms within a domain; however, only a single realm can be active at any given time.

A number of domain-wide security settings require configuring after a new domain is created. These actions are implemented using the Administration Console.

- With the domain active (administrative and managed WebLogic application servers running) open the Administration Console and select the 'Security' tab in the left pane. Navigate to 'Configuration'/'Embedded LDAP' tab. Change the credential. This is the master password used in managing the embedded LDAP server and its data. It is stored in a hashed 3DES format in the config.xml file for the domain.
- In the same window navigate to 'Configuration'/'General' tab. Uncheck the **Anonymous Admin Lookup Enabled** checkbox.
- To establish trust relationships with other domains, navigate to 'Configuration'/'Advanced' a credential can be entered. When establishing trust relationships, use the same credential for each domain that will be involved in the trust relationships. The trust relationships will break otherwise, leading to denial of service.

Some security settings apply at the realm level. Using the Administration Console, select the default 'myrealm' folder in the left pane. Select the 'User Lockout' tab in the right pane. Verify that "Lockout Enabled" is checked. Modify other lockout settings based on local policy. Repeat this process for any custom realms.

A domain may contain one or more WebLogic application servers. Each server runs in its own Java Virtual Machine. For each server in the domain, configure the SSL port numbers. Select the server under the **Servers** folder in the left pane of the Administration Console. Navigate to the **Configuration**/**General** tab in the right pane. Turn on the **SSL Listen Port Enabled** checkboxes—one for enabling an HTTP listen port and one for enabling an HTTPS listen port. Corresponding to each checkbox is a textbox for specifying a port number. Enter appropriate port numbers in the text fields labeled SSL Listen Port.

NOTE: If you do not provide port numbers, default ports 7001 (for HTTP) and 7002 (for HTTPS) are used. If there is the potential for conflict in the use of these ports by other applications or services, you should enter alternative port numbers.

Security Service Providers

As discussed in Chapter 2, WebLogic Platform security service providers offer a modular, flexible approach to incorporating security functionality. The default security realm initially is configured to use the default providers that are bundled with WebLogic Platform.

Default providers exist for adjudication, authentication, identity assertion, authorization, credential mapping, and role mapping.

There is no auditing provider configured initially. This is done for performance reasons. If there is a need to collect information about security related events, for the purposes of non-repudiation, create a new auditor and set the severity level of events for which data will be collected. This can be done using the Administration Console: navigate to the 'Auditing' folder below the 'Security Providers' folder for the realm of interest. Note that event filtering is based on event severity. There are no provisions to configure auditing based on event type.

The WebLogic Platform can be deployed with custom authentication providers as well as default authentication providers. Although user credentials for authentication can be stored in WebLogic application server's embedded LDAP server, use of an external enterprise directory service is recommended. Most organizations will already have this data stored in some other enterprise resource. In addition, the embedded LDAP server does not provide the scalability and performance that might be expected and available with other products. Finally, as suggested elsewhere in this document, services additional to the WebLogic application services should not be executing on the same platform as the WebLogic application server. WebLogic application server supports interfaces to a number of the more prominent product types or data store technologies in use today; these include LDAP, Active Directory, and RDBMS.

To establish an additional or alternate directory service, use the Administration Console. Navigate to the 'Authentication' folder below the 'Security Providers' folder for the realm of interest. Choose the appropriate link for the type of product being configured. Complete required information for the data store including hostname where the store resides, port, and credential used to bind to the store. Make sure the **SSL enabled** checkbox is selected. If this will be the only authentication provider used by the domain, make sure it also contains the WebLogic Platform 'Administrator' account information or it will not be possible to boot the servers. See the BEA administration and security documentation for more details.

Some security considerations pertain to Active Directory in a Windows environment. If the WebLogic application server account uses Active Directory to authenticate against when starting the WebLogic application server, take the following actions:

- Deny Logon Locally rights to the account used to start the WebLogic application server. This will prevent someone from trying to use the account to login to the Windows server. Configuring the account to only logon to the WebLogic application server may prevent someone from trying to use the account to gain access through other hosts in the domain.
- Create a global group for the WebLogic startup account. After adding the WebLogic startup account to this group, change the Primary group for the WebLogic startup account to the global group you just created. The default Primary group for any accounts that are created is set to Domain Users. You must remove the WebLogic startup account from the Domain Users group. This is done to prevent the WebLogic application server startup account from gaining privileges and permissions granted to domain users.
- When using the default Active Directory Authenticator, use IPSec to secure the connection between the WebLogic application server and Active Directory rather than TLS/SSL. Guidance can be found in NSA's "Microsoft Windows 2000 IPSec Guide". [NSA-2]

WebLogic application server allows an LDAP server to be identified by either hostname or IP address. Set the host field to the fully qualified domain name (FQDN) when using

SSL or TLS to connect to a directory server using one of WebLogic's default directory authenticators.

If your configuration will require other web or application servers to establish trust relationships with the WebLogic application server, then an identity assertion provider needs to be configured. Trust relationships are based on the use of tokens rather than username/password. A number of different token types are supported. Navigate to the 'Authentication' folder under the 'Providers' folder for the realm of interest. Choose the DefaultIdentityAsserter or the link to configure a new identity asserter. Enter appropriate information such as 'user name mapper class' and the token type to be used.

If the default Identity Asserter is used with the X.509 token type, be careful to avoid user identity ambiguity. Ensure that the attribute value that is used to map from X.509 certificates to user accounts is unique for each user within the CA's domain of users. For example, suppose that the default user name-mapper attribute type is the e-mail address and the delimiter is the symbol @. Then, if UserA in the XYZ organization and UserA in the ZYX organization are issued certificates by CA MYCA, ambiguity results because both attribute values map to UserA. The result is that UserA@zyx.gov, who is not a user with XYZ, can use his certificate to access resources on XYZ's WebLogic application server. In this example, use the entire e-mail address when mapping to user accounts to avoid the ambiguity.

Important Security Points

- ❑ Physically secure the computing resources that host WebLogic Platform.
- ❑ Install WebLogic Platform on its own dedicated host system, not one that supports other services such as database service.
- ❑ Ensure that no development tools are stored on the system that hosts WebLogic Platform.

This group of recommendations is relevant when the WebLogic application server will be accessed from the Internet.

