Oracle® Fusion Middleware

Securing a Production Environment for Oracle WebLogic Server

12*c* (12.1.3)
E41900-09
August 2018

Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server, 12*c* (12.1.3)

E41900-09

Contents

Preface

This preface describes the document accessibility features and conventions used in this guide—*Securing a Production Environment for Oracle WebLogic Server*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction and Roadmap

This chapter describes the contents and organization of this guide - *Securing a Production Environment for Oracle WebLogic Server*.
This chapter includes the following sections:

1.1 Document Scope and Audience

This guide is intended for the following audiences:

- Application Architects — Architects who, in addition to setting security goals and designing the overall security architecture for their organizations, evaluate WebLogic Server security features and determine how to best implement them. Application Architects have in-depth knowledge of Java programming, Java security, and network security, as well as knowledge of security systems and leading-edge, security technologies and tools.
- Security Developers — Developers who focus on defining the system architecture and infrastructure for security products that integrate into WebLogic Server and on developing custom security providers for use with WebLogic Server. Security Developers have a solid understanding of security concepts, including authentication, authorization, auditing (AAA), in-depth knowledge of Java (including Java Management eXtensions (JMX), and working knowledge of WebLogic Server and security provider functionality.
- Application Developers — Java programmers who develop and add security to Web applications and Enterprise JavaBeans (EJBs), and work with other engineering, quality assurance (QA), and database teams to implement security features. Application Developers have in-depth/working knowledge of Java (including Java Platform, Enterprise Edition (Java EE) Version 5 components such as servlets/JSPs and JSEE) and Java security.
- Server Administrators — Administrators who work closely with Application Architects to design a security scheme for the server and the applications running on the server, to identify potential security risks, and to propose configurations that prevent security problems. Related responsibilities may include maintaining critical production systems, configuring and managing security realms, implementing authentication and authorization schemes for server and application resources, upgrading security features, and maintaining security provider databases. Server Administrators have in-depth knowledge of the Java security architecture, including Web services, Web application and EJB security, Public Key security, SSL, and Security Assertion Markup Language (SAML).

- Application Administrators — Administrators who work with Server Administrators to implement and maintain security configurations and authentication and authorization schemes, and to set up and maintain access to deployed application resources in defined security realms. Application Administrators have general knowledge of security concepts and the Java Security architecture. They understand Java, XML, deployment descriptors, and can identify security events in server and audit logs.

1.2 Guide to This Document

This document is organized as follows:

- This chapter, Introduction and Roadmap, introduces the organization of this guide.
- Determining Your Security Needs, explains how to determine the security needs for your particular environment and describes basic measures to ensure that those needs are being met.
- Ensuring the Security of Your Production Environment, highlights essential security measures to consider before you deploy WebLogic Server into a production environment and describes how to use different security settings to secure a production environment.

1.3 Related Information

The following Oracle WebLogic Server documents contain information that is relevant to the WebLogic Security Service:

- *Administering Security for Oracle WebLogic Server* — explains how to configure security for WebLogic Server and how to use Compatibility security.
- *Developing Security Providers for Oracle WebLogic Server* — explains how vendors and application developers can develop custom security providers that can be used with WebLogic Server.
- *Understanding Security for Oracle WebLogic Server* — provides an overview of the features, architecture, and functionality of the WebLogic Security Service. It is the starting point for understanding the WebLogic Security Service.
- *Securing Resources Using Roles and Policies for Oracle WebLogic Server* — describes how to secure WebLogic resources. It primarily focuses on securing URL (Web) and Enterprise JavaBean (EJB) resources.
- *Java API Reference for Oracle WebLogic Server* — is reference documentation for the WebLogic security packages that are provided with and supported by this release of WebLogic Server.

1.4 Security Samples and Tutorials

Oracle provides code samples for Java Authentication and Authorization Service and for Outbound and Two-way SSL for Security developers. The examples and tutorials illustrate WebLogic Server Security in action, and provide practical instructions on how to perform key Security development tasks.

Oracle recommends that you run some or all of the Security examples before developing your own Security configurations.

1.4.1 Avitek Medical Records Application (MedRec) and Tutorials

MedRec is an end-to-end sample Java EE application shipped with WebLogic Server that simulates an independent, centralized medical record management system. The MedRec application provides a framework for patients, doctors, and administrators to manage patient data using a variety of different clients.

MedRec demonstrates WebLogic Server and Java EE features, and highlights Oracle-recommended best practices. MedRec is optionally installed with the WebLogic Server installation. You can start MedRec from the *ORACLE_HOME*\user_projects\domains\medrec directory, where *ORACLE_HOME* is the directory you specified as the Oracle Home when you installed Oracle WebLogic Server. See Sample Applications and Code Examples in *Understanding Oracle WebLogic Server*.

MedRec includes a service tier consisting primarily of Enterprise Java Beans (EJBs) that work together to process requests from Web applications, Web services, workflow applications, and future client applications. The application includes message-driven, stateless session, stateful session, and entity EJBs.

1.4.2 Security Examples in the WebLogic Server Distribution

WebLogic Server optionally installs API code examples in *EXAMPLES_HOME*\wl_server\examples\src\examples, where *EXAMPLES_HOME* represents the directory in which the WebLogic Server code examples are configured. See Sample Applications and Code Examples in *Understanding Oracle WebLogic Server*.

1.4.3 Additional Security Examples Available for Download

Additional API examples for download at http://www.oracle.com/technetwork/indexes/samplecode/index.html. These examples are distributed as ZIP files that you can unzip into an existing WebLogic Server samples directory structure.

You build and run the downloadable examples in the same manner as you would an installed WebLogic Server example. See the download pages of individual examples at http://www.oracle.com/technetwork/indexes/samplecode/index.html.

1.5 New and Changed Features in This Release

For a comprehensive listing of the new WebLogic Server features introduced in this release, see *What's New in Oracle WebLogic Server*.

2

Determining Your Security Needs

The security requirements you establish for your WebLogic Server environment are based upon multiple considerations, such as the types of resources hosted on WebLogic Server that need to be protected, the users and other entities that access those resources, recommendations from Oracle as well as in-house or independent security consultants, and more.
This chapter describes how you can determine your security needs and make sure that you take the appropriate security measures before you deploy WebLogic Server 12.1.3 and your Java EE applications into a production environment. This chapter includes the following sections:

2.1 Understand Your Environment

The WebLogic Server environment includes not only the resources that are hosted on WebLogic Server, but also the software systems and other entities with which those WebLogic Server resources interoperate, such as databases, and load balancers, and the users who have access to that environment.
To better understand your security needs, ask yourself the following questions:

- Which resources am I protecting?

  Many resources in the production environment can be protected, including information in databases accessed by WebLogic Server and the availability, performance, applications, and the integrity of the Web site. Consider the resources you want to protect when deciding the level of security you must provide.

- From whom am I protecting the resources?

  For most Web sites, resources must be protected from everyone on the Internet. But should the Web site be protected from the employees on the intranet in your enterprise? Should your employees have access to all resources within the WebLogic Server environment? Should the system administrators have access to all WebLogic resources? Should the system administrators be able to access all data? You might consider giving access to highly confidential data or strategic resources to only a few well trusted system administrators. Perhaps it would be best to allow no system administrators access to the data or resources.

