Oracle® Fusion Middleware

Securing a Production Environment for Oracle WebLogic Server

14c (14.1.2.0.0)
F54011-01
December 2024

Oracle Fusion Middleware Securing a Production Environment for Oracle WebLogic Server, 14c (14.1.2.0.0)

F54011-01

Primary Author: Oracle Corporation

Contents

Preface

1 Getting Started

2 Understand and Secure Your Environment

3 Lock Down WebLogic Server

Preface

This document describes how to secure and lock down an Oracle WebLogic Server production environment.

Audience

This document is intended for application architects, security architects, application developers and server administrators who design, implement, and test the security of their WebLogic Server configuration.

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 customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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 WebLogic Server security, including settings for security realms, providers, identity and trust, SSL, and others.
- *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.

New and Changed WebLogic Server Features

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

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 Getting Started

Learn about locking down your WebLogic Server production environment and see a list of the critical tasks that you need to perform to ensure that your system is secure.
Topics include:

- Introduction
- Critical Tasks for Locking Down WebLogic Server

Introduction

To ensure the security of your production environment, it is critical that you lockdown your system to prevent unauthorized access to your WebLogic Server resources and applications.

Lockdown refers to configuring your system to prevent unwanted intrusions. A comprehensive lockdown of a WebLogic Server production environment includes securing the host machine and database, ensuring that you install only the necessary WebLogic Server components, and limiting access only to authorized users. Lockdown also includes other configuration such as securing your domain using a domain-wide secure port for Administration Server communications, securing network resources using network channels and firewalls to limit access, and configuring the system to use SSL.

Oracle strongly recommends that you follow *all* of the guidelines provided in this document to protect your WebLogic Server environment.

Critical Tasks for Locking Down WebLogic Server

To ensure the security of your system, Oracle strongly recommends that you complete these critical tasks to lockdown your WebLogic Server system.

> **Note:**
>
> Keep in mind that these are not the *only* tasks that you need to complete to lockdown your system. However, Oracle strongly recommends that these are the tasks that you must complete, but you should do them in combination with more general security guidelines described in Understand and Secure Your Environment and the other tasks described in Lock Down WebLogic Server.

Table 1-1 Critical Tasks for Locking Down WebLogic Server

Task	Description	More Information
Install WebLogic Server in a secure manner.	Performing a secure installation includes steps to secure the host machine on which WebLogic Server is installed, to limit access to that host to only authorized users, and to install only the components necessary to run WebLogic Server.	• Install WebLogic Server in a Secure Manner
Apply the latest WebLogic Server, Java, and database Critical Patch Updates on a quarterly basis.	To ensure that your system is protected against vulnerabilities, it is critical that you apply the latest Java, database, and WebLogic Server Critical Patch Updates (CPUs) as soon as they are released.	• Apply the Latest Patches and Updates
Configure your domains to use secured production mode.	Secured production mode sets more secure configuration default values, as compared to production mode. For example, SSL/TLS and the administration channel, along with many other configurations listed in this table, are enabled by default in secured production mode. While Oracle recommends the use of secured production mode, particularly for new environments, or as a baseline to compare existing production domains against, it is possible to achieve the same security posture by explicitly choosing secure settings, instead of enabling secured production mode. Customers should evaluate whether secured production mode will be helpful to achieve their desired security posture, and then enable or disable secured production mode accordingly.	• Configure Production or Secured Production Mode • Understand How Domain Mode Affects the Default Security Configuration
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. The administration port accepts only secure, SSL traffic, and all connections via the port require authentication. The administration port is enabled by default in secured production mode.	• Configure an Administration Port for the Domain

Table 1-1 (Cont.) Critical Tasks for Locking Down WebLogic Server

Task	Description	More Information
Set permissions to restrict the access of the user account used to run WebLogic Server to just the WebLogic resources and domain data stored on disk. Ensure that this account is not an administrator account.	WebLogic domain and server configuration files should be accessible only by the operating system users who configure and run 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. Knowledgeable operating system users may be able to bypass WebLogic Server security if they have access to domain data stored on disk and in the persistent store.	• Set Permissions to Restrict Access to WebLogic Resources to One User Account
Use network channels to isolate incoming application traffic. Use a firewall to limit access to only HTTPS application traffic and block access to non-HTTPS traffic (T3/T3s/LDAP/IIOP/IIOPs).	Oracle strongly recommends that you do not expose non-HTTPS traffic (T3/T3s/LDAP/IIOP/IIOPs) outside of the external firewall. You can control this access using a combination of network channels and firewalls.	• Configure Network Channels and Firewalls to Prevent Access from Non-HTTPS Traffic
Block access to internal applications by disabling unneeded applications and using a firewall to block access to internal application context paths.	Depending on your application usage and the domain configuration, some internal applications may not be used in a particular domain. To reduce the attack surface, Oracle strongly recommends that you configure a firewall to block external access to internal applications and disable access to these applications.	• Disable Unused Internal Applications • Configure Firewall to Prevent Access to Internal Applications
Use SSL/TLS, but do not use the demonstration digital certificates in a production environment.	Configure SSL/TLS for the administration port, network channels, database connections, LDAP server connections, and other resources handling communication that must be secured. In particular, make sure that connections to remote server instances in the domain are secured with SSL. WebLogic Server includes a set of demonstration private keys, digital certificates, and trusted certificate authorities that are for development only. Oracle highly recommends that you use third-party Certificate Authority (CA) signed certificates in a production environment.	• Configure SSL/TLS

Table 1-1 (Cont.) Critical Tasks for Locking Down WebLogic Server

Task	Description	More Information
Restrict incoming serialized objects.	Serialization in Java can 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.	• Use JEP 290 to Restrict Incoming Serialized Java Objects
Disable remote anonymous RMI T3 and IIOP requests.	Disabling remote anonymous T3 and IIOP RMI requests will require that clients authenticate before invoking on WebLogic Server. Unauthenticated clients will be rejected.	• Disable Remote Anonymous RMI T3 and IIOP Requests
Review the Security Warnings Report.	Check the Security Warnings Report to identify if any potential security issues are currently affecting your domain and how to resolve those issues.	• Review Potential Security Issues

2 Understand and Secure Your Environment

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 includes the following sections:

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.

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 `https://www.oracle.com/consulting/index.html`.

Read Security Publications

Staying current with security publications, such as those made available on My Oracle Support, is critical to maintaining a secure operational environment for WebLogic Server. Read about security issues:

- 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, Security Alerts and Bulletins page at the following location:

  `https://www.oracle.com/security-alerts/`
- When developing your web applications, ensure that they minimize the risks identified in the OWASP Top Ten Web Application Security Risks at `https://owasp.org/www-project-top-ten/`.

Secure the Host Environment

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

The following table lists the recommendations for securing a WebLogic Server host environment. Also check with the manufacturer of the machine and operating system for recommended security measures. For details about securing WebLogic Server, see Lock Down WebLogic Server.

Table 2-1 Secure the WebLogic Server Host Environment

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 2-1 (Cont.) Secure the WebLogic Server Host Environment

Security Action	Description
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. 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.
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 must 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. 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.
On each host computer, give only one operating system (OS) 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 must be accessible only by the operating system user who configures or executes 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. See Set Permissions to Restrict Access to WebLogic Resources to One User Account
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.

Table 2-1 (Cont.) Secure the WebLogic Server Host Environment

Security Action	Description
Do not run Web servers as `root`.	When you run a Web server on Unix systems — such as Apache HTTP Server, Microsoft IIS, or Oracle iPlanet Web Server — make sure of the following: • The Web server must run only as an unprivileged user, never as `root`. • The directory structure in which the Web server is located, including all files, must 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.
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 must ensure that sufficient disk space is available for the audit log.
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.

Secure Your Database

Most Web applications use a database to store their data. Common databases used with WebLogic Server are Oracle, Microsoft SQL Server, IBM DB2, and MySQL.

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

Lock Down WebLogic Server

Lock down WebLogic Server by performing a secure installation, designing your domains to use secure configurations, securing network resources using firewalls, and securing WebLogic resources and applications.
If you are installing WebLogic Server in a production environment, Oracle strongly recommends that you follow the guidelines described in these sections:

Install WebLogic Server in a Secure Manner

Creating a secure environment for WebLogic Server begins with planning a secure installation, which includes restricting access to the WebLogic Server host machine only to authorized users, and installing only the components of WebLogic Server that are needed in the target environment.