- ❑ Install the WebLogic Platform with all incoming traffic blocked in some manner.
- ❑ Configure a DMZ for the WebLogic application server.
- ❑ Do not configure trust links back to the internal domain from the DMZ.
- ❑ Store only information that is meant for public dissemination on the WebLogic application server.

- ❑ For intranet service in a Windows environment, install WebLogic application server on a domain member server, not on a domain controller.
- ❑ If the domain[11] where you install WebLogic application server has trust links to other domains, ensure that the access granted to domain resources through this trust is in compliance with your enterprise's policy.
- ❑ Change account passwords every 90 days.
- ❑ Use complex passwords for accounts that will be used with certificate mapping.
- ❑ Do not use default accounts for administering a WebLogic application server.

[11] In this context, the term "domain" is being used in its general sense, not restricted to a Windows domain.

This group of recommendations is relevant when WebLogic Platform is installed on a Windows server.

- ❑ Before installing WebLogic Platform on a Microsoft Windows 2000 or 2003 server, apply the security guidance of the applicable NSA or Microsoft guide to the operating system (Windows Server 2000 or Windows Server 2003) used by your installation. [NSA-3, MS]
- ❑ If you are using Windows Server 2000, apply NSA's W2K Server.inf file (available for download, see NSA-3 reference) after modifying it appropriately to reflect the enterprise's security policy.
- ❑ Install Windows server on its own drive or partition.
- ❑ Apply the latest service packs and hotfixes.
- ❑ Disable the following services if testing shows that operational capability is not adversely affected: Alerter, Clipbook Server, Computer Browser, DHCP Client, Distributed File System, Fax Service, Internet Connection sharing, IPSEC policy agent (disable unless IPSEC policies will be used), License Logging Service, Logical Disk Manager Administrator, Messenger, NetMeeting Remote Desktop Sharing, Network DDE, Network DDE DSDM, Print Spooler, Remote Access Auto Connection Manager, Remote Access Connection Manager, Remote Registry Service (disable unless running HFNetCheck or other network management software requiring remote registry access), Removable Storage, RunAS Service, Smart Card, Smart Card Helper, Task Scheduler (disable unless batch jobs will be run from within the WebLogic application server or if scheduling tasks on the server is required), Telephony, Telnet, Windows Installer.
- ❑ Remove all unnecessary protocol stacks (do not remove TCP/IP).
- ❑ Rename the local computer's Administrator account.
- ❑ Remove the **Everyone** group from the installation drive or partition.
- ❑ Give the system account and the administrators group full control over the installation drive or partition.
- ❑ Enable Screensaver with Wait time of 15 minutes and password protection.

- ❑ If you install WebLogic Platform on a UNIX server, apply the recommendations of the NSA security guide *Guide to the Secure Configuration of Solaris 8*. [NSA-1] Although the guide is specific to Solaris, the principles are applicable to any UNIX-based operating system.
- ❑ Use Custom installation for the install and deselect each of the Workshop development tools and all sample, example, tutorial, and tour entries.

The following recommendations should be carried out when a new domain is created.

- ❑ Change the credential that is used for managing the embedded LDAP server and its data.
- ❑ Ensure that the **Anonymous Admin Lookup Enabled** checkbox is deselected.
- ❑ When establishing trust relationships with other domains, use the same credential for each domain that will be involved in the trust relationships.

- ❑ For the default realm, ensure that user lockout is enabled and modify other lockout settings based on your local policy. Do the same for any custom realms you have created.

- ❑ For each WebLogic application server in a domain, configure the SSL port numbers—turn on the **SLL Listen Port Enabled** checkboxes and enter appropriate port number in the **SSL Listen Port** text fields.
- ❑ If you need to collect information about events for the purposes of non-repudiation, create an auditor (auditing provider) and set the severity level of events for which data will be collected.
- ❑ When establishing an additional or alternate directory service, select the **SSL enabled** checkbox. If it will be the only authentication provider used by the domain, ensure that it contains the WebLogic Platform 'Administrator' account information.

The following recommendations pertain to Active Directory in a Windows environment.

- ❑ Deny Logon Locally rights to the account used to start the WebLogic application server.
- ❑ Create a global group for the WebLogic startup account. After adding the WebLogic startup account to this group, change the Primary group for the WebLogic startup account to the global group you just created.
- ❑ Remove the WebLogic startup account from the Domain Users group.
- ❑ When using the default Active Directory Authenticator, use IPSec to secure the connection between the WebLogic application server and Active Directory rather than TLS/SSL.

- ❑ Set the host field to the fully qualified domain name (FQDN) when using SSL or TLS to connect to a directory server using one of WebLogic's default directory authenticators.

This group of recommendations is relevant when the default Identity Asserter is used with the X.509 token type

- ❑ Use only one certificate authority to authenticate X.509 certificates.
- ❑ Ensure that the attribute value used to map from X.509 certificates to user accounts is unique within the certificate authority's domain of users.

Chapter

4

Post Installation

This section discusses the post installation configuration required to operate WebLogic Platform securely. Following WebLogic Platform installation, several activities must occur to secure both user and application environments. These include

- Changing passwords or deleting default accounts
- Assigning users to groups and controlling access to several privileged administrative groups
- Defining roles in accordance with an overall security approach as a mechanism for controlling access to resources
- Establishing explicit access controls for all deployed resources
- Configuring SSL

There are three subsections covering user related configuration, application related configuration, and SSL configuration.

User Related Configuration

Users

The embedded LDAP server contains a number of accounts for the default security realm. These entries remain in the embedded LDAP server even if an alternate user data store in an external LDAP server is configured.

The *weblogic* user account is the default system administrator account. The *portaladmin* user account is defined with membership in the *PortalSystemAdministrator* group. With regard to both of these accounts, define new user accounts with strong passwords and, only after the new accounts are created, delete the default accounts. There may be other accounts defined, such as *yahooadmin,* which may be associated with some reusable platform components such as portlets. Delete all other accounts from the embedded LDAP server that are not explicitly required.

Groups

The WebLogic Platform default security realm is pre-configured with a number of groups. This group information is stored in the embedded LDAP server. Of significance are the *Administrators* and *PortalSystemAdministrator* groups.

Membership in the *Administrators* group grants a user full control over a WebLogic application server via the Administration Console. Limit user membership in the *Administrators* group to personnel authorized to control and manage the execution and configuration of a WebLogic application server.