- What will happen if the protections on strategic resources fail?

  In some cases, a fault in your security scheme is easily detected and considered nothing more than an inconvenience. In other cases, a fault might cause great damage to companies or individual clients that use the Web site. Understanding the security ramifications of each resource will help you protect it properly.

2.2 Hire Security Consultants or Use Diagnostic Software

Whether you deploy WebLogic Server on the Internet or on an intranet, it is a good idea to hire an independent security expert to go over your security plan and procedures, audit your installed systems, and recommend improvements. Oracle Consulting offers services and products that can help you to secure a WebLogic Server production environment. See the Oracle Consulting page at http://www.oracle.com/in/products/ondemand/index.html.

2.3 Read Security Publications

Read about security issues:

- For the latest information about securing Web servers, Oracle recommends the "Security Practices & Evaluations" information available from the CERT Coordination Center operated by Carnegie Mellon University at `http://www.cert.org/`.
- Register your WebLogic Server installation with My Oracle Support. By registering, Oracle Support will notify you immediately of any security updates that are specific to your installation. You can create a My Oracle Support account by visiting `http://www.oracle.com/support/index.html`.
- For security advisories, refer to the Critical Patch Updates and Security Alerts page at the following location:

  `http://www.oracle.com/technetwork/topics/security/alerts-086861.html`

  If you have an installation of WebLogic Server 10*g* Release 3 (10.3) or earlier, visit the BEA Security Advisories Archive Page at `http://www.oracle.com/technetwork/topics/security/beaarchive-159946.html`.

2.4 Install WebLogic Server in a Secure Manner

Currently, the WebLogic Server installation includes some additional WebLogic Server development utilities (for example, wlsvc). These development programs could be a security vulnerability. The following are recommendations for making a WebLogic Server installation more secure:

- Do not install the WebLogic Server sample applications. When installing WebLogic Server, make sure that the option to install the Server Examples component is not selected.
- Do not run WebLogic Server when configured in development mode. Rather, you should make sure WebLogic Server is configured to run in production mode. Production mode sets the server to run with settings that are more secure and appropriate for a production environment.

> **Note:**
>
> When WebLogic Server is configured in development mode, certain error conditions, such as a misbehaving application or an invalid configuration of WebLogic Server, may result in a trace stack being displayed. While error responses generally are not dangerous, they have the potential to give attackers information about the application or the WebLogic Server installation that can be used for malicious purposes.

3 Ensuring the Security of Your Production Environment

A comprehensive lockdown of the WebLogic Server production environment includes securing the host machine and limiting access only to authorized users. Lockdown also includes securing network resources by creating firewalls, using a domain-wide secure port for Administration Server communications, and securing the WebLogic Security Service.
This chapter includes the following sections:

3.1 An Important Note Regarding Null Cipher Use in SSL

SSL clients start the SSL handshake by connecting to the server. As part of the connection, the client sends the server a list of the cipher suites it supports.

A cipher suite is an SSL encryption method that includes the key exchange algorithm, the symmetric encryption algorithm, and the secure hash algorithm. A cipher suite is used to protect the integrity of a communication. For example, the cipher suite called RSA_WITH_RC4_128_MD5 uses RSA for key exchange, RC4 with a 128-bit key for bulk encryption, and MD5 for message digest.

The server selects a mutually-supported cipher suite from the list supplied by the client for the client and server to use for this session.

However, an incorrectly configured client might specify a set of cipher suites that contain only null ciphers. A null cipher passes data on the wire in clear-text. (An example of a cipher suite with a null cipher is TLS_RSA_WITH_NULL_MD5.) Using a null cipher makes it possible to see the SSL messages by using a network packet sniffer. In essence, SSL is used but does not provide any security.

The server selects the null cipher only when it is the only cipher suite they have in common. If the server selects a null cipher from the client's cipher suite list, the log contains the following message: SSL has established a session that uses a Null cipher.

However, an incorrectly configured client might specify a set of cipher suites that contain only null ciphers.

A null cipher passes data on the wire in clear-text. (An example of a cipher suite with a null cipher is TLS_RSA_WITH_NULL_MD5.) Using a null cipher makes it possible to

see the SSL messages by using a network packet sniffer. In essence, SSL is used but does not provide any security.

The server selects the null cipher only when it is the only cipher suite they have in common.

If the server selects a null cipher from the client's cipher suite list, the log contains the following message: “SSL has established a session that uses a Null cipher.”

This message is output only when the server has selected a null cipher from the client's list.

> **Note:**
>
> If there is any potential whatsoever that an SSL client might use a null cipher to inappropriately connect to the server, you should check the log file for this message. Oracle recommends that new client configurations be given extra attention with respect to the use of a null cipher to ensure that they are properly configured.
>
> It is unlikely that an existing client configuration would suddenly start using null ciphers if it had not been doing so previously. However, an existing client configuration that is unknowingly configured incorrectly could be using null ciphers.

Other SSL errors unrelated to null ciphers are possible as well, and each will display an appropriate error message in the log.

For information on configuring SSL, see Configuring SSL in *Administering Security for Oracle WebLogic Server*. For information on viewing log files, see View and configure logs in the *Oracle WebLogic Server Administration Console Online Help*.

3.1.1 New Control to Prevent Null Cipher Use

As of release 10*g* Release 3 (10.3), WebLogic Server includes a WebLogic Server Administration Console control to prevent the server from using a null cipher.

The **Allow Unencrypted Null Cipher** control, which is available in the WebLogic Server Administration Console by selecting **Servers** > *ServerName* > **Configuration** > **SSL** > **Advanced**, determines whether null ciphers are allowed. By default, this control is not set and the use of a null cipher is not allowed on the server. In such a configuration, if the SSL clients want to use the null cipher suite (by indicating TLS_RSA_WITH_NULL_MD5 as the only supported cipher suite), the SSL handshake will fail.

If you set this control, the null cipher suite (for example, TLS_RSA_WITH_NULL_MD5) is added to the list of supported cipher suites by the server. The SSL connection has a chance to use the null cipher suite if the client wants to do so. If the null cipher suite is used, the message will be unencrypted.

> **Caution:**
>
> Do not set this control in a production environment unless you are aware of the implications and consequences of doing so.

This control is also exposed as a system runtime parameter, `weblogic.security.SSL.allowUnencryptedNullCipher`, and as an `AllowUnencryptedNullCipher` attribute on the SSLMBean.

3.2 Securing the WebLogic Server Host

A WebLogic Server production environment is only as secure as the security of the machine on which it is running. It is important that you secure the physical machine, the operating system, and all other software that is installed on the host machine.

> **Note:**
>
> FIPS mode is supported for JSSE via the RSA provider.
>
> To enable FIPS in JSSE, follow these steps:
>
> 1. Put the `cryptojFIPS.jar` jar (*`WL_HOME`*`/server/lib/cryptojFIPS.jar`) in the classpath, ahead of `cryptoj.jar`. For example, you could add `cryptojFIPS.jar` to the `PRE_CLASSPATH` variable in the server start script, typically `startWebLogic.cmd/sh`).
> 2. Enable the RSA JSSE provider, as described in Using the RSA JSSE Provider in WebLogic Server in *Administering Security for Oracle WebLogic Server*.
>
> FIPS 140-2 is a standard that describes U.S. Federal government requirements for sensitive, but unclassified use.