- Before you install WebLogic Server, be sure to secure the host machine, operating system, and file system to ensure that access is restricted only to authorized users. For specific recommendations, see Secure the Host Environment.
- When running the installation program, do not install WebLogic Server sample applications. For additional information, see Performing a Secure Installation of WebLogic Server in *Administering Security for Oracle WebLogic Server*.

Apply the Latest Patches and Updates

To ensure the security of your installed environment, Oracle strongly recommends that you apply the latest WebLogic Server, Java, and database Critical Patch Updates as soon as they are released.

The following table describes the patches and updates you need to apply to ensure that your WebLogic Server installation is protected by the latest software updates.

Table 3-1 Apply Latest Patch Sets and Updates

Security Action	Description
Install the latest Patch Set Updates (PSUs).	Fixes for WebLogic Server security vulnerabilities are included in WebLogic Server 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 strongly 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)*.
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 must proactively plan to upgrade the WebLogic Server version and patch set you are using as required to ensure that it will remain under Premier or Extended Support and under error correction.
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.

Configure a Secure Domain

Configuring a secure domain involves choosing the correct domain mode for your environment, configuring a password validation provider, configuring auditing and user lockouts, limiting the accounts with access to WebLogic resources, and so on.

Topics include:

- Understand How Domain Mode Affects the Default Security Configuration
- Configure Production or Secured Production Mode
- Configure an Administration Port for the Domain
- Disable Unused Internal Applications
- Configure Additional Security Settings After Domain Creation
- Set Permissions to Restrict Access to WebLogic Resources to One User Account
- Do Not Include Unencrypted Passwords in Commands and Scripts
- Secure WebLogic Resources
- Review Potential Security Issues

Understand How Domain Mode Affects the Default Security Configuration

A domain mode determines the default security configuration for your domain.

The domain modes, in order from least to most secure default values, are:

- Development mode
- Production mode
- Secured production mode

In development mode, the security configuration is most relaxed. You can start the Administration Server using a boot identity file or deploy an application using the `autodeploy` directory. In production mode, the default security configuration are set to more secure values, such as requiring a user name and password to deploy applications and start the Administration Server. In secured production mode, the default values of the security configuration are more strict, insecure configuration items are logged as warnings, and default authorization and role mapping policies are more restrictive.

A domain mode specifies the *default* values of a domain's security configuration. You can customize the behavior of the different domain modes by setting attribute values that override the default values. For example, you can enable the Administration port in a development mode domain or disable SSL/TLS in a secured production mode domain.

For instructions on how to change the domain mode of WebLogic Server instances, see Change the Domain Mode in *Oracle WebLogic Remote Console Online Help*. You can also use the WebLogic Scripting Tool to specify the domain mode. Set the `DomainMBean.ProductionModeEnabled` attribute to `true` to enable production mode. See DomainMBean in *MBean Reference for Oracle WebLogic Server*.

Note:

As of WebLogic Server 14.1.2.0.0, when you select production mode as the domain mode, it automatically enables the stricter default settings of secured production mode. If you want to implement the more moderate default settings of production mode, you must explicitly disable secured production mode on each server instance.

Table 3-2 describes how the default security and performance-related configuration parameters differ depending on whether your domain is configured in development mode, production mode, or secured production mode.

Note:

WebLogic Server automatically checks if your domain meets certain security standards. For each potential security issue in a domain, a warning is logged and displayed in the Security Warnings Report in WebLogic Remote Console. As a domain progresses from development to production to secured production mode, the security validation checks become more strict.

Table 3-2 describes how the security and performance-related configuration parameters differ when switching from one domain mode to another. It contains four columns. The first column lists the security or performance-related feature such as SSL/TLS, Administration port, listen port, and so on. The second column describes the impact of development mode on that feature. The third and fourth columns describe how the feature is affected when the domain is configured in production mode and secured production mode, respectively.

Table 3-2 Differences in Domain Modes

Feature	Development Mode	Production Mode	Secured Production Mode
SSL/TLS	You can use the demonstration digital certificates and the demonstration keystores provided by the WebLogic Server security services. With these certificates, you can design your application to work within environments secured by SSL/TLS. See Overview of Configuring SSL in WebLogic Server in *Administering Security for Oracle WebLogic Server*.	Demonstration digital certificates and the keystores are not recommended in production mode. If you do so, a warning message appears.	In this mode, WebLogic Server logs a warning if the SSL/TLS configuration is insecure. Also, the Administration Server will not start if the SSL Identity certificate is expired. WebLogic Server validates the minimum SSL/TLS version, constraints, and ciphers.

Table 3-2 (Cont.) Differences in Domain Modes

Feature	Development Mode	Production Mode	Secured Production Mode
Administration port	The administration port is disabled by default.	The administration port is disabled by default. To enable an administration port for your domain, see Configure the Domain-Wide Administration Port in *Oracle WebLogic Remote Console Online Help*.	The administration port is enabled by default. The administrative traffic is no longer allowed on the non-administration ports. If you want to connect to the Administration Server using WLST, you must specify the T3s protocol and the administration port. If you want to connect to the Administration Server using WebLogic Remote Console, you must specify the `https` protocol and the administration port. The default administration port is 9002.
Listen Port	The server listen port is enabled by default. The default port value is 7001.	The server listen port is enabled by default. The default port value is 7001.	The listen port is disabled by default. To enable and manage the listen port, see Specify Listen Ports in *Oracle WebLogic Remote Console Online Help*.
SSL/TLS listen Port	The SSL/TLS listen port is disabled by default.	The SSL/TLS listen port is disabled by default. You can enable the SSL/TLS listen port for servers in your domain. See Specify Listen Ports in *Oracle WebLogic Remote Console Online Help*.	The SSL/TLS listen port is enabled by default. The default port value is 7002.
Auditing	Security or configuration auditing is disabled by default.	Security or configuration auditing is disabled by default.	When the domain is created, the WebLogic Auditing provider is configured by default. Configuration changes are audited. WebLogic Server logs a warning if an Auditing provider is not configured.

Table 3-2 (Cont.) Differences in Domain Modes

Feature	Development Mode	Production Mode	Secured Production Mode
Deploying applications	WebLogic Server instances can deploy and update applications that reside in the *domain_name*/`autodeploy` directory automatically. Oracle recommends that you use this method only in a single-server development environment. See Deploying Applications and Modules with weblogic.deployer in *Deploying Applications to Oracle WebLogic Server*.	The auto-deployment feature is disabled. Use WebLogic Remote Console, the `weblogic.deployer` tool, or the WebLogic Scripting Tool.	The auto-deployment feature behavior in secured production mode is the same as in production mode.
Log file rotation	By default, when you start the WebLogic Server instance, the server automatically renames (rotates) its local server log file as `*SERVER-NAME*.log.*n*`. For the remainder of the server session, messages accumulate in the log file until the file grows to a size of 500 kilobytes. See Rotate Log Files in *Oracle WebLogic Remote Console Online Help* . The default value of the **Limit number of retained files** setting in Logging Configuration is `true`. This value limits the number of log files that the server instance creates to store old messages.	The server rotates the local log file after the size of the file reaches 5000 kilobytes. When the server is configured for production mode, by default, all versions of the log files are kept. Administrators may want to customize the number of log files that are retained. Use the LogFile MBean attributes to configure the location, file-rotation criteria, and number of files that a WebLogic Server instance uses to store log messages. The default value of the **Limit number of retained files** setting in Logging Configuration is `true`. The server creates 100 log files of 5 megabytes each. You must clean up the files as needed.	The log file rotation behavior in secured production mode is the same as in production mode.

Table 3-2 (Cont.) Differences in Domain Modes

Feature	Development Mode	Production Mode	Secured Production Mode
`boot.properties`	A `boot.properties` file is created, which allows you to boot the server without specifying a user name and password.	A `boot.properties` file is *not* automatically created. However, if it makes sense for your environment, you can manually create a `boot.properties` file. See Boot Identity Files in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.	The `boot.properties` file behavior in secured production mode is the same as in production mode.
Deployment of internal applications	For a development domain, the default is for WebLogic Server to deploy internal applications on the first access (on-demand).	For a production domain, the default is for WebLogic Server to deploy internal applications as part of server startup. If you want to change this behavior, see On-Demand Deployment of Internal Applications in *Deploying Applications to Oracle WebLogic Server*.	The deployment of internal applications in secured production mode is the same as in production mode.
Node Manager user name and password	In development mode, Node Manager uses the default user name and password credentials.	When a domain is created in production mode, then the user name and password for Node Manager are randomly generated. See Specifying Node Manager User Name and Password in *Administering Node Manager for Oracle WebLogic Server*.	In secured production mode, the Node Manager user name and password are generated the same way as in production mode.
Web Services Test Client	In a development environment, the Web Services Test Client is enabled, by default.	In a production environment, the Web Services Test Client is disabled (and not deployed), by default. It is recommended that you not enable the Web Services Test Client in production mode.	The default behavior of the Web Services Test Client in secured production mode is the same as in production mode.