Membership in the *PortalSystemAdministrator* group enables access to WebLogic Portal management tools, which allow a user administration privileges for deployed portal applications. Limit user membership in the *PortalSystemAdministrator* group to personnel authorized to administer deployed portal applications.

Roles

Somewhat analogous to group, roles are entities that users have membership in. However, unlike groups, membership is determined dynamically based on some specified set of conditions. Roles represent privileges and are used to control access to application resources such as web pages or Enterprise JavaBeans (EJB). The association between security role and resource is referred to as the “security policy”. Role information is stored in the LDAP server.

Figure 5 illustrates how the authentication, role mapping, and authorization security service providers manage access control to domain resources through the use of roles.

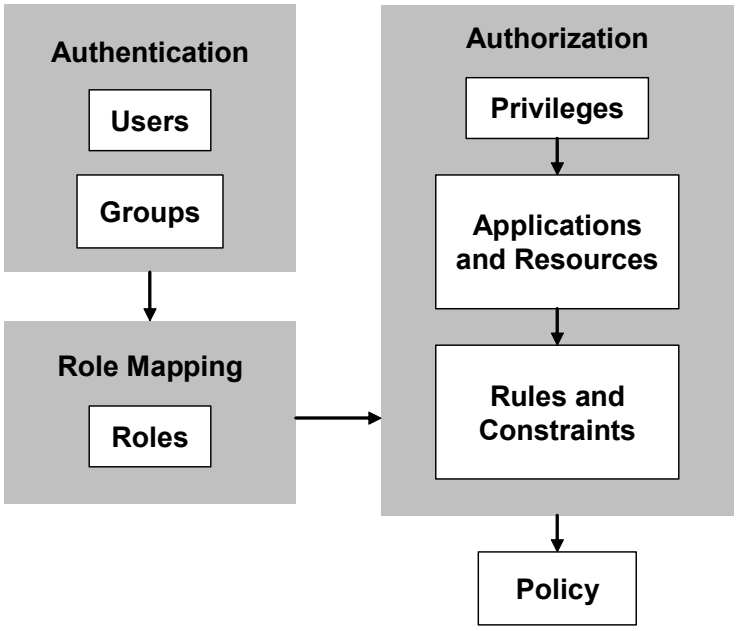

Figure 5. Role-based Policy Architecture

WebLogic Platform’s default security realm is pre-configured with a number of roles. These are referred to as “Global Roles” since they apply to all resources across the respective domain. One of these roles is the *admin* role. The rule governing whether a user is granted the *admin* role is based on whether the user is evaluated as being a member of the *Administrators* group. The *admin* role is granted full privilege to start up and configure a WebLogic application server.

As opposed to global roles, which apply to all resources across a domain, it is possible to create what are referred to as “Scoped Roles” which can be applied to specific instances of a resource (e.g. – a web page, or EJB method) within an application deployed in a domain. The BEA documentation provides instructions for using the console tool to create scoped roles at the resource level. [BEA-6]

The definition of a default set of global roles for WebLogic Platform administration and management is adequate and requires no customization.

However, a comprehensive approach to securing both platform and application resources in a domain needs to be implemented. This needs to be conceived based on the nature of the application(s) and their intended operational use. Use the Administration Console to create appropriate roles to be used in the realization of an overall access control strategy.

Application Related Configuration

Application Resource Access Control

The topic of application resource access control is very broad in scope and well covered in available BEA documentation. [BEA-4, 5, 6, 7]

No application resource is protected until a security policy is created and applied against that resource. Several options exist for establishing access control for resources in a given realm. The WebLogic application server can be set up to perform security checks for all web applications and EJBs hosted in the domain or it can be configured to establish security based on optional settings in the deployment descriptor files. The former is recommended since it forces explicit security for all resources. In the left pane of the Administration Console, select the appropriate security realm. In the right pane, pick the 'General' tab. For the attribute "Set Roles and Security Policies", choose "All Web Applications and EJBs".

> **NOTE: Setting up a WebLogic application server to perform security checks for all web applications and EJBs hosted in the domain inevitably has some impact on performance.**

Select a method for how resources will be secured using the "Future Deployments" attribute drop-down menu. Options include configuring all resource security through the Administration Console, or relying on security information in application deployment descriptor files. No recommendation is being made since the appropriate choice is based on how applications are being developed and deployed to the platform, and how overall security management will be conducted.

See the BEA documentation [BEA-6] for information regarding the generation and application of security policies. Be aware that the authorization provider stores security policies inside its database. If a security policy is to reference a particular user, group, or role, that entity must exist in the authentication or role mapping provider's database. A number of default security policies are predefined for all default global roles and default groups. BEA recommends these default security policies not be eliminated nor modified to be more restrictive, as this could have a negative impact on the proper operation of WebLogic application server.

Configuring SSL

This section discusses issues and choices to make when configuring SSL for a WebLogic application server, discussing HTTP over SSL, certificate and trust management, and one-way and two-way trusts.

HTTP over SSL (HTTPS)

The default configuration of WebLogic application server uses port 7001 for HTTP connections and 7002 for HTTP over SSL (HTTPS). The administrator can change these

ports during installation or at any time from the Administration Console. The standard ports for HTTP and HTTPS are 80 and 443, respectively. If connections from web clients to WebLogic application server will traverse a firewall or other network security boundary, the network support organization for your enterprise will need to be consulted to determine which ports to use.

For any web server to support SSL, it needs to have an SSL certificate. This certificate contains, among other things, the hostname of the server, the dates during which the certificate is valid, and the organization that issued the certificate. During installation of Weblogic application server, a test certificate containing this information is automatically created; this enables SSL but causes a security alert message to be displayed during the SSL handshake when accessed by a web browser. This does not indicate an error, it shows that the certificate is for testing only and must not mistakenly be used in a production environment.

Figure 6. Browser Security Alert Concerning Test Certificate

For production use, the server certificate that was automatically created in the installation of the Weblogic application server must be deleted and replaced with a certificate that has a new private key and is signed by a recognized Certificate Authority (CA).