The following are recommendations for securing a WebLogic Server host in a production environment. Also check with the manufacturer of the machine and operating system for recommended security measures.

> **Important:**
>
> WebLogic domain and server configuration files should be accessible only by the operating system users who configure or execute WebLogic Server.

Table 3-1 Securing the WebLogic Server Host

Security Action	Description
Physically secure the hardware.	Keep your hardware in a secured area to prevent unauthorized operating system users from tampering with the deployment machine or its network connections.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Log out of the WebLogic Server Administration Console before navigating to a non-secure site.	If you are logged on to the WebLogic Server Administration Console, be sure to log out completely before browsing to an unknown or non-secure Web site.
Secure networking services that the operating system provides.	Have an expert review network services such as e-mail programs or directory services to ensure that a malicious attacker cannot access the operating system or system-level commands. The way you do this depends on the operating system you use. Sharing a file system with other machines in the enterprise network imposes risks of a remote attack on the file system. Be certain that the remote machines and the network are secure before sharing the file systems from the machine that hosts WebLogic Server.
Use a file system that can prevent unauthorized access.	Make sure that the file system on each WebLogic Server host can prevent unauthorized access to protected resources. For example, on a Windows computer, use only NTFS.
Set file access permissions for data stored on disk.	Set operating system file access permissions to restrict access to data stored on disk. This data includes, but is not limited to, the following: • The security LDAP database, which by default is in `\domains\domain-name \servers\server-name\data\ldap\ldapfiles`. • The directory and filename location of a private keystore, as described in Storing Private Keys, Digital Certificates, and Trusted Certificate Authorities in *Administering Security for Oracle WebLogic Server*. • The directory and filename location of a Root Certificate Authority (CA) keystore, as described in Storing Private Keys, Digital Certificates, and Trusted Certificate Authorities in *Administering Security for Oracle WebLogic Server*. For example, operating systems such as Unix and Linux provide utilities such as umask and chmod to set the file access permissions. At a minimum, consider using "umask 066", which denies read and write permission to Group and Others.
Set file access permissions for data stored in persistent store.	Set operating system file access permissions to restrict access to data stored in the persistent store. When using a synchronous write policy of `Direct-Write-With-Cache`, limit access to the cache directory, especially if there are custom configured user access limitations on the primary directory. See Overview of the Persistent Store in "Using the WebLogic Persistent Store" in *Administering Server Environments for Oracle WebLogic Server* for the WebLogic services and subsystems that can create connections to the persistent store. The default persistent store maintains its data in a `data\store\default` directory inside the servername subdirectory of a domain's root directory.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Limit the number of user accounts on the host machine.	Avoid creating more user accounts than you need on WebLogic Server host machines, and limit the file access privileges granted to each account. On operating systems that allow more than one system administrator user, the host machine should have two user accounts with system administrator privileges and one user with sufficient privileges to run WebLogic Server. Having two system administrator users provides a back up at all times. The WebLogic Server user should be a restricted user, not a system administrator user. One of the system administrator users can always create a new WebLogic Server user if needed. **Important:** WebLogic domain and server configuration files should be accessible only by the operating system users who configure or execute WebLogic Server. (See the security action provided later in this table that advises you to give only one user account access to WebLogic resources, in addition to the two system administrator users who also have access privileges.) Review active user accounts regularly and when personnel leave. *Background Information:* Some WebLogic Server configuration data and some URL (Web) resources, including Java Server Pages (JSPs) and HTML pages, are stored in clear text on the file system. A sophisticated user or intruder with read access to files and directories might be able to defeat any security mechanisms you establish with WebLogic Server authentication and authorization schemes.
For your system administrator user accounts, choose names that are not obvious.	For additional security, avoid choosing an obvious name such as "system", "admin", or "administrator" for your system administrator user accounts.
Safeguard passwords.	The passwords for user accounts on production machines should be difficult to guess and should be guarded carefully. Set a policy to expire passwords periodically. Never code passwords in client applications.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Do not include unencrypted passwords in command lines.	Several WebLogic Server commands, including WLST and `weblogic.Deployer` commands in scripts, permit you to specify unencrypted passwords in the command line. But specifying unencrypted passwords in the command line is a security risk: they can be easily viewed from the monitor screen by others, and they are displayed in process listings that log the execution of those commands. When entering commands that require an unencrypted password, whether in a command window or script, take the following precautions to ensure that the passwords are entered securely: • Enter passwords only when prompted. If you omit the password from the command line, you are subsequently prompted for it when the command is executed. The characters you type are not echoed. • For scripts that start WebLogic Server instances, create a boot identity file. The boot identity file is a text file that contains user credentials for starting and stopping an instance of WebLogic Server. An Administration Server can refer to this file for user credentials instead of prompting you to provide them when the script is run. Because the credentials are encrypted, using a boot identity file is much more secure than storing unencrypted credentials in a startup or shutdown script. In script-based Node Manager commands that start remote Administration Server instances, ensure that the remote start username and password are obtained from the Administration Server's boot identity file. • For WLST scripts that contain commands requiring a user name and password, create a user configuration file. This file, which you can create via the WLST `storeUserConfig` command, contains: – Your credentials in an encrypted form – A key file that WebLogic Server uses to unencrypt the credentials During WLST sessions, or in WLST scripts, the user configuration file can be passed in commands such as the following: – `connect` — for connecting to a running WebLogic Server instance – `startServer` — for starting the Administration Server – `nmConnect` — for connecting WLST to Node Manager to establish a session • For `weblogic.Deployer` scripts containing commands requiring a user name and password, you can specify the user configuration file created via the WLST `storeUserConfig` command instead of entering your unencrypted credentials. For more information about passing user credentials securely in scripts, see the following topics: • Starting and Stopping Servers and Boot Identity Files in *Administering Server Startup and Shutdown for Oracle WebLogic Server* • Security for WLST in *Understanding the WebLogic Scripting Tool* • Configuring Remote Server Start Security for Script-based Node Manager in *Administering Node Manager for Oracle WebLogic Server* • Syntax for Invoking weblogic.Deployer in *Deploying Applications to Oracle WebLogic Server*