Table 3-2 (Cont.) Differences in Domain Modes

Feature	Development Mode	Production Mode	Secured Production Mode
Classloader Analysis Tool	Classloader Analysis Tool (CAT) is deployed as an internal on-demand application only in development mode. Deployment happens upon first access.	If the server is running in production mode, it is not deployed automatically. You can deploy it in production mode; there are no limitations on its use, but you must deploy it manually, just like any other Web application. See Using the Classloader Analysis Tool (CAT) in *Developing Applications for Oracle WebLogic Server*.	The CAT tool behavior in secured production mode is the same as in production mode.
FastSwap deployment	You can use FastSwap deployment to minimize redeployment. FastSwap is only supported when WebLogic Server is running in development mode. See Using FastSwap Deployment to Minimize Redeploymentin *Deploying Applications to Oracle WebLogic Server*.	FastSwap is automatically disabled in production mode.	FastSwap is automatically disabled in secured production mode.
Application use of JDBC over RMI	In development mode, the `RmiJDBCSecurity` attribute on the `DataSourceMBean` is set to `Compatibility`, which allows applications to access JDBC objects over RMI.	In production mode, the default setting for the `RmiJDBCSecurity` attribute is the same as development mode, `Compatibility`. However, Oracle strongly recommends that you configure RMI JDBC security to disable JDBC application calls over RMI. To do so, set the `RmiJDBCSecurity` attribute on the `DataSourceMBean` to `Secure`.	In secured production mode, the `RmiJDBCSecurity` attribute on the `DataSourceMBean` is set to `Secure`, and all incoming application JDBC calls over RMI by remote clients and servers are rejected. RMI JDBC security does not disable Logging Last Resource, One Phase Commit, and Emulate Two Phase Commit data source transaction participants that span servers.
JMS File Store	A file store is automatically created in the file system.	The file store directory is not created automatically in the file system and users must manually create the directory with the necessary permissions.	The default behavior for the JMS file store is the same as in production mode.

Configure Production or Secured Production Mode

In production environments, you should select either production or secured production mode as the domain mode and then modify individual configurations as needed for your particular circumstances.

WebLogic Server contains a number of MBeans that have attributes that affect the security of the WebLogic domain. When you enable secured production mode, most of these attributes are set to the most secure value automatically. 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*.

It is unlikely that the default security configuration provided by either production mode or secured production mode will perfectly match the security requirements or realities of your environment. Therefore, you should select the domain mode that most closely matches your needs and then customize it using individual MBean attributes. For a comparison of the differences between production mode and secured production mode, see Understand How Domain Mode Affects the Default Security Configuration.

> **Note:**
>
> As of WebLogic Server 14.1.2.0.0, when you select production mode, it automatically enables secured production mode, which has stricter default settings. If you do not want secured production mode enabled, then you must explicitly disable it.

Certain MBean attributes cannot be set automatically and must be set to most secure value manually if required for your environment. For example, in secured production mode, SSL/TLS is enabled by default (`SSLMBean.Enabled=true`). However, the `TwoWaySSLEnabled` and `ClientCertificateEnforced` attributes on `SSLMBean` and `NetworkAccessPointMBean` are not automatically set to `true` because they require certificates from all clients. If two-way SSL/TLS is required, then you must configure these attributes manually.

For details about how to configure secure production domains, see Creating a WebLogic Domain for Production Use and Using Secured Production Mode in *Administering Security for Oracle WebLogic Server*.

> **Note:**
>
> When the Administration port is enabled (as in when secured production mode), the URL to connect to the Administration Server from WebLogic Remote Console is `https://`*`hostname`*`:`*`port`*. Note the *`s`* after `http`. The default port number in secured production mode is `9002`.

Configure an Administration Port for the Domain

Oracle strongly recommends that you use a domain-wide administration port for administrative traffic.

> **Note:**
>
> If your domain is configured to run in secured production mode, then the administration port is enabled by default and the administrative traffic is no longer allowed on the non-administration ports. In secured production mode, WebLogic Server logs a warning if the administration port has been explicitly disabled.

An administration port limits all administrative traffic between server instances in a WebLogic Server domain to a single port. If an administration port is enabled, WebLogic Server automatically generates an administrative channel for your domain, based on the port settings upon server instance startup. The administrative channel provides a listen address and listen port to handle administration traffic.

When the server is run without an administration port, a management client 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 WebLogic Remote Console using SSL which protects sensitive data from being sniffed on the wire by an attacker and protects against some cross site scripting attacks.

> **Note:**
>
> If multiple server instances run on the same computer in a domain that uses a domain-wide administration port, you must perform one of the following:
>
> - Host the server instances on a multi-homed machine and assign each server instance a unique listen address
> - Override the domain-wide port on all but one of the servers instances on the machine. In WebLogic Remote Console, on the **Environment**: **Servers**: *Server* page for each managed server, enter a unique port value in the **Local Administration Port Override** field.

See the following topics for more information:

- Configure the Domain-Wide Administration Port in *Oracle WebLogic Remote Console Online Help*
- Administration Port and Administrative Channel in *Administering Server Environments for Oracle WebLogic Server*

Disable Unused Internal Applications

Depending on your application usage and the domain configuration, some internal applications may not be used in a particular domain. Oracle strongly recommends that you disable access to these applications to reduce the attack surface.

You can disable unused internal applications using the configuration settings. Some internal applications are disabled by default; they must be enabled only if needed. The following table provides a list of internal applications that can be disabled and the process to disable them.

Table 3-3 Disabling Internal Applications

Internal Application	Process to Disable
Restful Services	If you disable Restful services, you cannot use WebLogic Remote Console to manage the domain. Set the `Enabled` attribute in the `RestfulManagementServicesMBean` to `false`, or, in WebLogic Remote Console, deselect the **Enable RESTful Management Services** option on the Environment: Domain: Management Services page.
WebLogic Remote Console Helper	If you disable the WebLogic Remote Console Helper, you cannot use web authentication to delegate the authentication of users from WebLogic Remote Console to an external authentication service. Set the `RemoteConsoleHelperEnabled` attribute in the `DomainMBean` to `false`, or, in WebLogic Remote Console, deselect the **Remote Console Helper Enabled** option on the Environment: Domain: General page.
Management EJB (Java EE Management APIs)	Set the `ManagementEJBEnabled` attribute in the `JMXMBean` to `false`.
Default Internal Servlets	Set the `DefaultInternalServletsDisabled` attribute in the `ServerMBean` to `true`. In secured production mode, this attribute is set to `true` by default and internal servlets are disabled.

Web Service Asynchronous Request-Response	Use the `OptionalFeatureMBean` to add an asynchronous request-response internal application, and set the feature to `false`. You can do this using WLST as shown in the following snippet:
	`optf =`
	`cmo.getOptionalFeatureDeployment()`
	`async = optf.createOptionalFeature`
	`async.setEnabled(false)`

Web Service Atomic Transactions (WSAT)	Use the `OptionalFeatureMBean` to add a WSAT internal application with the name `WSAT`, and set the feature to `false`. You can do this using WLST as shown in the following snippet:
	`optf =`
	`cmo.getOptionalFeatureDeployment()`
	`wsat =`
	`optf.createOptionalFeature("WSAT")`
	`wsat.setEnabled(false)`

Table 3-3 (Cont.) Disabling Internal Applications

Internal Application	Process to Disable

Ready App	Use the `OptionalFeatureMBean` to add a feature with the name `READYAPP`, and set the feature to `false`. You can do this using WLST as shown in the following snippet:
	`optf = cmo.getOptionalFeatureDeployment()`
	`ra = optf.createOptionalFeature("READYAPP")`
	`ra.setEnabled(false)`

Configure Additional Security Settings After Domain Creation

After you create your domain and configure you domain mode, there are a number of additional steps and configuration that you must complete to secure the domain.

The following table describes additional configuration and settings that you must configure to ensure the security of your domain.

Table 3-4 Additional Configuration and Settings to Secure the Domain

Security Action	Description
Create no fewer than two user accounts with system administrator privileges.	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. One of the system administrator users is created when you create the domain. 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. Avoid using obvious names such as `system`, `admin`, or `administrator`. **Note:** If you have enabled secured production mode, then WebLogic Server logs warnings if users in the administrator group have obvious user names such as `system`, `admin`, `administrator`, or `weblogic`.
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*.