The new digital certificate must be installed into WebLogic application server's identity keystore. In Weblogic application server, most of the certificate management must be done from the command line, using either the "Cert Gen utility" (http://e-docs.bea.com/wls/docs81/secmanage/ssl.html#1190032) or the Keytool utility, which was supplied as part of the Sun Microsystems J2EE package contained in WebLogic Platform.

The Cert Gen utility, like the default certificate, can be used for development or testing, but BEA recommends against its use for production purposes. The Cert Gen utility is included for administrators or developers to obtain experience creating and managing

certificates or as a no-cost test for an application's operation and performance with SSL. The Cert Gen utility is also useful to change the hostname of the test certificate in the case hostname changes in a dynamic development or lab environment. Never use the Cert Gen utility to create a production certificate; instead use the Sun Microsystems Corporation's Keytool or another third party tool made for the purpose of securely managing certificates.

Information on creating a certificate request, sending it to a CA, and installing the signed certificate into the server can be found in the *Configuring SLL* document available at http://e-docs.bea.com/wls/docs81/secmanage/ssl.html.

After this process is followed, and the signed certificate is installed on the server, the web browser should be able to directly access the Weblogic application server over HTTPS without generating the above security alert. If it still appears, it is likely that the trust relationship between the server and the client needs to be established[12]. If this is the case, the public key certificate must be obtained from the CA who signed the certificate and installed into each individual web client. This can be done manually, automatically by a script, or, in the case of Microsoft Windows, through a Global Policy Object or an enterprise management system such as Microsoft Systems Management Server (SMS).

Load the CA's public key into the keystore of each web browser client if HTTPS displays a trust warning after loading the signed certificate into the server.

Once a signed certificate is loaded into the Weblogic application server running in Production Mode, the private key passphrase must be manually typed into the operator console whenever Weblogic application server is started. Weblogic application server's Development Mode allows the server to be automatically started when the private key passphrase is stored in the "boot.config" file. Do not use Development Mode for a production server.

NOTE: Starting an SSL-enabled WebLogic application server requires manually typing the private SSL key passphrase. Systems that need to be automatically restarted should use the BEA Node Manager (http://e-docs.bea.com/wls/docs81/adminguide/nodemgr.html) for automatic restarts.

Certificate and Trust Management

Weblogic application server uses the J2EE Public Key Infrastructure (PKI) to manage the certificates that maintain the identity of the server and the chain of trust[13]. The keys associated with the identity and trust certificates are stored in two files called the Identity and Trust Keystores. The Cert Gen and Keytool utilities allow the administrator to manage these keystores. Weblogic Platform ships with multiple demonstration keystores; care must be taken when managing them to ensure that the correct keystore is active. To see which keystore is active, the administrator can consult the appropriate properties file or, to make changes, can use the KeyStores & SSL tab in the Administration Console.

Since the keystores shipped with Weblogic Server have standard private key passphrases and incorrect information in certain fields, these keystores should be replaced in production WebLogic application servers. Use the Sun Keytool utility or another third party key management tool for this purpose. Good security practice suggests several precautions. Always replace the default WebLogic keystores before using SSL on production servers. Backup the signed SSL certificates, the private key,

[12] This happens in some Department of Defense web servers that use the DoD CA, which is not in the default trust list of standard web browsers.

[13] The ordered list of signatures in a certificate is referred to as a chain of trust.

and the trust keystores and store in a secure location. Protect the private and trust keys stored in the WebLogic application server's keystores against disclosure. Limit access to production WebLogic application servers to the smallest possible number of administrators. This includes live media such as disk drives and off-line media such as backup tapes.

One-way and Two-way Trusts

The trusted link that HTTPS certificates set up between a web client and a web server is not necessarily symmetrical; SSL requires the server to have a certificate, but the client may or may not. In the usual case where only the web server uses a certificate, this is known as a "one-way" trust. This is because the client trusts the server's certificate and CA, but there is no corresponding certificate sent from the client to the server. This is useful in the case of some web commerce, where it is important to the customer that the web site is not being impersonated, but not as useful for a higher security environment where the server wants to cryptographically authenticate the client.

Two-way trust can be used when higher security than username/password authentication is needed. A two-way trust can be established if a certificate is created, signed, and loaded onto the client web browser. Organizational procedures should be followed. In general, the client generates a certificate signing request (CSR), the CSR is sent to the appropriate CA, and the returned signed certificate is installed into the web browser.

Once the signed certificate is installed, the client's browser sends the certificate to any SSL server that sends a certificate request containing the correct certificate type and CA during initiation of an SSL session. The client sends the certificate and a message digest, which is signed by the client's private key. The server uses the client's public key to verify the digest sent by the client. See, for example, the book by Rescoria (**RESCORIA**).

Use the Administration Console to enable two-way trust between the Weblogic application server and the web client, using the Administration Console **Keystores & SSL** tab. There are 3 settings

- **Client Certs Not Requested**—this is the default, meaning one-way SSL
- **Client Certs Requested But Not Enforced**—this setting requests a client to present a certificate; if a certificate is not presented, the SSL connection continues
- **Client Certs Requested And Enforced**—this setting requires a client to present a certificate; if a certificate is not presented, the SSL connection is terminated

These settings can be chosen to match the security requirements of the enterprise's policy. When the Weblogic application server obtains the client certificate, it can log the information for audit purposes and, when the third setting is in effect, it can authenticate the client.

To show how to use the client certificate for user authentication instead of a username and a password, Weblogic Server contains an example security service provider that parses a username from the email address that is contained in the client certificate. This can be seen in the Administration Console by choosing the <domainname>->Security->Realms-><realmname>->Providers->Authentication->DefaultIdentityAsserter. This provider can be used as a rudimentary certificate-based authentication method for a WebLogic application server in a development environment, eliminating the need for the user to type a username and a password to enter a system, but do not use it for a production environment.