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
On each host computer, give only one user account access to WebLogic resources (in addition to the two system administrator users who also have access privileges).	**Important:** WebLogic domain and server configuration files should be accessible only by the operating system users who configure or execute WebLogic Server. No other operating system user (apart from the system administrators) should have read, write, or execute access to WebLogic Server product files nor to your domain files. On each WebLogic Server host computer, use the operating system to establish a special user account (for example, `wls_owner`) specifically to run WebLogic Server. Grant to this operating-system (OS) user account access privileges only to the following directories: • **Oracle home** The top-level directory that is created for all the Oracle Fusion Middleware products that are installed on your machine; this directory is created when WebLogic Server is installed. • **WebLogic Server product installation directory** This directory contains all the WebLogic Server software components that you choose to install on your system, including program files. By default, this directory is a subdirectory of the Oracle home and is named `wlserver`. For more information, refer to Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*. • **WebLogic domain directories** These contain the configuration files, security files, log files, Java EE applications, and other Java EE resources for a single WebLogic domain. By default, a domain is a subdirectory of Oracle home (for example, `Oracle/Middleware/user_projects/domains/`*`domain1`*); however, domain directories can be located outside the WebLogic Server installation directory and Oracle home as well. If you create multiple domains on a WebLogic Server host computer, each domain directory must be protected. This protection limits the ability of other applications that are executing on the same machine as WebLogic Server to gain access to WebLogic Server files and your domain files. Without this protection, some other application could gain write access and insert malicious, executable code in JSPs and other files that provide dynamic content. The code would be executed the next time the file was served to a client. Knowledgeable operating system users may be able to bypass WebLogic Server security if they are given write access, and in some cases read access, to the following files: • WebLogic Server Installation • JDK files (typically in the WebLogic Server installation, but can be configured to be separate) • Domain directory • JMS SAF files • File backed HTTP sessions Everything that uses the persistent store, such as JMS SAF files, has sensitive data that should be protected from read access as well as from write access. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. If you use the file store to store files on WebLogic Server, the applications can be stored anywhere. You must remember the locations of all of the files in order to protect them from read and write access. If you use the JDBC store to store applications, make sure to properly secure the database by protecting it from read and write access. For more information on using the persistent store, see Using the WebLogic Persistent Store in *Administering Server Environments for Oracle WebLogic Server*.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Configure the Password Validation provider immediately after configuring a new WebLogic domain	The Password Validation provider, which is included with WebLogic Server, can be configured with several out-of-the-box authentication providers to manage and enforce password composition rules. Consequently, whenever a password is created or updated in the security realm, the corresponding authentication provider automatically invokes the Password Validation provider to ensure that the password meets the composition requirements that are established. For information about how to configure and use the Password Validation provider, see Configuring the Password Validation Provider in *Administering Security for Oracle WebLogic Server*.
To bind to protected ports on UNIX, configure WebLogic Server to switch user IDs or use Network Address Translation (NAT) software.	On UNIX systems, only processes that run under a privileged user account (in most cases, root) can bind to ports lower than 1024. UNIX systems allow only one system administrator (root) user. However, long-running processes like WebLogic Server should not run under these privileged accounts. Instead, you can do either of the following: • For each WebLogic Server instance that needs access to privileged ports, configure the server to start under the privileged user account, bind to privileged ports, and change its user ID to a non-privileged account. If you use Node Manager to start the server instance, configure Node Manager to accept requests only on a secure port and only from a single, known host. Note that Node Manager needs to be started under a privileged user account. See Create and configure machines to run on UNIX in the *Oracle WebLogic Server Administration Console Online Help*. • Start WebLogic Server instances from a non-privileged account and configure your firewall to use Network Address Translation (NAT) software to map protected ports to unprotected ones.
Do not run Web servers as `root`	When you run a Web server on Unix systems — such as Apache HTTP Server, Microsoft IIS, or Sun Java System Web Server — make sure of the following: • The Web server should run only as an unprivileged user, never as `root`. • The directory structure in which the Web server is located, including all files, should be protected from access by unprivileged users. Taking these steps helps ensure that unprivileged users cannot insert code that can potentially be executed by the Web server.
Do not develop on a production machine.	Develop first on a development machine and then move code to the production machine when it is completed and tested. This process prevents bugs in the development environment from affecting the security of the production environment.
Do not install development or sample software on a production machine.	Do not install development tools on production machines. Keeping development tools off the production machine reduces the leverage intruders have should they get partial access to a WebLogic Server production machine. Do not configure the WebLogic Server sample applications on a production machine. When the installation program prompts you to select an installation type: • If you choose **WebLogic Server Installation**, **Coherence Installation**, or **Fusion Middleware Infrastructure**, the sample applications are not available for being configured post-installation. • If you choose **Complete Installation** or **Fusion Middleware Infrastructure With Examples**, the sample applications are available for being configured post-installation. This selection should be avoided on production machines.
Enable security auditing.	If the operating system on which WebLogic Server runs supports security auditing of file and directory access, Oracle recommends using audit logging to track any denied directory or file access violations. Administrators should ensure that sufficient disk space is available for the audit log.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Consider using additional software to secure your operating system.	Most operating systems can run additional software to secure a production environment. For example, an Intrusion Detection System (IDS) can detect attempts to modify the production environment. Refer to the vendor of your operating system for information about available software.
Apply operating system patch sets and security patches.	Refer to the vendor of your operating system for a list of recommended patch sets and security-related patches.
Ensure that the WebLogic Server version and patch set you are using is actively supported and under error correction.	New bug fixes, including fixes for security vulnerabilities, are only provided for product versions and patch sets that are under Premier or Extended Support, and are also under error correction. To verify that your WebLogic Server version is under Premier or Extended Support, refer to the Oracle Lifetime Support Policy for Oracle Fusion Middleware. To verify that your WebLogic Server version and patch set is under error correction, refer to the Oracle Error Correction Policy as documented in the My Oracle Support document *Error Correction Support Dates for Oracle WebLogic Server (Doc ID 950131.1)*. You should proactively plan to upgrade the WebLogic Server version and patch set you are using as required to ensure that it remains under Premier or Extended Support and under error correction.
Install the latest Patch Set Updates (PSUs).	Fixes for WebLogic Server security vulnerabilities are included in WebLogic Server Patch Set Updates (PSUs), released with the Critical Patch Update (CPU) program. PSUs are issued for WebLogic Server versions and patch sets that are actively supported and under error correction, on a planned schedule, per the Critical Patch Updates, Security Alerts and Bulletins. Oracle recommends that you schedule the installation of these PSUs, and apply them in as timely a manner as possible after they are released. If you are responsible for security-related issues at your site, register your WebLogic Server installation with Oracle Support and create a My Oracle Support account at https://support.oracle.com. When PSUs are released, their content is documented in the My Oracle Support document *Patch Set Update (PSU) Release Listing for Oracle WebLogic Server (WLS) (Doc ID 1470197.1)*. For additional information about WebLogic Server security vulnerabilities, see the My Oracle Support document *Security Vulnerability FAQ for Oracle Database and Fusion Middleware Products (Doc ID 1074055.1)*.
Maintain the security of the JDK and JVM versions used on the production system.	Ensure that the JDK and JVM versions are certified with WebLogic Server as listed in Oracle Fusion Middleware Supported System Configurations, are currently supported by their vendors, and have the latest security updates applied. For users of Oracle JDKs and JVMs, we strongly recommend: • Using JDK and JVM versions that are currently supported per the Oracle Java SE Support Roadmap. • Applying the latest Java Critical Patch Updates (CPUs) as soon as they are released. The Critical Patch Updates, Security Alerts and Bulletins page references the latest *Patch Availability Document for Oracle Java SE* documents that are available on My Oracle Support.