Table 3-4 (Cont.) Additional Configuration and Settings to Secure the Domain

Security Action	Description
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 must 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 Configure Machines in *Oracle WebLogic Remote 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. **Note:** If you are using a domain that is running in secured production mode, then WebLogic Server logs a warning if the following are true: • Ports less than 1024 are used. • The `PostBindGIDEnabled` and `PostBindUIDEnabled` attributes of the `UnixMachineMBean` are *not* set to `true`.
Secure your JNDI root context.	Group `Everyone` must not have access to the JNDI Root Content resource if WebLogic Server is externally visible. By default, JNDI resources have a default group security policy of `Everyone`. **Note:** If secured production mode is enabled for your domain, then WebLogic Server does not allow remote anonymous JNDI access for list or modify operations. You can control anonymous JNDI access by setting the `RemoteAnonymousJNDIEnabled` attribute that is contained in the `SecurityConfigurationMBean`.

Table 3-4 (Cont.) Additional Configuration and Settings to Secure the Domain

Security Action	Description
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 must 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. See Avoiding and Managing Overload in *Administering Security 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 WebLogic Remote Console. Likewise, a user who has been locked out of WebLogic Remote Console is not necessarily prevented from attempting to log in to an application hosted on a Managed Server. See Set User Lockout Attributes in *Oracle WebLogic Remote 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. **Note:** If you have configured your domain to run in secured production mode, then WebLogic Server logs a warning if the user lockout is configured to a value less than the default value.

Table 3-4 (Cont.) Additional Configuration and Settings to Secure the Domain

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 WebLogic Remote Console. See Configure an Auditing Provider in *Oracle WebLogic Remote 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*. **Note:** If secured production mode is enabled for your domain, then WebLogic Server logs a warning if an audit provider is not configured. In this mode, the `DomainMBean.ConfigurationAuditType` attribute has a secure default value of `CONFIG_CHANGE_AUDIT`. Use the `SecureModeMBean.WarnOnAuditing` attribute to specify whether warnings should be logged if auditing is not enabled.
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. See Users, Groups, And Security Roles in *Securing Resources Using Roles and Policies for Oracle WebLogic Server*.
If you have a requirement to comply with Federal Information Processing Standards (FIPS) 140-2, ensure that FIPS mode is enabled.	FIPS mode is supported for JSSE using the Jipher JCE provider. FIPS 140-2 is a standard that describes U.S. Federal government requirements for sensitive, but unclassified use. To enable FIPS, see Enabling FIPS Mode in *Administering Security for Oracle WebLogic Server*.

Set Permissions to Restrict Access to WebLogic Resources to One User Account

On each host computer, give only one operating system (OS) user account access to WebLogic resources (in addition to the two system administrator users who also have access privileges) and set operating system file access permissions to restrict access to data stored on disk and in the persistent store.

Important: WebLogic domain and server configuration files must be accessible only by the operating system user who configures or executes WebLogic Server. No other operating system user (apart from the system administrators) should have read, write, or run access to WebLogic Server product files, or to your domain files.

> **Note:**
>
> By default WebLogic Server scripts use 027 as the umask, which allows read and execute access from other members in the group. To prevent access to WebLogic resources from other group members, ensure that the operating system user is the only member of the group.

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 this 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 directory and is named `wlserver`.
- **WebLogic domain directories**

  These directories contain the configuration files, security files such as `SerializedSystemIni.Dat`, 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/WebLogicServer/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.
- **Keystore directories**

  These directories include the private keystore and the Root Certificate Authority (CA) keystore that contain private keys, their associated digital certificates, and trusted CA certificates. See Storing Private Keys, Digital Certificates, and Trusted Certificate Authorities in *Administering Security for Oracle WebLogic Server*.
- **Application archive directories**

  These optional directories contain the application archives that are provisioned to WebLogic Server during deployment in the provisioning stage for the domain. These directories are separate from the WebLogic Server installation and domain directories.

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 must 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 files 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 data, make sure to properly secure the database by protecting it from read and write access.

> **Note:**
>
> If your domain is running in secured production mode and your file system supports POSIX, then WebLogic Server logs warnings if directories and files (such as domain directories, JMS SAF files, etc) have incorrect permissions. Use umask 027 as the minimum value when setting permissions.
>
> WebLogic Server scripts such as `$ORACLE_HOME/oracle_common/common/bin/wlst.sh` and `$ORACLE_HOME/oracle_common/common/bin/config.sh` specify a umask of 027, therefore any files and directories are created with the correct permissions.

Do Not Include Unencrypted Passwords in Commands and Scripts

Several WebLogic Server commands, including WLST and `weblogic.Deployer` commands, permit you to specify unencrypted passwords in the command line. Oracle strongly recommends that you do not include unencrypted passwords in command lines or scripts.

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.
- 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*

Secure WebLogic Resources

The WebLogic Security Service combines several layers of security features to prevent unauthorized access to your WebLogic Server resources such as JDBC, JMS or EJB resources.

To secure resources in your WebLogic Server domain, review the items in the following table.

Table 3-5 Securing WebLogic Resources

Security Action	Description
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`, which will reject all incoming application JDBC calls over RMI by remote clients and servers. 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. • Ensure that the SSL Listen Port setting is enabled for the server. See Specify Listen Ports in *Oracle WebLogic Remote Console Online Help*. **Note:** If you have configured the domain to run in secured production mode, then the `RmiJDBCSecurity` attribute is set to `Secure`.

Table 3-5 (Cont.) Securing WebLogic Resources

Security Action	Description
Configure 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 within and between domains, WebLogic Server supports local domain security and cross domain security. Local domain security establishes trust between servers within a domain. WebLogic Server establishes a security role for a local domain user, and uses the WebLogic Credential Mapping security provider to store the credentials to be used by the local domain user. 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: • Configuring Secure Inter-Domain and Intra-Domain Transaction Communication in *Developing JTA Applications for Oracle WebLogic Server* • Enable Local Domain Security for JTA in *Oracle WebLogic Remote Console Online Help* • Configuring Cross-Domain Security in *Administering Security for Oracle WebLogic Server*
Verify all WebLogic security policies.	In WebLogic Server, security policies 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. For information about various resource types, and how you can secure resource types using policies, see the following topics in *Securing Resources Using Roles and Policies for Oracle WebLogic Server*: • Understanding WebLogic Resource Security • Resource Types You Can Secure with Policies
Add JTA TransactionLoggable allowlist	The JTA TransactionLoggable allowlist is added to address a potential vulnerability with the JTA transaction log store implementation. When a TransactionLoggable object is written to the persistent store, the class name is persisted and used during recovery to instantiate a new instance of the TransactionLoggable class. The TransactionLoggable allowlist restricts the writing and reading of TransactionLoggable classes to and from the persistent store. The allowlist is disabled by default. When enabled, the allowlist is populated with a set of WLS-internal TransactionLoggable classes. • To enable the default allowlist, set the system property to `weblogic.transaction.loggable.allowList`. • To enable and add classes to the default allowlist, set the system property with a comma-separated list of fully qualified class names. For example, `weblogic.transaction.loggable.allowList=p1.ClassA,p2.ClassB`.

Review Potential Security Issues

WebLogic Server regularly validates your domain configuration settings against a set of security configuration guidelines to determine whether the domain meets key security guidelines recommended by Oracle.

If your domain does not meet a recommendation for a security configuration setting, a warning is logged in the Security Warnings Report in the WebLogic Remote Console. When there are active warnings in the Security Warnings Report, a banner with red text appears across the top of the WebLogic Remote Console. Click View/Refresh Report to see the report. In the Security Warnings Report, you will see any issues that need to be addressed and on which servers.

Warnings may appear for common issues that may indicate an insecure domain such as inadequate SSL/TLS configuration, outdated patch updates, or imminent certificate expiration. To protect your domain, resolve these warnings as consistent with your security and business requirements. You can resolve the warnings by updating your domain configuration settings to align with Oracle recommendations, or by disabling the security validation checks for that domain configuration setting.

Each warning in the Security Warnings Report includes a recommended solution for how to update the domain configuration setting. If you follow the recommended solution, the warning should be resolved. The same issue may affect multiple servers within your domain simultaneously. As you review the Security Warnings Report, make sure that you fix the issue on every affected server. Depending on the problem and its resolution, you may need to restart servers to update the Security Warnings Report.