Important Security Points

- ❑ As a minimum, modify the password for the *weblogic* user account. Alternately, delete the account, making sure that at least one other user is defined to belong to the *Administrators* group.
- ❑ As a minimum, modify the password for the *portaladmin* user account. Alternately, delete the account.
- ❑ Delete all accounts from the embedded LDAP database that are not explicitly required.
- ❑ Limit user membership in the *Administrators* group to personnel authorized to control and manage the execution and configuration of WebLogic application server.
- ❑ Limit user membership in the *PortalSystemAdministrator* group to personnel authorized to administer deployed portal applications.
- ❑ Use the Administration Console to create appropriate roles to be used in the realization of an overall access control strategy.
- ❑ Set up WebLogic application server to perform security checks for all web applications and EJBs hosted in the domain.
- ❑ Do not delete or modify the predefined, default security policies for default global roles and groups.
- ❑ If connections from web clients to WebLogic application server will traverse a firewall or other network security boundary, consult your network support organization for your enterprise to determine which HTTP and HTTPS ports should be used.
- ❑ For the production environment, delete the server certificate that was automatically created during the installation of the WebLogic application server and replace it with a certificate that has a new private key and is signed by a recognized certificate authority.
- ❑ Do not use the Cert Gen utility to create a production certificate; instead use the Sun Microsystems Corporation's Keytool or another third party tool designed to securely manage certificates.
- ❑ Load the certificate authority's public key into the keystore of each web browser client if HTTPS displays a trust warning after loading the signed certificate into the server.
- ❑ Do not use Development Mode startup for a production server. If your system needs automatic restarts, use the BEA Node Manager.
- ❑ Ensure that the correct keystore is active.
- ❑ Replace the default WebLogic keystores before using SLL on production servers.
- ❑ Back up the signed SLL certificates, the private key, and the trust keystores to a secure location.
- ❑ Protect the private and trust keys store in the WebLogic application server's keystores against disclosure.
- ❑ Limit access to production WebLogic application servers, as well as disk drives and off-line media, to the smallest possible number of administrators.

- Use two-way SSL trust for a higher security environment in which the server needs to cryptographically authenticate clients.
- Do not use the BEA-provided DefaultIdentityAsserter to authenticate users to a system from the SSL client certificate in a production environment.

Chapter

5

Summary of Important Security Points

Category	Important Security Point
General	❑ Configure the WebLogic application server's startup process to use a prompt to acquire username and password and, if possible, provide physical security protection for the server. ❑ Use the fileRealm.properties file only when managing a WebLogic application server that is based on version 6.x of the WebLogic Server product. ❑ Protect the five configuration files required by WebLogic Server and WebLogic Portal from tampering. ❑ Protect the data of the database used with WebLogic Platform from unauthorized access. ❑ Lock down external services used by WebLogic under your control and take measures to counter the threat posed by other, potentially unsecured external services. ❑ Lock down the underlying operating system used with WebLogic Platform; use the guide or benchmark appropriate for the operating system. ❑ Use a robust DBMS. NOTE: The PointBase DBMS provided with the WebLogic Platform is strictly for evaluation and tutorial purposes. Non-evaluation use of the PointBase Server requires a separate license be obtained from PointBase. ❑ Secure the DBMS that is used with WebLogic Platform in accordance with an appropriate guide or benchmark, such as the NSA guide to the secure configuration and administration of Oracle 9i Database Server. [CHRISHAY-2]
Installation	❑ Physically secure the computing resources that host WebLogic Platform. ❑ Install WebLogic Platform on its own dedicated host system, not one that supports other services such as database service. ❑ Ensure that no development tools are stored on the system that hosts WebLogic Platform.
	This group of recommendations is relevant when the WebLogic application server will be accessed from the Internet. ❑ Install the WebLogic Platform with all incoming traffic blocked in some manner.

Category	Important Security Point
	❑ Configure a DMZ for the WebLogic application server. ❑ Do not configure trust links back to the internal domain from the DMZ. ❑ Store only information that is meant for public dissemination on the WebLogic application server.
	❑ For intranet service in a Windows environment, install WebLogic application server on a domain member server, not on a domain controller. ❑ If the domain[14] where you install WebLogic application server has trust links to other domains, ensure that the access granted to domain resources through this trust is in compliance with your enterprise's policy. ❑ Change account passwords every 90 days. ❑ Use complex passwords for accounts that will be used with certificate mapping. ❑ Do not use default accounts for administering a WebLogic application server.
	This group of recommendations is relevant when WebLogic Platform is installed on a Windows server. ❑ Before installing WebLogic Platform on a Microsoft Windows 2000 or 2003 server, apply the security guidance of the applicable NSA or Microsoft guide to the operating system (Windows Server 2000 or Windows Server 2003) used by your installation. [NSA-3, MS] ❑ If you are using Windows Server 2000, apply NSA's W2K Server.inf file (available for download, see NSA-3 reference) after modifying it appropriately to reflect the enterprise's security policy. ❑ Install Windows server on its own drive or partition. ❑ Apply the latest service packs and hotfixes. ❑ Disable the following services if testing shows that operational capability is not adversely affected: Alerter, Clipbook Server, Computer Browser, DHCP Client, Distributed File System, Fax Service, Internet Connection sharing, IPSEC policy agent (disable unless IPSEC policies will be used), License Logging Service, Logical Disk Manager Administrator, Messenger, NetMeeting Remote Desktop Sharing, Network DDE, Network DDE DSDM, Print Spooler, Remote Access Auto Connection Manager, Remote Access Connection Manager, Remote Registry Service (disable unless running HFNetCheck or other network management software requiring remote registry

[14] In this context, the term "domain" is being used in its general sense, not restricted to a Windows domain.

Category	Important Security Point
	access), Removable Storage, RunAS Service, Smart Card, Smart Card Helper, Task Scheduler (disable unless batch jobs will be run from within the WebLogic application server or if scheduling tasks on the server is required), Telephony, Telnet, Windows Installer. ❑ Remove all unnecessary protocol stacks (do not remove TCP/IP). ❑ Rename the local computer's Administrator account. ❑ Remove the **Everyone** group from the installation drive or partition. ❑ Give the system account and the administrators group full control over the installation drive or partition. ❑ Enable Screensaver with Wait time of 15 minutes and password protection.
	❑ If you install WebLogic Platform on a UNIX server, apply the recommendations of the NSA security guide *Guide to the Secure Configuration of Solaris 8*. [NSA-1] Although the guide is specific to Solaris, the principles are applicable to any UNIX-based operating system. ❑ Use Custom installation for the install and deselect each of the Workshop development tools and all sample, example, tutorial, and tour entries.
	The following recommendations should be carried out when a new domain is created. ❑ Change the credential that is used for managing the embedded LDAP server and its data. ❑ Ensure that the **Anonymous Admin Lookup Enabled** checkbox is deselected. ❑ When establishing trust relationships with other domains, use the same credential for each domain that will be involved in the trust relationships.
	❑ For the default realm, ensure that user lockout is enabled and modify other lockout settings based on your local policy. Do the same for any custom realms you have created. ❑ For each WebLogic application server in a domain, configure the SSL port numbers—turn on the **SLL Listen Port Enabled** checkboxes and enter appropriate port number in the **SSL Listen Port** text fields. ❑ If you need to collect information about events for the purposes of non-repudiation, create an auditor (auditing provider) and set the severity level of events for which data will be collected. ❑ When establishing an additional or alternate directory service