Table 3-1 (Cont.) Securing the WebLogic Server Host

Security Action	Description
Do not run WebLogic Server in development mode in a production environment.	Production mode sets the server to run with settings that are more secure and appropriate for a production environment. **Caution:** Note the following: • When WebLogic Server is configured in development mode, certain error conditions, such as a misbehaving application or an invalid configuration of WebLogic Server, may result in a trace stack being displayed. While error responses generally are not dangerous, they have the potential to give attackers information about the application or the WebLogic Server installation that can be used for malicious purposes. However, when WebLogic Server is configured in production mode, stack traces are not generated; therefore, you should never run WebLogic Server in development mode in a production environment. • Oracle recommends that you *not* enable the Web Services Test Client in production mode. For more information about the Web Services Test Client, see Testing Web Services in *Administering Web Services*. For more information about development mode and production mode, see Development and Production Modes in *Understanding Domain Configuration for Oracle WebLogic Server*. For information about how to change the WebLogic Server instances in a domain to run in production mode, see Change to production mode in the *Oracle WebLogic Server Administration Console Online Help*. You can also enable production mode using the WebLogic Scripting Tool by setting `DomainMBean.isProductionModeEnabled` MBean attribute to `true`.
Secure your JNDI root context	Group `Everyone` must not have access to the JNDI Root Content resource if the WebLogic Server Administration Console is externally visible. By default, JNDI resources have a default group security policy of `Everyone`.
Enable "most secure" values for WebLogic Server MBeans	WebLogic Server contains a number of MBeans that have attributes that affect the security of the WebLogic domain. Not all default values of these attributes are the most secure, so Oracle recommends setting them to their secure values in a production environment. For a complete list of these attributes and their most secure values, see Secure Values for MBean Attributes in MBean Reference for Oracle WebLogic Server.

3.3 Securing Network Connections

When designing network connections, you balance the need for a security solution that is easy to manage with the need to protect strategic WebLogic resources. The following table describes options for securing your network connections.

Table 3-2 Securing Network Connections

Security Action	Description
Use hardware and software to create firewalls.	A firewall limits traffic between two networks. Firewalls can be a combination of software and hardware, including routers and dedicated gateway machines. They employ filters that allow or disallow traffic to pass based on the protocol, the service requested, routing information, packet content, and the origin and destination hosts or networks. They can also limit access to authenticated users only. The WebLogic Security Service supports the use of third-party Identity Assertion providers, which perform perimeter-based authentication (Web server, firewall, VPN) and handle multiple security token types/protocols (SOAP, IIOP-CSIv2). For more information, refer to Perimeter Authentication in *Understanding Security for Oracle WebLogic Server*. For more information about using firewalls with WebLogic Server, refer to Security Options for Cluster Architectures in *Administering Clusters for Oracle WebLogic Server*.
Use WebLogic Server connection filters.	Instead of, or in addition to, using hardware and third-party software to create firewalls, consider using WebLogic Server connection filters to limit network traffic based on protocols, IP addresses, and DNS node names. Connection filters are most appropriate when the machines in a WebLogic Server domain can access each other without going through a firewall. For example, you might use a firewall to limit traffic from outside the network, and then use WebLogic Server connection filters to limit traffic behind the firewall. See Using Connection Filters in *Administering Security for Oracle WebLogic Server*.
Use a domain-wide Administration Port for administrative traffic.	An Administration Port limits all administrative traffic between server instances in a WebLogic Server domain to a single port. When the server is run without an Administration Port, an application can inadvertently transmit confidential server configuration on the wire in clear-text. Running the server with an Administration Port significantly reduces the chances of this happening. Furthermore, having an Administrative Port configured is helpful should a Denial of Service (DoS) attack occur because the resources for handling requests for, and the limitations on Administration Port requests are separate from those of the rest of the server. When used in conjunction with a connection filter, you can specify that a WebLogic Server instance accepts administrative requests only from a known set of machines or subnets and only on a single port. Enabling the Administration Port requires clients to interact with the WebLogic Server Administration Console using SSL which protects sensitive data from being sniffed on the wire by an attacker and protects against some cross-site scripting attacks. See Configure the domain-wide administration port and Enable configuration auditing in the *Oracle WebLogic Server Administration Console Online Help*.

Table 3-2 (Cont.) Securing Network Connections

Security Action	Description
Secure the embedded LDAP port.	To protect the embedded LDAP port against brute force attacks, close off the embedded LDAP listen port using a connection filter in a single server configuration. While this does not protect the embedded LDAP port in a multiple server configuration, the default connection filter implementation supports filtering based on the source IP address which should be used to allow access only from servers that are part of the domain. As a result, only the machines in the domain can access the LDAP port. For more information on using connection filters, see Using Network Connection Filters in *Developing Applications with the WebLogic Security Service*.
Do not enable remote access to the JVM platform MBean server.	As of JDK 1.5, the JDK provides an MBean server (the platform MBean server) and a set of MBeans that contain monitoring information about the JVM. You can configure the WebLogic Server Runtime MBean Server to run as the platform MBean server, which enables JMX clients to access the JVM MBeans and WebLogic Server MBeans from a single MBean server connection. Remote access to the platform MBean server can be secured only by standard JDK security features (see `http://docs.oracle.com/javase/7/docs/technotes/guides/management/agent.html`). If you have configured the WebLogic Server Runtime MBean Server to be the platform MBean server, enabling remote access to the platform MBean server creates an access path to WebLogic Server MBeans that is not secured through the WebLogic Server security framework. If it is essential that remote JMX clients have access to the JVM MBeans, Oracle recommends that you access them through the WebLogic Server Runtime MBean Server. See Registering MBeans in the JVM Platform MBean Server in *Developing Manageable Applications Using JMX for Oracle WebLogic Server*.
Restrict application use of JDBC over RMI.	JDBC application calls made over RMI are not secure and may allow unrestricted access to the database. Oracle recommends configuring RMI JDBC security to disable JDBC application calls over RMI. To do so: • Set the `RmiJDBCSecurity` attribute on the `DataSourceMBean` to `secure`. • Ensure that the SSL Listen Port setting is enabled for the server in the **Configuration** > **General** page of the WebLogic Server Administration Console. Note that RMI JDBC security does not disable Logging Last Resource, One Phase Commit, and Emulate Two Phase Commit data source transaction participants that span servers.

Table 3-2 (Cont.) Securing Network Connections

Security Action	Description
Configure Cross-Domain Security for JTA communication.	Communication channels must be secure to prevent a malicious third-party from using man-in-the-middle attacks to affect transaction outcomes and potentially gaining administrative control over one or more domains. To ensure secure communication channels between domains, WebLogic Server supports a type of domain trust that is referred to as Cross-Domain Security. Cross-Domain Security establishes trust between two domains - a domain pair - such that principals in a subject from one WebLogic domain can make calls in another domain. WebLogic Server establishes a security role for cross-domain users, and uses the WebLogic Credential Mapping security provider in each domain to store the credentials to be used by the cross-domain users. For more information and configuration details, see: • Cross Domain Security in *Developing JTA Applications for Oracle WebLogic Server 12.1.3* • Configuring Cross-Domain Security in *Administering Security for Oracle WebLogic Server 12.1.3*
When enabling SNMP, be sure to configure and use SNMPv3 protocol.	By default, Simple Network Management Protocol (SNMP) is disabled in WebLogic Server. However, once you enable SNMP, the SNMPv1 and SNMPv2 protocols are enabled. SNMPv1 and SNMPv2 are not secure and can cause certain potential security problems to occur on the SNMP service, including unauthorized access and Denial of Service attacks. Oracle strongly recommends disabling the SNMPv1 and SNMPv2 protocols and using the SNMPv3 protocol instead. When using the SNMPv3 protocol, additional security configuration is required because both the SNMP agent and manager must encode identical credentials in their protocol data units (PDUs) for the communication to succeed. If you cannot use SNMPv3, you must limit the weak security problems in SNMP v1 and SNMPv2 by ensuring that your network is secure and that the firewall is configured to restrict access to the ports in your WebLogic Server environment. See the following topics in *Monitoring Oracle WebLogic Server 12.1.3 with SNMP* for details: • Disabling SNMPv1 and v2 • Configuring Security for SNMPv3