For detailed advice on implementing the solutions identified, see Security Warning Resolution Suggestions in *Oracle WebLogic Remote Console Online Help*.

Although Oracle recommends resolving the warnings by changing the domain configuration setting, you may determine that based on your security and business requirements, certain warnings do not apply to your domain. For those warnings, you can disable the relevant security configuration settings. In WebLogic Remote Console, go to the Edit Tree: Environment: Domain page, choose the Security tab and then its Warnings subtab. Deselect any settings whose warnings you don't want to see.

You can also configure security configuration settings on the `SecureMode` MBean using WLST by navigating to the domain configuration and setting the relevant attributes to true or false. For example, using WLST:

```
edit()
startEdit()
cd("SecurityConfiguration/mydomain/SecureMode/mydomain")
cmo.setWarnOnAnonymousRequests(false)
activate()
```

The Certificate Expiry for identity and trust attributes can be configured through the `SecurityConfiguration` MBean.

WebLogic Server always verifies if your domain has the minimum required JDK version; you cannot disable the JDK version check.

Some level of security validation occurs in all domain modes. Validation is most strict in secured production mode and least strict in development mode. In secured production mode, almost all security configuration settings are enabled by default. Table 3-6 lists the security configuration settings.

Domains are scanned every 24 hours. You can also run the scan manually as needed.

Note:

Do not rely on the Security Warnings Report alone to determine the security of your domain. While these security configuration settings cover a broad set of potential security issues, other security issues that do not generate warnings may still exist in your domain.

Table 3-6 Security Validation Checks

Security Configuration Setting	Description	Applicable Domain Mode
Warn on Patches	Issues a warning if the domain does not have the latest WebLogic Server or Coherence critical patch update.	Production mode
Warn on Anonymous Requests	Issues a warning if anonymous request configuration attributes (`RemoteAnonymousRMIT3Enabled`, `RemoteAnonymousRMIIIOPEnabled`) are not disabled.	Production mode
Check Identity Certificates	Issues a warning if Identity certificates are set to expire within the period specified by the **Number of days before expiration for warnings** configuration setting.	Production mode
Check Trust Certificates	Issues a warning if Trust certificates are set to expire within the period specified by the **Number of days before expiration for warnings** configuration setting.	Production mode
Number of days before expiration for warnings	Enter (in days) how early WebLogic Server should warn of impending Identity or Trust certification expiration.	Production mode
Number of days between certificates checking	Enter (in days) how often WebLogic Server should check if the Identity or Trust certificates are set to expire.	Production mode
Warn on Insecure SSL	Issues a warning if SSL/TLS configuration is insecure. This includes checking for host verification, SSL versions, constraints, and so on.	Production mode
Warn on Insecure File System	Issues a warning if the file permissions in the domain directory are insecure.	Production mode
Warn on Insecure Ports	Issues a warning if the network port configuration is insecure.	Production mode
Warn on User Lockout	Issues a warning if user lockout settings are not secure.	Production mode

Table 3-6 (Cont.) Security Validation Checks

Security Configuration Setting	Description	Applicable Domain Mode
Warn on Username Passwords	Issues a warning if usernames or passwords do not meet recommended complexity standards.	Production mode
Warn on Insecure Applications	Issues a warning if applications are not secure.	Production mode
Warn on Auditing	Issues a warning if auditing is not enabled.	Secured production mode

Secure the Network

Secure the network in the production environment by using software and hardware to create firewalls, components such as network channels to isolate incoming and outgoing application traffic, and connection filters to deny access at the network level.

As part of securing the network, be sure to enable the administration port to limit all administrative traffic between server instances in a WebLogic Server domain to a single port. See Configure an Administration Port for the Domain.

Topics include:

- Configure Firewalls
- Configure Connection Filters
- Configure Timeouts
- Configure Sockets and File Descriptors
- Configure SSL/TLS
- Use JEP 290 to Restrict Incoming Serialized Java Objects
- Disable Remote Anonymous RMI T3 and IIOP Requests

Configure Firewalls

A firewall controls network traffic by acting as a barrier between a trusted and an untrusted network. Along with firewalls, you can use network channels, an administration port, WebLogic Server connection filters, and perimeter authentication to restrict access to resources based on user and network information.

Oracle strongly recommends that you:

- Configure a HTTPS protocol network channel to segregate HTTPS application traffic. Doing so ensures that HTTPS application traffic will run on a dedicated port by itself. Configure the firewall to allow external access to the HTTPS port, but block external access to any of the non-HTTPS ports.
- Configure internal channels for non-HTTPS protocols and use firewalls so that the internal channels are accessible only to trusted client IP addresses.
- Do not enable tunneling on channels.

Topics

Configure Network Channels and Firewalls to Prevent Access from Non-HTTPS Traffic

Oracle strongly recommends that you use a combination of network channels and firewalls to restrict external access to only HTTPS application traffic and that you block external access of non-HTTPS traffic (T3/T3s/IIOP/IIOPs).

Network channels define the attributes of a network connection to WebLogic Server, such as the protocol the network supports, the listen address, listen ports for secure and non-secure communication, and so on. Using network channels allows administrators to have more control over exposing network access to WebLogic Server. See Understanding Network Channels in *Administering Server Environments for Oracle WebLogic Server*.

Once you have defined a network channel, you can further isolate the network connections for that channel using a load balancer or firewall.

- To restrict the use of T3/T3s/IIOP/IIOPs protocols to *only* WebLogic servers and clients that are behind the firewall:
  1. Create a network channel to support only HTTPS traffic coming from the external applications. For the steps required to create a network channel, see Configure Custom Network Channels in *Oracle WebLogic Remote Console Online Help*.
  2. Configure the firewall so that the network channel that you created in the previous step is available externally, and that the default network channel and other customer internal channels are only accessible internally. Refer to your firewall documentation for the required steps.
  3. Do not enable tunneling on the externally available network channel. Tunneling is *not* enabled by default.
- If you already have existing network channels for HTTPS traffic from external applications, Oracle strongly recommends that you disable tunneling to avoid a T3 or IIOP call being wrapped inside the HTTPS protocol. If your existing network channel enables tunneling, disable it using WebLogic Remote Console:
  1. In the **Edit Tree** perspective, go to **Environment**, then **Servers**, then *myServer*, then **Channels**, then *myChannel*.
  2. On the **Channel General** tab, click **Show Advanced Fields** and then turn off the **Tunneling Enabled** option.
  3. Click **Save**.

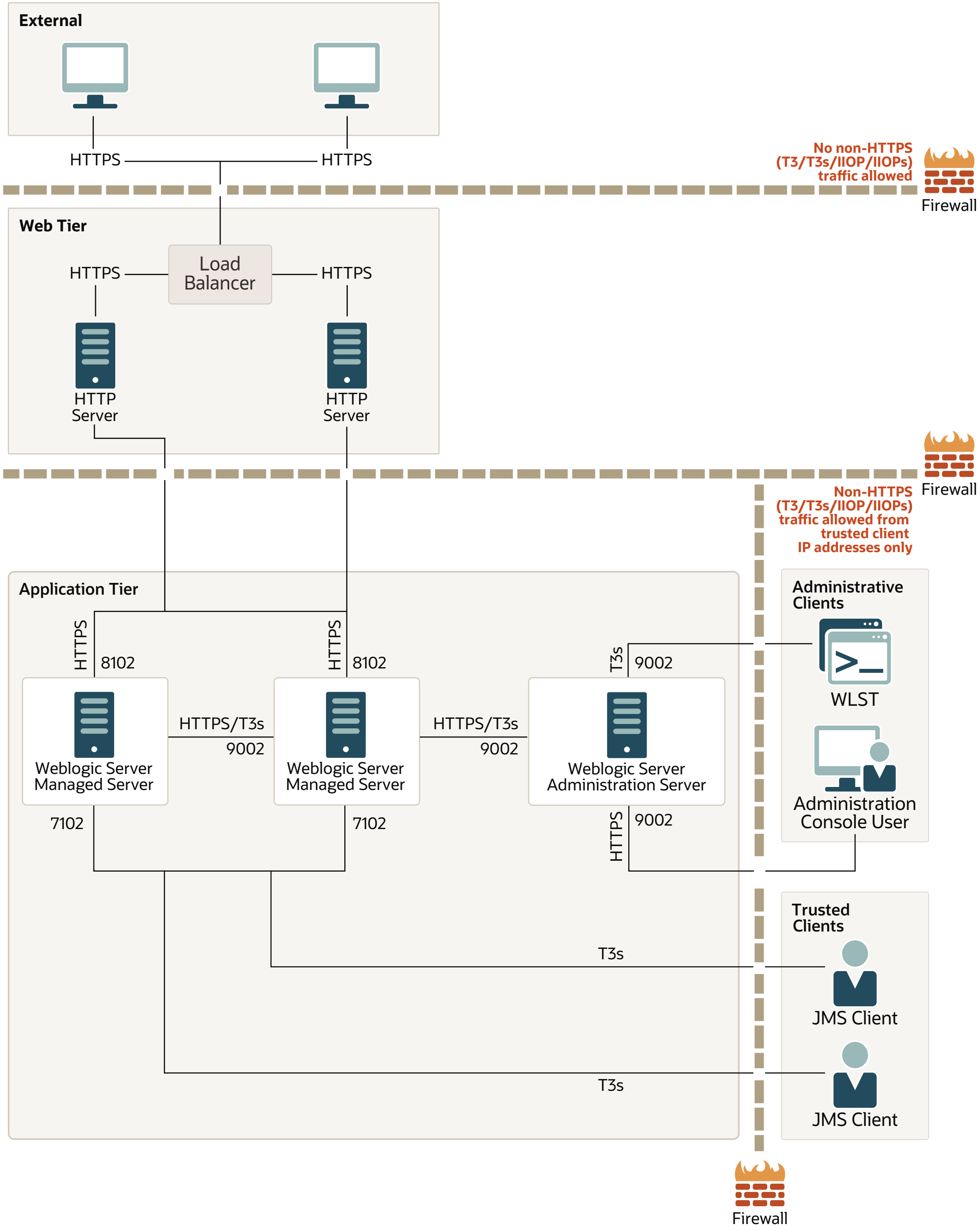
Figure 3-1 represents a secure WebLogic Server domain configuration using firewalls, network channels, and the administration port. This configuration includes:

- An external tier where users access the web tier through a firewall which is open only to HTTPS traffic. The firewall is configured to block access from non-HTTPS protocols such as T3/T3s/IIOP, and IIOPs.
- A web tier that consists of a load balancer and two HTTP servers. A firewall between the web tier and the application tier is configured to allow HTTPS traffic on port 8102 only.

- An application tier that includes a WebLogic Server domain configured with an Administration Server and two Managed Servers. The domain is configured using:
  - An administration port enabled and configured to use port 9002 (the default in secured production mode). The administration port separates administration traffic from application traffic in the domain and is used by each Managed Server in the domain exclusively for communication with the Administration Server. For more information about using an administration port, see Configure an Administration Port for the Domain.
  - Two network channels:
    - One network channel is configured on port 7102 to support only T3s traffic coming from trusted clients.
    - A second network channel is configured on port 8102 to support only HTTPS traffic coming from the external applications through the firewall.

      You can specify any available port number for the network channels.
- A firewall between the application tier and trusted application clients that is configured to allow administration traffic on port 9002 and T3s traffic on port 7102. Within the internal firewall:
  - Administrators in a trusted administrator group use WLST and WebLogic Remote Console with a set of trusted IP addresses to communicate with the Administration Server using the administration port 9002.
  - Trusted JMS and EJB clients running on a set of trusted IP addresses use the T3s protocol on port 7102 to communicate with the Managed Servers.

Figure 3-1 WebLogic Server Secure Configuration


External
HTTPS
HTTPS
No non-HTTPS (T3/T3s/IIOP/IIOPs) traffic allowed
Firewall
Web Tier
HTTPS
Load Balancer
HTTPS
HTTP Server
HTTP Server
Firewall
Non-HTTPS (T3/T3s/IIOP/IIOPs) traffic allowed from trusted client IP addresses only
Application Tier
HTTPS
8102
HTTPS
8102
T3s
9002
Administrative Clients
WLST
HTTPS/T3s
9002
HTTPS/T3s
9002
Weblogic Server Managed Server
Weblogic Server Managed Server
Weblogic Server Administration Server
Administration Console User
7102
7102
HTTPS
9002
Trusted Clients
T3s
JMS Client
T3s
JMS Client
Firewall

Configure Firewall to Prevent Access to Internal Applications

Enable the administration port for your domain, and configure a firewall to prevent external access to internal applications accessible on the Administration port. Using both the administration port and the firewall ensures that internal applications and RESTful services cannot be accessed externally.

To block access to internal applications:

1. Ensure that you have disabled any unused internal applications as described in Disable Unused Internal Applications.
2. Configure a firewall to limit access to internal applications that are accessible on the non-Administration ports, such as SAML and web services. To do so, disable access to the appropriate context paths.

The following table lists the WebLogic Server internal applications and their context paths.

Table 3-7 WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
File Distribution	Yes	`bea_wls_management_internal2`	Used for distributing the initial LDAP data to a Managed Server. Only Managed Servers access this internal application.
WebLogic Remote Console Helper	Yes	`console`	Facilitates web authentication for WebLogic Remote Console You can configure the Remote Console Helper using the `RemoteConsoleHelperMBean` or in WebLogic Remote Console, on the Environment: Domain page. Make sure to enable Show Advanced Fields. Users can change the context path.
WebLogic Server Test Client	Yes	`wls_utc`	Used to test web services without the need to write client code. The test client is disabled in production mode. WebLogic developers access this application from browsers on their client machines.
RESTful Services	Yes	`management`	Provides the REST API functionality used for management and monitoring. WebLogic Administrators, Deployers, Monitors, and Operators access the RESTful services from their client machines.

Table 3-7 (Cont.) WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
Deployment Service	Yes	`bea_wls_deployment_internal/DeploymentService`	Used to coordinate deployment and activate changes between the Administration Server and Managed Servers. This web application may be called by WLST or `weblogic.Deployer` clients to upload the application archive or plan. WebLogic Administration Servers and Managed Servers access this application. WebLogic Administrators, Deployers and Operator access this application using WLST and `weblogic.Deployer` on their client machines.
Cluster servlet	Yes	`bea_wls_cluster_internal`	Used for cluster communication including replication, saving state, and session recovery. Only cluster members use this API.
Internal servlets	No	`bea_wls_internal`	Used for tunneling RMI/IIOP over HTTP. This application is disabled by default. Managed Servers and WebLogic Server clients running on client machines access this application.
Web service async response	No	`_async`	Contains the web service Async Response Service. This application is disabled by default.
Web Service AT app	No	`wls-wsat`	Contains the WebLogic Server Web Services Atomic Transactions Service. This application serves as the transaction coordinator. Web service clients running on client machines access this application.

Table 3-7 (Cont.) WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
Classloader Analysis Tool	Yes	`wls-cat`	A web-based class analysis tool which simplifies filtering classloader configuration and aids you in analyzing classloading issues, such as detecting conflicts, debugging application classpaths and class conflicts, and proposes solutions to help you resolve them. This application is disabled in production domains. WebLogic application developers access this application from browsers on their client machines.
SAML ITS Apps Basic	No	`samlits_ba`	Supports the Intersite Transfer Service for basic authentication. This application is only enabled if the appropriate `FederationServiceMBean` `IntersiteTransferURIs` are configured. Application Single Sign-On (SSO) integration functions, SAML partners, and application users using web browsers can access these endpoints for SAML Single Sign-On (SSO) support with deployed applications. **Note**: The SAML 1.1 Identity Assertion provider, the SAML 1.1 Credential Mapping provider, and related configuration and services for SAML 1.1 federation services, are deprecated as of WebLogic Server 14.1.2.0.0 and will be removed in a future release. Oracle recommends using SAML 2.0 instead.

Table 3-7 (Cont.) WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
SAML ITS Apps Cert	No	`samlits_cc`	Supports the Intersite Transfer Service for client cert authentication. This application is only enabled if the appropriate `FederationServiceMBean` `IntersiteTransferURI`s are configured. Application Single Sign-On (SSO) integration functions, SAML partners, and application users using web browsers can access these endpoints for SAML Single Sign-On (SSO) support with deployed applications. **Note**: The SAML 1.1 Identity Assertion provider, the SAML 1.1 Credential Mapping provider, and related configuration and services for SAML 1.1 federation services, are deprecated as of WebLogic Server 14.1.2.0.0 and will be removed in a future release. Oracle recommends using SAML 2.0 instead.