Category	Important Security Point
	select the **SSL enabled** checkbox. If it will be the only authentication provider used by the domain, ensure that it contains the WebLogic Platform 'Administrator' account information.
	The following recommendations pertain to Active Directory in a Windows environment. ❑ Deny Logon Locally rights to the account used to start the WebLogic application server. ❑ Create a global group for the WebLogic startup account. After adding the WebLogic startup account to this group, change the Primary group for the WebLogic startup account to the global group you just created. ❑ Remove the WebLogic startup account from the Domain Users group. ❑ When using the default Active Directory Authenticator, use IPSec to secure the connection between the WebLogic application server and Active Directory rather than TLS/SSL.
	❑ Set the host field to the fully qualified domain name (FQDN) when using SSL or TLS to connect to a directory server using one of WebLogic's default directory authenticators.
	This group of recommendations is relevant when the default Identity Asserter is used with the X.509 token type ❑ Use only one certificate authority to authenticate X.509 certificates. ❑ Ensure that the attribute value used to map from X.509 certificates to user accounts is unique within the certificate authority's domain of users.
Post-Installation	❑ As a minimum, modify the password for the *weblogic* user account. Alternately, delete the account, making sure that at least one other user is defined to belong to the *Administrators* group. ❑ As a minimum, modify the password for the *portaladmin* user account. Alternately, delete the account. ❑ Delete all accounts from the embedded LDAP that are not explicitly required. ❑ Limit user membership in the *Administrators* group to personnel authorized to control and manage the execution and configuration of WebLogic application server. ❑ Limit user membership in the *PortalSystemAdministrator* group to personnel authorized to administer deployed portal applications. ❑ Use the Administration Console to create appropriate roles to

Category	Important Security Point
	be used in the realization of an overall access control strategy. ❑ Set up WebLogic application server to perform security checks for all web applications and EJBs hosted in the domain. ❑ Do not delete or modify the predefined, default security policies for default global roles and groups. ❑ If connections from web clients to WebLogic application server will traverse a firewall or other network security boundary, consult your network support organization for your enterprise to determine which HTTP and HTTPS ports to use. ❑ For the production environment, delete the server certificate that was automatically created during the installation of the WebLogic application server and replace it with a certificate that has a new private key and is signed by a recognized certificate authority. ❑ Do not use the Cert Gen utility to create a production certificate; instead use the Sun Microsystems Corporation's Keytool or another third party tool designed to securely manage certificates. ❑ Load the certificate authority's public key into the keystore of each web browser client if HTTPS displays a trust warning after loading the signed certificate into the server. ❑ Do not use Development Mode startup for a production server. If your system needs automatic restarts, use the BEA Node Manager. ❑ Ensure that the correct keystore is active. ❑ Replace the default WebLogic keystores before using SLL on production servers. ❑ Back up the signed SLL certificates, the private key, and the trust keystores to a secure location. ❑ Protect the private and trust keys store in the WebLogic application server's keystores against disclosure. ❑ Limit access to production WebLogic application servers, as well as disk drives and off-line media, to the smallest possible number of administrators. ❑ Use two-way SSL trust for a higher security environment in which the server needs to cryptographically authenticate clients. ❑ If the BEA-provided DefaultIdentityAsserter is used to authenticate users to a system from the SSL client certificate, remove all trusts from the WebLogic application server's trust keystore except the single domain containing the known users in the system.

Appendix

A

Bibliography

(**BEA-1**) BEA Systems, Inc., 2003, *BEA WebLogic Server: Securing a Production Environment*, Version 8.1, revised November 24, 2003, BEA Systems, Inc.

(**BEA-2**) BEA Systems, Inc., 2004, *BEA WebLogic Enterprise Security: An Introduction to BEA WebLogic Enterprise Security*, Version 4.2, revised May 2004, http://e-docs.bea.com/wles/docs42/secintro/index.html, BEA Systems, Inc.

(**BEA-3**) BEA Systems, Inc., 2004, *BEA WebLogic Enterprise Security: Managing Distributed Application Security*, BEA Systems, Inc.

(**BEA-4**) BEA Systems, Inc., July 9, 2003, *Managing WebLogic Security*, Release 8.1, BEA Systems, Inc.

(**BEA-5**) BEA Systems, Inc., July 2003, *Introducing BEA WebLogic Platform 8.1 Security*, Version 8.1, BEA Systems, Inc.

(**BEA-6**) BEA Systems, Inc., 2004, *Securing WebLogic Resources*, http://e-docs.bea.com/wls/docs81/secwlres/index.html, BEA Systems, Inc.

(**BEA-7**) BEA Systems, Inc., 2004, *Types of WebLogic Resources*, http://e-docs.bea.com/wls/docs81/secwlres/types.html, BEA Systems, Inc.

(**CHRISHAY-1**) Christman, S. M. and J. Hayes, Maj USAF, August 26, 2003, *Guide to the Secure Configuration and Administration of Microsoft SQL Server 2000*, Report C4-50R-02, Version 1.5, Network Applications Team of the Systems and Network Attack Center, National Security Agency, Ft. Meade, Maryland.

(**CHRISHAY-2**) Christman, S. M. and J. Hayes, Maj USAF, September 30, 2003, *Guide to the Secure Configuration and Administration of Oracle9i® Database Server*, Report C4-10R-03, Version 1.2, Network Applications Team of the Systems and Network Attack Center, National Security Agency, Ft. Meade, Maryland.

(**DOD**) Department of Defense, USA, October 2002, *Information Assurance*, DoD Directive Number 8500.1, 24, certified current as of 21 November 2003, U.S. Department of Defense, Washington, DC.