Table 3-2 (Cont.) Securing Network Connections

Security Action	Description
Make sure configuration settings for complete message timeout are sized appropriately for your system.	To reduce the potential for Denial of Service (DoS) attacks, make sure that the complete message timeout parameter is configured properly for your system. This parameter sets the maximum number of seconds that a server waits for a complete message to be received. The default value is 60 seconds, which applies to all connection protocols for the default network channel. This setting might be appropriate if the server has a number of high-latency clients. However, you should tune this to the smallest possible value without compromising system availability. If you need a complete message timeout setting for a specific protocol, you can alternatively configure a new network channel for that protocol. For information about displaying the WebLogic Server Administration Console page from which the complete message timeout parameter can be set, see Configure protocols in the *Oracle WebLogic Server Administration Console Online Help*.
On UNIX systems, set number of file descriptors appropriately for your system.	On UNIX systems, each socket connection to WebLogic Server consumes a file descriptor. To optimize availability, the number of file descriptors for WebLogic Server should be appropriate for the host machine. By default, WebLogic Server configures 1024 file descriptors. However, this setting may be low, particularly for production systems. Note that when you tune the number of file descriptors for WebLogic Server, your changes should be in balance with any changes made to the complete message timeout parameter. A higher complete message timeout setting results in a socket not closing until the message timeout occurs, which therefore results in a longer hold on the file descriptor. So if the complete message timeout setting is high, the file descriptor limit should also be set high. This balance provides optimal system availability with reduced potential for DoS attacks. For information about tuning the number of available file descriptors, consult your UNIX vendor's documentation.
Use the Java security manager to control access to MLet MBeans.	MLet MBeans allow a client user to upload the MBean implementation and then execute that implementation in WebLogic Server. Since any authenticated user can instantiate and invoke on them, WebLogic Server disables the use of MLet MBeans by default with the `ManagementAppletCreateEnabled` attribute of the JMX MBean. WebLogic Server does not recommend enabling the use of MLet MBeans. If you choose to enable MLet MBeans, then you should ensure that only authorized users can access the MLet MBeans by running with the Java security manager and using permissions to restrict access to the MLet MBeans. To grant MBean register permissions for the `javax.management.loading.MLet` MBean to authorized users with Administrator or Deployer roles, use the grant principal `weblogic.security.principal.WLSPolicyFileGroupPrincipalImpl` "Administrators" and "Deployers" element.

Table 3-2 (Cont.) Securing Network Connections

Security Action	Description
Restrict incoming serialization data.	Although it is a useful feature, serialization in Java can also be used to inject malicious code using serialized Java objects that can cause Denial of Service (DoS) or Remote Code Execution (RCE) attacks during deserialization. WebLogic Server uses the JDK JEP 290 mechanism to filter incoming serialized Java objects to protect against these malicious attacks. **Note:** JEP 290 support is provided in WebLogic Server 12.1.3 with the following Patch Set Update (PSU): • WebLogic Server October 2017 PSU or later. For details about the WebLogic Server PSUs, see the My Oracle Support document *Patch Set Update (PSU) Release Listing for Oracle WebLogic Server (WLS) (Doc ID 1470197.1)*. JEP 290 integration is available when WebLogic Server 12.1.3 is configured with the following JDK versions: • JDK 8 Update 121 (JDK 8u121) or later • JDK 7 Update 131 (JDK 7u131) or later Oracle WebLogic Server 12.1.3 was certified on Java SE 7 when it was first made generally available. In addition, Oracle WebLogic Server 12.1.3 has been certified on Java SE 8 as described in **Supported Configurations** in *What's New in Oracle WebLogic Server 12.1.3.0*. If you are running a JDK that does not support JEP 290, then WebLogic Server continues to run and provide some protection against known deserialization attacks, but will not have the protection of the more advanced features of JEP 290. WebLogic Server uses JEP 290 as follows: • Implements a WebLogic Server-specific object input filter to enforce a blacklist of prohibited classes and packages for input streams used by WebLogic Server. The filter also

Table 3-2 (Cont.) Securing Network Connections

Security Action	Description
	enforces a default value for the maximum depth of a deserialized object tree. • Provides system properties that you can use to add or remove classes and packages from the default filter to blacklist or whitelist particular classes. You can also use the system properties to filter deserialized classes based on the nesting depth of the deserialized object, the number of internal references in the deserialized object, the size of object arrays, and/or the maximum size in bytes of a deserialized object. The settings in the WebLogic Server default filter can change over time. For a list of the most current default filter settings and the system properties that you can use to customize the filter, see the My Oracle Support document *Restricting Incoming Serialized Java Objects to Oracle WebLogic Server (Doc ID 2421487.1).* You can access My Oracle Support at https://support.oracle.com/. To ensure that your system is protected with the most current default filter, be sure to apply the latest Java and WebLogic Server Critical Patch Updates (CPUs) as soon as they are released. The Critical Patch Updates, Security Alerts and Bulletins page references the latest Java and WebLogic Server updates that are available on My Oracle Support. For more information about JEP 290, see `http://openjdk.java.net/jeps/290`.

3.4 Securing Your Database

Most Web applications use a database to store their data. Common databases used with WebLogic Server are Oracle, Microsoft SQL Server, and Informix.
The databases frequently hold the Web application's sensitive data including customer lists, customer contact information, credit card information, and other proprietary data. When creating your Web application you must consider what data is going to be in the database and how secure you need to make that data. You also need to understand the security mechanisms provided by the manufacturer of the database and decide whether they are sufficient for your needs. If the mechanisms are not sufficient, you can use other security techniques to improve the security of the database, such as encrypting sensitive data before writing it to the database. For example, leave all customer data in the database in plain text except for the encrypted credit card information.

3.5 Securing the WebLogic Security Service

The WebLogic Security Service provides a powerful and flexible set of software tools for securing the subsystems and applications that run on a server instance. The following table provides a checklist of essential features that Oracle recommends you use to secure your production environment.