Table 3-7 (Cont.) WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
SAML ACS Apps	No	`samlacs`	Supports the SAML Assertion Consumer Service. This application is only enabled if appropriate `FederationServiceMBean AssertionConsumerURI`s are configured. Application Single Sign-On (SSO) integration functions, SAML partners, and application users using web browsers can access these endpoints for SAML Single Sign-On (SSO) support with deployed applications. **Note**: The SAML 1.1 Identity Assertion provider, the SAML 1.1 Credential Mapping provider, and related configuration and services for SAML 1.1 federation services, are deprecated as of WebLogic Server 14.1.2.0.0 and will be removed in a future release. Oracle recommends using SAML 2.0 instead.

Table 3-7   (Cont.) WebLogic Server Internal Applications Context Paths

Internal Application	Administration Port Only (if enabled)	Context Path	Description
SAML ARS App	No	`samlars`	Listens for incoming assertion retrieval requests. This application is only enabled if `FederationServicesMBean AssertionRetrievalURIs` are configured for the `samlars` application. Application Single Sign-On (SSO) integration functions, SAML partners, and application users using web browsers can access these endpoints for SAML Single Sign-On (SSO) support with deployed applications. **Note**: The SAML 1.1 Identity Assertion provider, the SAML 1.1 Credential Mapping provider, and related configuration and services for SAML 1.1 federation services, are deprecated as of WebLogic Server 14.1.2.0.0 and will be removed in a future release. Oracle recommends using SAML 2.0 instead.
SAML2 Application	No	`saml2`	Contains the services used for SAML 2 support. This includes the SP Initiator, IdP SSO service, SP Assertion Consumer Service, and Artifact Resolution Service. This application is only enabled if `SingleSignOnServicesMBean ServiceProviderEnabled` or `IdentityProviderEnabled` attributes are set to `true`. Application Single Sign-On (SSO) integration functions, SAML partners, SP Single Logout, and application users using web browsers can access these endpoints for SAML Single Sign-On (SSO) support with deployed applications.

Configure Firewall for Cluster Communication

It is important to understand the communication between servers in a cluster so that you can configure firewalls appropriately.

WebLogic Server allows you to configure either multicast or unicast communication between cluster members. A firewall should allow the cluster network traffic from subnets with cluster members, but prevent it from other subnets. For more information about communications within a cluster, see Communications In a Cluster in *Administering Clusters for Oracle WebLogic Server*. In some cases, more complex port splitting may be required, especially if you use JMS or EJBs. In such cases, more than two ports may be necessary. Port splitting gives you the flexibility to define different firewall rules for different protocols. For example, if the IP of the remote client using the non-HTTPS protocol is known, a firewall rule based on that IP can be configured, assuming that the relevant non-HTTPS protocol is appropriately split out to its own port.

See Security Options for Cluster Architectures in *Administering Clusters for Oracle WebLogic Server*.

Configure Connection Filters

In addition to creating firewalls, use WebLogic Server connection filters to limit incoming connections.

When you use a connection filter, the connections to ports exposed externally come only from expected front-end hosts, and connections for administration traffic come only from the expected subnets where other WebLogic Servers are running.

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.

In a single server configuration, Oracle strongly recommends that you close off the embedded LDAP listen port using a connection filter to protect the embedded LDAP port against brute force attacks. 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 details about configuring connection filters, see Using Connection Filters in *Administering Security for Oracle WebLogic Server*.

Configure Timeouts

To reduce the potential for Denial of Service (DoS) attacks, make sure that you restrict message size, configure complete message timeouts appropriately for your system, and limit the number of sockets allowed for a server.

Table 3-8   Configure Secure Timeouts

Security Action	Description	More Information
Configure the `Complete Message Timeout` parameter appropriately for your system.	The `Complete Message Timeout` parameter sets the maximum number of seconds that a server waits for a complete message to be received. This timeout helps guard against a Denial of Service (DoS) attack in which a caller indicates that it will be sending a message of a certain size that it never finishes sending. The default value for this parameter 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.	See Configure Protocols in *Oracle WebLogic Remote Console Online Help*.

Table 3-8 (Cont.) Configure Secure Timeouts

Security Action	Description	More Information
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 accept messages from the Administration Server. • Restrict the size and time limits of requests on external channels. *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.	To configure these protocol settings, see : • Configure HTTP Protocol • Enable T3 Protocol • Configure IIOP In *Oracle WebLogic Remote Console Online Help*. See also Reducing the Potential for Denial of Service Attacks in *Tuning Performance of Oracle WebLogic Server*.

Configure Sockets and File Descriptors

To prevent DoS attacks, Oracle strongly recommends that you limit the number of sockets allowed for a server. To optimize availability on UNIX systems, be sure to set the number of file descriptors consumed by sockets to a number that is appropriate for your system.

Table 3-9 describes the actions that you need to take to set the number of sockets and file descriptors.

Table 3-9 Sockets and File Descriptors

Security Action	Description	More Information
Set the number of sockets allowed for 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.	To set **Maximum Open Sockets**, open WebLogic Remote Console and in the **Edit Tree**, go to **Environment**, then **Servers**, then *myServer*. On the **Advanced** tab, select the **Tuning** subtab.
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 must 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 must 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 must also be set high. This balance provides optimal system availability with reduced potential for DoS attacks.	• For more information about the complete message timeout parameter, see Configure Timeouts. • For information about tuning the number of available file descriptors, consult your UNIX vendor's documentation.

Configure SSL/TLS

To prevent sensitive data from being compromised, secure data transfers using SSL/TLS. SSL/TLS provides secure connections by allowing two applications connecting over a network to authenticate each other's identity and by encrypting the data exchanged between the applications.

Oracle strongly recommends that you configure SSL/TLS for the administration port, network channels, database connections, LDAP server connections, and other resources handling communication that must be secured. In particular, make sure that connections to remote server instances in the domain are secured with SSL/TLS. The specific components for which either one- or two-way SSL/TLS needs to be configured depends on the overall topology of the production environment. For details about configuring SSL/TLS, see Configuring SSL in *Administering Security for Oracle WebLogic Server*.

If you want to manage or disable the non-HTTPS listen port, see Specify Listen Ports in *Oracle WebLogic Remote Console Online Help*.

WebLogic Server also provides the ability to enable HTTP Strict Transport Security (HSTS), which is a web security policy mechanism that allows a web server to be configured so that web browsers, or other user agents, can access the server using only secure connections, such as HTTPS. For details about enabling HSTS in WebLogic Server, see Using HTTP Strict Transport Security in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.

Oracle strongly recommends that you do not allow the use of unencrypted null ciphers in a production environment. SSL/TLS clients start the SSL/TLS 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/TLS 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. However, an incorrectly configured client might specify a set of cipher suites that contain only null ciphers, which passes data on the wire in clear-text. The server selects the null cipher only when it is the only cipher suite that it has in common with the client, and includes a message in the server log that the SSL/TLS session is using a null cipher. WebLogic Server includes a WebLogic Remote Console control to prevent the server from using a null cipher. For more information about null ciphers and the WebLogic Remote Console control, see An Important Note Regarding Null Cipher Use in SSL in *Administering Security for Oracle WebLogic Server*.

Use JEP 290 to Restrict Incoming Serialized Java Objects

To improve security, WebLogic Server uses the JDK JEP 290 mechanism to filter incoming serialized Java objects and limit the classes that can be deserialized. The filter helps to protect against attacks from specially crafted, malicious serialized objects that can cause denial of service (DOS) or remote code execution (RCE) attacks.

There are two models to prevent deserialization exploits: allowlist and blocklist. With the blocklist model, WebLogic Server defines a set of well-known classes and packages that are vulnerable and blocks them from being deserialized, and all other classes can be deserialized. In the allowlist model, WebLogic Server and the customer define a list of the acceptable classes and packages that are allowed to be deserialized, and blocks all other classes. While both approaches have benefits, the allowlist model is more secure because it only allows deserialization of classes known to be required by WebLogic Server and customer applications.

You can choose whether to use blocklists or allowlists. For details about using both methods, see Using JEP 290 in Oracle WebLogic Server in *Administering Security for Oracle WebLogic Server*.

WebLogic Server uses blocklists by default. At startup, WebLogic Server configures a default JEP 290 blocklist filter that includes a set of prohibited classes and packages, and default values for some JEP 290 options. You can use WebLogic Server system properties to customize, replace, or disable the filter. The WebLogic Server also supports dynamic blocklists, which provide the ability to update your blocklist filters by creating a configuration file that can be updated or replaced while the server is running. See Using a Dynamic Blocklist Configuration File in *Administering Security for Oracle WebLogic Server*.