(**RESCORIA**) Rescoria, Eric, 2001, *SSL and TLS: Designing and Building Secure Systems*, Addison-Wesley.

(**MS**) Microsoft Solutions for Security Group, April 24, 2003, Windows Server 2003 Security Guide, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399.

(**NSA-1**) Operating Systems Division UNIX Team of the Systems and Network Attack Center, October 9, 2003, *Guide to the Secure Configuration of Solaris 8*, C4I Technical Report C4I-0008R-2003, Version 1.0, National Security Agency, Ft. Meade, Maryland.

(**NSA-2**) Network Attack Techniques Division of the Systems and Network Attack Center (SNAC), August 13, 2001, *Microsoft Windows 2000 IPsec Guide*, Technical Report C4-045R-01, National Security Agency, 9800 Savage Rd. Suite 6704, Ft. Meade, MD 20755-6704.

(**NSA-3**) NSA Guide for Securing Microsoft Server 2000, see description and instructions at the NSA site, http://www.nsa.gov/snac/downloads_win2003.cfm?MenuID=scg10.3.1.1.

Appendix

B

Glossary

This appendix provides expansions for acronyms and definitions for technical terms used in this guide.

Acronyms

3DES	**Triple Data Encryption Standard**
BEA	**BEA Systems, Inc.**
CA	**Certificate Authority**
CSR	**Certificate Signing Request**
DBMS	**Database Management System**
DDE	**Dynamic Data Exchange**
DHCP	**Dynamic Host Configuration Protocol**
DSDM	**DDE Share Database Manager**
EIS	**Enterprise Information Systems**
EJB	**Enterprise JavaBeans**
ERP	**Enterprise Resource Planning**
FQDN	**Fully Qualified Domain Name**
HTTP	**HyperText Transfer Protocol**
HTTP-S	**HyperText Transfer Protocol - Secure**
IA	**Information Assurance**
IDE	**Integrated Development Environment**
IP	**Internet Protocol**
IPSEC	**Secure Internet Protocol**
ISP	**Important Security Point**
J2EE	**Java 2 Enterprise Edition**
JAAS	**Java Authentication and Authorization Service**
JCE	**Java Cryptography Extensions**
JDBC	**Java Database Connectivity**
JDK	**Java Development Kit**
JMS	**Java Messaging Service**
JNDI	**Java Naming and Directory Interface**
JSSE	**Java Secure Sockets Extensions**

JTA	**Java Transaction API**
JVM	**Java Virtual Machine**
LDAP	**Lightweight Directory Access Protocol**
NSAPI	**Netscape Server Application Programming Interface**
PKI	**Public Key Infrastructure**
RMI	**Remote Method Invocation**
SAML	**Security Assertion Markup Language**
SMS	**Systems Management Server**
SNAC	**Systems and Network Attack Center**
SOAP	**Simple Object Access Protocol**
SSL	**Secure Sockets Layer**
SSP	**Security Service Provider**
TCP/IP	**Transmission Control Protocol / Internet Protocol**
TLS	**Transaction Layer Security**
URI	**Uniform Resource Identifier**
URL	**Uniform Resource Locator**
W2K	**Windows 2000 (operating system)**
WS	**Web Services**
WSDL	**Web Service Description Language**
XML	**eXtensible Markup Language**

Definitions

Adjudication If multiple authorization providers are configured within a realm, an adjudication provider is required to resolve potential conflicts between them.

Administration Console The administrator's console in WebLogic Server, used to configure and control WebLogic application servers.

Application Server A J2EE-based product that resides in the middle-tier of a server centric architecture. It provides an EJB server and middleware services for security and state maintenance along with data access and persistence. Application servers evolved from the need to support applications that share data and resources with other systems, and generate dynamic information for Web pages and other user interfaces. They introduced into server-side architecture, a new layer of functions and services between Web servers and underlying applications and databases.

Auditing The means by which system events are recorded.

Authentication In security systems, the process of identifying an individual, usually based on a username and password. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

Authorization (Distinct from *Authentication*). The process of giving individuals access to system objects based on their identity.

Certificate An attachment to an electronic message used for security purposes in the SSL protocol. The most common use of a digital certificate is to verify that a user sending a message is who he or she claims to be, and to provide the receiver with the means to encode a reply.

Certificate Authority A certificate authority is a trusted third-party signer of security certificates. A server employing two-way SSL uses a certificate signed by a certificate authority to authenticate itself to a browser and then the browser uses its certificates to authenticate itself to the server.

Communications Service Request (CSR) A certificate request sent to a CA.

Credential An object that is verified when presented to the verifier in an authentication transaction.

Credential mapping Allows WebLogic application server to log into remote systems on behalf of a user or computer that has already been authenticated. The credential-mapping provider provides a means to map the authenticated credentials to the set of credentials required by the remote system.

Enclave A system that requires an independent security classification from other systems. Example enclaves include DMZ(s), server networks, host networks, and extranet systems.

Firewall Used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria

Group Set of users with similar privileges.

HyperText Transfer Protocol (HTTP) The underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and what actions Web servers and browsers should take in response to various commands.

HTTPS HTTP over SSL also referred to as Secure HTTP.

J2EE A J2EE Platform, consisting of a Web server and an EJB server, is the Middle Tier of a multi-tier distributed model that includes a Client (browser) Tier, a Middle Tier, and an Enterprise Information System (database) Tier.

JRockit Part of Weblogic Platform; BEA's optimized implementation of the Java Virtual Machine (JVM) for Microsoft Windows.

Keystore A mechanism designed to store password-protected private keys. The keystore is the file that actually holds the set of keys.

Keystore Provider In the WebLogic Server security architecture, a Keystore provider is used to access keystores. WebLogic Server only supports the WebLogic Keystore provider.

Keytool Sun Microsystems Corporation's third party tool made for the purpose of securely managing certificates.

LDAP A set of protocols for accessing information directories (databases). LDAP is based on the standards contained within the X.500 standard, but is significantly simpler.

Lockdown Securing a computer system both physically and via software mechanisms.

Non-repudiation Services that provide non-forgeable evidence that a specific action occurred.

One-way Trust The usual case in SSL where only the web server provides a certificate for authentication. Used where it is important to the client (customer) that the server (web site) is not being impersonated.