Table 3-3 Securing the WebLogic Security Service

Security Action	Description
Deploy production-ready security providers to the security realm.	The WebLogic Security Service uses a pluggable architecture in which you can deploy multiple security providers, each of which handles a specific aspect of security. By default WebLogic Server includes its own security providers that provide a complete security solution. If you have purchased or written your own security providers: • Make sure that you have deployed and configured them properly. You can verify which security providers are currently deployed in the WebLogic Server Administration Console. In the left pane, select Console, select Security Realms, then click on the name of the realm and select the Providers tab. • Make sure that the realm in which you deployed your security providers is the default (active) realm. For instructions on how to set the default security realm in the WebLogic Server Administration Console, see Change the default security realm in the *Oracle WebLogic Server Administration Console Online Help*. • Refer to Customizing the Default Security Configuration in *Administering Security for Oracle WebLogic Server*.
Use SSL, but do not use the demonstration digital certificates in a production environment.	To prevent sensitive data from being compromised, secure data transfers by using HTTPS. WebLogic Server includes a set of demonstration private keys, digital certificates, and trusted certificate authorities that are for development only. Everyone who downloads WebLogic Server has the private keys for these digital certificates. Do not use the demonstration identity and trust. Refer to Configure keystores in the *Oracle WebLogic Server Administration Console Online Help* and Configuring SSL in *Administering Security for Oracle WebLogic Server*.
Make sure that WebLogic Server enforces security constraints on digital certificates.	When communicating by SSL, by default WebLogic Server rejects any digital certificates in a certificate chain that do not have the Basic Constraint extension defined by the Certificate Authority. This level of enforcement protects your Web site from the spoofing of digital certificates. Make sure that no server startup command includes the following option, which disables this enforcement: `-Dweblogic.security.SSL.enforceConstraints=false` For more information about this option, see SSL Certificate Validation in *Administering Security for Oracle WebLogic Server*. In your development environment, you might have disabled the enforcement of security constraints to work around incompatibilities with demonstration digital certificates that WebLogic Server provided in releases prior to 7.0 Service Pack 2. Make sure you enable this feature in your production environment.

Table 3-3 (Cont.) Securing the WebLogic Security Service

Security Action	Description
Verify that host name verification is enabled to avoid man-in-the-middle attacks.	By default, the WebLogic SSL implementation validates that the host to which a connection is made is the intended or authorized party. However, during the implementation of WebLogic Server at your site, you might have disabled host name verification. To enable host name verification, see Configure a custom host name verifier in the *Oracle WebLogic Server Administration Console Online Help*. Refer to Using Host Name Verification in *Administering Security for Oracle WebLogic Server*. *Background Information:* A man-in-the-middle attack occurs when a machine inserted into the network captures, modifies, and retransmits messages to the unsuspecting parties. One way to avoid man-in-the-middle attacks is to validate that the host to which a connection is made is the intended or authorized party. An SSL client can compare the host name of the SSL server with the digital certificate of the SSL server to validate the connection. The WebLogic Server HostName Verifier protects SSL connections from man-in-the-middle attacks.
Restrict the size and the time limit of requests on external channels to prevent Denial of Service attacks.	To prevent some Denial of Service (DoS) attacks, WebLogic Server can restrict the size of a message as well as the maximum time it takes a message to arrive. The default setting for message size is 10 megabytes and 480 seconds for the complete message timeout. Oracle recommends that you: • Set the size limit of requests on internal channels so that a Managed Server can to accept messages from the Administration Server. • Restrict the size and time limits of requests on external channels. • Configure internal channels so that they are only accessible internally and not externally. • Configure external channels to support only the protocols used by external clients to reduce the attack surface. In most cases, the supported protocols are HTTP and HTTPS. In addition, Oracle does not recommend enabling tunneling on channels that are available external to the firewall. To configure these settings for the HTTP, T3, and IIOP protocols refer to the following tasks in the *Oracle WebLogic Server Administration Console Online Help*: • Configure HTTP protocol • Configure T3 Protocol • Enable and configure IIOP See also Reducing the Potential for Denial of Service Attacks in *Tuning Performance of Oracle WebLogic Server*. *Background Information:* A DoS attack leaves a Web site running but unusable. Hackers deplete or delete one or more critical resources of the Web site. To perpetrate a DoS attack on a WebLogic Server instance, an intruder bombards the server with many requests that are very large, are slow to complete, or never complete so that the client stops sending data before completing the request.

Table 3-3 (Cont.) Securing the WebLogic Security Service

Security Action	Description
Set the number of sockets allowed to a server to prevent DoS attacks.	To prevent some DoS attacks, limit the number of sockets allowed for a server so that there are fewer than the number of sockets allowed to the entire process. This ensures that the number of file descriptors allowed by the operating system limits is not exceeded. Even after the server's limit is exceeded, administrators can access the server through the Administration Port. You can configure this setting using the MaxOpenSockCount flag. In the *Oracle WebLogic Server Administration Console Online Help*, see Servers: Configuration: Tuning.
Configure WebLogic Server to avoid overload conditions.	Configure WebLogic Server to avoid overload conditions in order to allow WebLogic Server sufficient processing power so that an administrator can connect to it and attempt to correct the problem in case the server comes under heavy load. Because communication over administration channels is not prevented when the system is overloaded, administrators can always connect regardless of any current overload condition. In case of heavy load, the administrator should bring the server into the admin state, locate the offending user, and then prevent that user from overloading the server with requests. To configure WebLogic Server to avoid overload conditions, set the shared capacity attribute in the overload protection MBean. The setting you choose for this attribute is the threshold after which no more non-administrator requests are accepted by WebLogic Server. For more information on overload conditions, see Avoiding and Managing Overload in *Administering Server Environments for Oracle WebLogic Server*.
Configure user lockouts and login time limits to prevent attacks on user accounts.	By default, the WebLogic Security Service provides security against dictionary and brute force attacks of user accounts. If during development you changed the settings for the number of invalid login attempts required before locking the account, the time period in which invalid login attempts have to take place before locking the account, or the amount of time the user account is locked, review the settings and verify that they are adequate for your production environment. **Note:** User lockout is effected by the WebLogic Security Service on a per-server basis. For example, a user who has been locked out of an application hosted on a given Managed Server (or cluster) is not necessarily locked out of the WebLogic Server Administration Console. Likewise, a user who has been locked out of the WebLogic Server Administration Console is not necessarily prevented from attempting to log in to an application hosted on a Managed Server. See Protect user accounts in the *Oracle WebLogic Server Administration Console Online Help*. *Background Information:* In a dictionary/brute force attack, a hacker sets up a script to attempt logins using passwords out of a “dictionary”. The WebLogic Server user lockout and login settings can protect user accounts from dictionary/brute force attacks.
If you use multiple Authentication providers, be sure to set the JAAS control flag correctly.	If a security realm has multiple Authentication providers configured, configure the order and precedence of *each* provider by setting the JAAS control flags. See Set the JAAS control flag in the *Oracle WebLogic Server Administration Console Online Help*.

Table 3-3 (Cont.) Securing the WebLogic Security Service

Security Action	Description
Enable security auditing.	Auditing is the process of recording key security events in your WebLogic Server environment. When the Auditing provider that the WebLogic Security Service provides is enabled, it logs events in *DomainName*\DefaultAuditRecorder.log. You enable an Auditing provider in the WebLogic Server Administration Console on the **Security Realms** > *RealmName* > **Providers** > **Auditing** page. See Configure Auditing providers in the *Oracle WebLogic Server Administration Console Online Help*. **Note:** Using an Auditing provider might adversely affect the performance of WebLogic Server even if only a few events are logged. Review the auditing records periodically to detect security breaches and attempted breaches. Noting repeated failed logon attempts or a surprising pattern of security events can prevent serious problems. If you develop your own custom Auditing provider and would like more information on posting audit events from a provider's Mbean, refer to Best Practice: Posting Audit Events from a Provider's MBean in *Developing Security Providers for Oracle WebLogic Server*.
Ensure that you have correctly assigned users and groups to the default WebLogic Server security roles.	By default, all WebLogic resources are protected by security policies that are based on a default set of security roles. Make sure you have assigned the desired set of users and groups to these default security roles. Refer to Users, Groups, And Security Roles in *Securing Resources Using Roles and Policies for Oracle WebLogic Server*.
Create no fewer than two user accounts with system administrator privileges.	One of the system administrator users should be created when the domain is created. Create other user(s) and assign them the Admin security role. When creating system administrator users give them unique names that cannot be easily guessed. Having at least two system administrator user accounts helps to ensure that one user maintains account access in case another user becomes locked out by a dictionary/brute force attack.