If you prefer, you can choose to use the allowlist model. First, you must create an allowlist that contains the classes and packages that are deserialized in the applications in your domain. To do so, you enable recording, which records all of the deserialized classes and packages for both WebLogic Server and application deserialization. When deserialization occurs, each class is recorded in an allowlist configuration file. When you are satisfied with the allowlist, you then configure WebLogic Server to use the allowlist configuration file for the JEP 290 filtering. See Using an Allowlist for JEP 290 Filtering in *Administering Security for Oracle WebLogic Server*.

WebLogic Server also provides the ability to log the current blocklist and allowlist classes and packages. See Enabling Filter Logging in *Administering Security for Oracle WebLogic Server*.

> **Note:**
>
> WebLogic Server Patch Set Updates (PSUs) may include updates to the set of prohibited classes and packages used in the default blocklist filter. To ensure that your system is protected against deserialization vulnerabilities with the most current default filter, be sure to apply the latest WebLogic Server PSUs and Java 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`.

Disable Remote Anonymous RMI T3 and IIOP Requests

You can disable anonymous RMI requests from clients.

The ability to disable anonymous requests from clients provides two benefits:

- Unauthenticated clients are rejected and are not allowed to invoke on WebLogic Server.
- If anonymous requests are disabled, then additional JEP 290 filtering is performed and helps protect against deserialization exploits.

To disable anonymous RMI T3 and IIOP requests, do one of the following:

- Use WebLogic Remote Console to disable the remote anonymous RMI T3 and IIOP requests:
  1. In the **Edit Tree**, go to **Environment**, then **Domain**.
  2. Click the **Security Tab**.
  3. Click **Show Advanced Fields**.
  4. Disable the **Remote anonymous RMI access via IIOP** and **Remote anonymous RMI access via T3** options.
  5. Click **Save**.
- Use WLST to set the `RemoteAnonymousRMIT3Enabled` and `RemoteAnonymousRMIIIOPEnabled` attributes to `false` to disable anonymous requests. (The default is `true`.) For example, using WLST online:

  ```
  edit()
  startEdit()
  cd("SecurityConfiguration/mydomain")
  cmo.setRemoteAnonymousRMIIIOPEnabled(false)
  cmo.setRemoteAnonymousRMIT3Enabled(false)
  activate()
  ```

- Set the `RemoteAnonymousRMIT3Enabled` and `RemoteAnonymousRMIIIOPEnabled` system properties to `false` when starting WebLogic Server. For example:

```
-Dweblogic.security.remoteAnonymousRMIT3Enabled=false

-Dweblogic.security.remoteAnonymousRMIIIOPEnabled=false
```

> **Note:**
>
> Although use of these system properties will disable remote anonymous T3 and IIOP access, the security validation infrastructure in WebLogic Server may falsely warn that remote anonymous T3 and IIOP access is enabled.

You cannot disable remote anonymous RMI T3 and IIOP requests if any of the following is used in your WebLogic Server environment:

- T3 or IIOP clients that do not pass credentials (username and password) when creating a JNDI initial context to WebLogic Server
- T3 clients that use the deprecated `weblogic.rmi` APIs
- Environments that configure inter-domain transaction communication with Security Interoperability Mode set to `performance` or `default` (when an administrative channel is not configured). If you want to disable Anonymous RMI T3 and IIOP requests, Oracle recommends that you enable Cross Domain Security for inter-domain communication. See Configuring Secure Inter-Domain and Intra-Domain Transaction Communication in *Developing JTA Applications for Oracle WebLogic Server*.

Disabling remote anonymous requests when they are required in your environment will result in the anonymous requests being rejected and <BEA-000582> and <BEA-002045> errors will be logged in the server log.

Avoid Using These Configurations and Settings in a Locked Down Environment

Oracle strongly recommends that you avoid using configurations and settings that are not secure, such as development mode and demonstration certificates, and that you do not disable default secure settings designed to protect your environment.

Table 3-10 Configurations and Settings that You Must Not Use in a Locked Down Environment

Configuration/Setting	Description	More Information
Do not enable tunneling on channels that are available external to the firewall.	If you allow tunneling, then the external client can send T3/IIOP traffic which can contribute to T3/RMI serialization security vulnerabilities.	• Configure Network Channels and Firewalls to Prevent Access from Non-HTTPS Traffic

Table 3-10 (Cont.) Configurations and Settings that You Must Not Use in a Locked Down Environment

Configuration/Setting	Description	More Information
Do not run WebLogic Server in development mode in a production environment.	Production mode or secured production mode sets the server to run with settings that are more secure and appropriate for a production environment. **Caution:** 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 you configure WebLogic Server in production mode or secured production mode, stack traces are not generated; therefore, you must never run WebLogic Server in development mode in a production environment.	• Configure Production or Secured Production Mode
Do not use MLet MBeans.	MLet (Management applet) 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. Oracle strongly recommends that you do not enable the use of MLet MBeans. If you choose to enable MLet MBeans, then you must 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.	See WLSPolicyFileGroupPrincipalImpl in *Java API Reference for Oracle WebLogic Server*.

Table 3-10   (Cont.) Configurations and Settings that You Must Not Use in a Locked Down Environment

Configuration/Setting	Description	More Information
Do not disable 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` **Note:** If secured production mode is enabled for your domain, then WebLogic Server logs a warning if the `weblogic.security.SSL.enforceConstraints` system property value is set to `false`.	See SSL Certificate Validation in *Administering Security for Oracle WebLogic Server*.
Do not use the demonstration digital certificates in a production environment.	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.	• Configure Keystores in *Oracle WebLogic Remote Console Online Help* • Configuring SSL in *Administering Security for Oracle WebLogic Server*
Do not use SSLv2, SSLv3, TLSv1.0, TLSv1.1 protocol versions.	TLS V1.2 is the default minimum protocol version configured in WebLogic Server. Oracle recommends the use of TLS V1.2 or later in a production environment. WebLogic Server logs a warning if the TLS version is set below 1.2.	See Specifying the SSL Protocol Version in *Administering Security for Oracle WebLogic Server*.

Table 3-10 (Cont.) Configurations and Settings that You Must Not Use in a Locked Down Environment

Configuration/Setting	Description	More Information
Do not enable remote access to the JVM platform MBean server.	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 Monitoring and Management Using JMX Technology in *Java Platform, Standard Edition Monitoring and Management Guide*). 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.	Registering MBeans in the JVM Platform MBean Server in *Developing Manageable Applications Using JMX for Oracle WebLogic Server*.

Table 3-10 (Cont.) Configurations and Settings that You Must Not Use in a Locked Down Environment

Configuration/Setting	Description	More Information
Do not disable host name verification.	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: `-Dweblogic.security.SSL.ignoreHostnameVerification=true` *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. **Note:** If secured production mode is enabled for your domain, then WebLogic Server logs a warning if host name verification is disabled.	Using Host Name Verification in *Administering Security for Oracle WebLogic Server*. To enable host name verification if it is disabled, see Enable Host Name Verification in *Oracle WebLogic Remote Console Online Help* .
Do not enable network classloading.	The `network-class-loading-enabled` element in the central domain configuration file controls whether a server can attempt to load classes from the network. By default, this is set to false, preventing the server from loading classes remotely. If you enable `network-class-loading-enabled`, it exposes WebLogic Server to remote code execution vulnerabilities.	Overview of Domain Configuration Files in *Understanding Domain Configuration for Oracle WebLogic Server*.

Secure Applications

Although much of the responsibility for securing resources in a WebLogic 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 those settings. 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 3-11 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 WebLogic Remote 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, WebLogic Remote Console can be used for subsequent updates. See 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, see 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. See security-constraint in *Developing Web Applications, Servlets, and JSPs for Oracle WebLogic Server*.

Table 3-11 (Cont.) Securing Applications

Security Action	Description
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, see Setting Up a Default Servlet in *Developing Web Applications, Servlets, and JSPs 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 (OWASP) at `https://owasp.org/`). 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`* See Using the Java Security Manager to Protect WebLogic Resources in *Developing Applications with the WebLogic Security Service*. **Note:** When your domain is running in secured production mode, WebLogic Server logs a warning if security manager is not enabled. However, you can specify whether this warning should be logged or not by using the `WarnOnJavaSecurityManager` attribute contained in the `SecureModeMBean`.

Table 3-11 (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. See the following topics on the Open Web Application Security Project (OWASP) website: • *Input Validation Cheat Sheet* at `https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html` • *Cross Site Scripting Prevention Cheat Sheet* at `https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.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*.
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. See Securing a WebSocket Application 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. See Securing a WebSocket Application in *Developing Applications for Oracle WebLogic Server*.