Public Key Encryption A cryptographic system that uses two keys -- a *public key* known to everyone and a *private* or *secret key* known only to the recipient of the message. When John wants to send a secure message to Jane, he uses Jane's public key to encrypt the message. Jane then uses her private key to decrypt it.

Public Key Infrastructure (PKI) A system of digital certificates, Certificate Authorities, and other registration authorities that verify and authenticate the validity of each party involved in an Internet transaction.

Role Somewhat analogous to group, roles are entities which users have membership in. However, unlike groups, membership is determined dynamically based on some specified set of conditions.

Role Mapping Determining the set of roles that are valid for a user with respect to the targeted resource. A role-mapping provider dynamically obtains a set of security roles granted to a requestor for a given WebLogic resource.

Secure Sockets Layer (SSL) A protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a private key to encrypt data that's transferred over the SSL connection.

Security Service Providers (SSPs) Security service providers are modules that interface with a WebLogic security realm to provide security services to applications; they call into the WebLogic security framework on behalf of applications. If the WebLogic security service providers supplied with the WebLogic Server product do not fully meet security requirements, they can be supplemented or replaced with custom security service providers.

Signed Certificate A digital certificate that is signed (verified) by a trusted Certificate Authority.

Simple Object Access Protocol (SOAP) A standard format for applications to call each other’s methods and pass data to one another

Two-way Trusts Where both the client and the server provide certificates for authentication.

Uniform Resource Identifier (URI) The generic term for all types of names and addresses that refer to objects on the World Wide Web. A URL is one kind of URI.

Uniform Resource Locator (URL) The global address of documents and other resources on the World Wide Web. The first part of the address indicates what protocol to use, and the second part specifies the IP address or the domain name where the resource is located.

Web Browser A software application used to locate and display Web pages. The two most popular browsers are Netscape Navigator and Microsoft Internet Explorer.

Web Server Software that allows a computer to deliver (informally, “serve up”) web pages. There are many web server software packages, including public domain software from NCSA and Apache, and commercial packages from Microsoft and Netscape.

Web Service An application distributed on the web that cooperatively provides a service through the use of standard technologies. The service is accessible from a Java program through a URI.

WebLogic application server The application server (see definition above) created by configuring WebLogic Server software for operational use; it is an implementation of the J2EE platform.

WebLogic Integration Part of Weblogic Platform; the component that provides support for application integration.

WebLogic Platform An integrated suite of extensible components for developing and hosting web-based e-commerce oriented services.

WebLogic Portal Part of Weblogic Platform; the framework for development and deployment of portal-oriented web applications.

WebLogic Server The central component of WebLogic Platform; it provides an application infrastructure of reusable J2EE services that can be configured to host J2EE-compliant applications.

WebLogic Workshop Part of Weblogic Platform; an integrated development environment (IDE) and runtime framework.

Web Service Description Language (WSDL) A standard format for describing web services.

EXtensible Markup Language (XML) A common human readable language by which different applications may communicate with one another over a network.

Appendix

C

Standards

Compliance with standards facilitates lower cost of ownership, greater interoperability, and less chance of implementation error. WebLogic Enterprise Security adheres to standards specified in Table 1.

Table 1. BEA WebLogic Enterprise Security Standards

Technology	Standard	WebLogic Use
XML	Security Assertion Markup Language (SAML)	Participate in SAML-based SSO environment.
XML	Simple Object Access Protocol (SOAP) 1.1	Protocol over which BEA WebLogic Enterprise Security communicates with its various entities, including communication over which policy is delivered to various components.
Java	CertPath	Retrieve X.509 digital certificates associated with infrastructure protection; available for customer direct use.
Java	KeyStore	Retrieve RSA private keys associated with X.509 digital certificates associated with infrastructure protection; available for customer direct use.
Java	Java Secure Sockets Extensions (JSSE)	Protect infrastructure network connections for establishment of mutual trust.
Java	Java Cryptography Extensions (JCE)	Integrate cryptographic libraries.
Java	Java Authentication and Authorization Service (JAAS)	Provide authentication service implementations.
Miscellaneous	X.509	Validate the identity of infrastructure components through digital certificates; supported as proof of identity for customer use.
Miscellaneous	Lightweight Directory Access Protocol (LDAP) v3	Retrieve configuration information from the Service Control Manager, user identity and user attributes from an LDAP v3 directory server.
Miscellaneous	Netscape Server Application Programming Interface (NSAPI)	Support compliant runtime for authentication, SAML single sign-on, and protection of hosted web pages.

Technology	Standard	WebLogic Use
Miscellaneous	FIPS 140	Support certification of the embedded cryptographic libraries used for cryptographic protection and TLS protocol.
Miscellaneous	TLS v1 and SSL	Protect network communication between infrastructure components.
Miscellaneous	JDBC	Provide access to database stores using the database provider.

Appendix

D

Testing Security Control of Users

Testing was performed to verify proper management of user accounts, as follows:

- Sensitive authentication information is protected successfully. When using SSL for Weblogic to communicate with the Netscape Directory Server LDAP database, the data streams for both database binding and administrator logon were monitored. The credentials were encrypted as advertised, that is, the passwords were encoded into human unrecognizable form.
- The Administration Console has proper authentication control. When trying to logon without entering a username and password, access was denied.
- A sample application portal has proper authentication control. When trying to logon to the portal without entering a username and password, access was denied.
- The Administration Console has proper password authentication control. When trying to logon with a valid username and invalid password, access was denied.
- A sample application portal has proper password authentication control. When trying to logon with a valid username and invalid password, access was denied.
- The Administration Console has proper piggyback password authentication control. After a proper logon and logout, an attempt to logon with the same username but without a password failed, as expected.
- A sample application portal has proper piggyback password authentication control. After a proper logon and logout, an attempt to logon with the same username but without a password failed, as expected.
- Users of a sample application portal can be properly added using the LDAP console. A new user and password were created using the Netscape Directory Server LDAP console; this user could successfully access the sample portal application.
- Users of a sample application portal can be properly deleted using the LDAP console. A user and password were deleted using the Netscape Directory Server LDAP console; this user could then no longer access the sample portal application.
- Similarly, WebLogic application server Groups can be successfully added and deleted using the LDAP console. Note: Don't forget to create the chapter sidebars (use Chapter Side Bar and Even Page Side Bar buttons).

This Page Intentionally Left Blank