3.6 Securing Applications

Although much of the responsibility for securing the WebLogic resources in a WebLogic Server domain fall within the scope of the server, some security responsibilities lie within the scope of individual applications. For some security options, the WebLogic Security Service enables you to determine whether the server or individual applications are responsible. For each application that you deploy in a production environment, review the items in the following table to verify that you have secured its resources.

> **Note:**
>
> The HTTP Publish-Subscribe server included in WebLogic Server has specific lockdown steps, which are described in Using the HTTP Publish-Subscribe Server in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.

Table 3-4 Securing Applications

Security Action	Description
Determine which deployment model secures your Web applications and EJBs.	By default, each Web application and EJB uses deployment descriptors (XML files) to declare its secured resources and the security roles that can access the secured resources. Instead of declaring security in Web application and EJB deployment descriptors, you can use the WebLogic Server Administration Console to set security policies that secure access to Web applications and EJBs. This technique provides a single, centralized location from which to manage security for all Web applications and EJBs. You can combine these two techniques and configure WebLogic Server to copy security configurations from existing deployment descriptors upon the initial deployment of a URL (Web) or EJB resource. Once these security configurations are copied, the WebLogic Server Administration Console can be used for subsequent updates. Refer to Options for Securing Web Application and EJB Resources in *Securing Resources Using Roles and Policies for Oracle WebLogic Server*.
Set the `FrontendHost` attribute on the WebServerMBean or ClusterMBean to prevent redirection attacks	When a request on a web application is redirected to another location, the Host header contained in the request is used by default in the Location header of the response. Because the Host header can be spoofed — that is, corrupted to contain a different host name and other parameters — this behavior can be exploited to launch a redirection attack on a third party. To prevent the likelihood of this occurrence, set the `FrontendHost` attribute on either the WebserverMBean or ClusterMBean to specify the host to which all redirected URLs are sent. The host specified in the `FrontendHost` attribute will be used in the Location header of the response instead of the one contained in the original request. For more information, see `FrontendHost` in *MBean Reference for Oracle WebLogic Server*.
Use JSP comment tags instead of HTML comment tags.	Comments in JSP files that might contain sensitive data and or other comments that are not intended for the end user should use the JSP syntax of <%/* *xxx* */%> instead of the HTML syntax <!-- *xxx* -->. The JSP comments, unlike the HTML comments, are deleted when the JSP is compiled and therefore cannot be viewed in the browser.
Do not install uncompiled JSPs and other source code on the production machine.	Always keep source code off of the production machine. Getting access to your source code allows an intruder to find security holes. Consider precompiling JSPs and installing only the compiled JSPs on the production machine. For information about precompiling JSPs, refer to Precompiling JSPs in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.
Configure your applications to use SSL.	Set the `transport-guarantee` to `CONFIDENTIAL` in the `user-data-constraint` element of the web.xml file whenever appropriate. Refer to security-constraint in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.

Table 3-4 (Cont.) Securing Applications

Security Action	Description
Do not use the `Servlet` servlet.	Oracle does not recommend using the `Servlet` servlet in a production environment. Instead, map servlets to URIs explicitly. Remove all existing mappings between WebLogic servlets and the `Servlet` servlet from all Web applications before using the applications in a production environment. For information on mapping servlets, refer to Configuring Servlets in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.
Do not leave `FileServlet` as the default servlet in a production environment.	Oracle does not recommend using the `FileServlet` servlet as the default servlet a production environment. For information on setting up a default servlet, refer to Setting Up a Default Servlet in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.
Verify all WebLogic security policies.	In WebLogic Server 7.0, security policies replace ACLs and answer the question "who has access" to a WebLogic resource. Make sure that you have not removed security policies from WebLogic resources, and make sure that your security role assignments provide users the kind of access that you intend. Refer to *Securing Resources Using Roles and Policies for Oracle WebLogic Server*.
Examine applications for security.	There are instances where an application can lead to a security vulnerability. Many of these instances are defined by third-party organizations such as Open Web Application Security Project (see `http://www.owasp.org/index.php/Category:OWASP_Project` for a list of common problems). Of particular concern is code that uses Java native interface (JNI) because Java positions native code outside of the scope of Java security. If Java native code behaves errantly, it is only constrained by the operating system. That is, the Java native code can do anything WebLogic Server itself can do. This potential vulnerability is further complicated by the fact that buffer overflow errors are common in native code and can be used to run arbitrary code.

If your applications contain untrusted code, enable the Java security manager.	The Java security manager defines and enforces permissions for classes that run within a JVM. In many cases, where the threat model does not include malicious code being run in the JVM, the Java security manager is unnecessary. However, when third parties use WebLogic Server and untrusted classes are being run, the Java security manager may be useful.
	To enable the Java security manager for a server instance, use the following Java options when starting the server:
	`-Djava.security.manager`
	`-Djava.security.policy[=]=filename`
	Refer to Using the Java Security Manager to Protect WebLogic Resources in *Developing Applications with the WebLogic Security Service*.

Table 3-4 (Cont.) Securing Applications

Security Action	Description
Replace HTML special characters when servlets or JSPs return user-supplied data.	The ability to return user-supplied data can present a security vulnerability called *cross-site scripting*, which can be exploited to steal a user's security authorization. For a detailed description of cross-site scripting, refer to Understanding Malicious Content Mitigation for Web Developers (a CERT security advisory) at `http://www.cert.org/tech_tips/malicious_code_mitigation.html`. To remove the security vulnerability, before you return data that a user has supplied, scan the data for HTML special characters. If you find any such characters, replace them with their HTML entity or character reference. Replacing the characters prevents the browser from executing the user-supplied data as HTML. See Securing User-Supplied Data in JSPs and Securing Client Input in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.
Ensure security checks on performed on JMS resources.	Set the `weblogic.jms.securityCheckInterval` attribute to zero to ensure that an authorization check is performed for every Send, Receive, and `getEnumeration` action on a JMS resource.
Configure WebSocket applications to use authentication and authorization and verified-origin policies.	Use standard Web container authentication and authorization functionality (BASIC, FORM, CLIENT-CERT) to prevent unauthorized clients from opening WebSocket connections. You can also configure WebSocket applications to only accept WebSocket connections from expected origins. Apply a verified-origin policy to WebSocket applications by specifying the `Origin` HTTP header in the `accept` method of the `WebSocketListener` implementation class. For more information, see Securing WebSocket Applications in *Developing Applications for Oracle WebLogic Server*.
Establish secure WebSocket connections by using the `wss://` URI.	WebSocket applications should use the `wss://` URI to establish a secure WebSocket connection and prevent data from being intercepted. The `wss://` URI ensures that clients send handshake requests as HTTPS requests, encrypting transferred data by TLS/SSL. For more information, see Securing WebSocket Applications in *Developing Applications for Oracle WebLogic Server*.