BEA WebLogic Server™

Securing a Production Environment

Version 8.1
Revised: June 28, 2006

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

About This Document

Determining Your Security Needs

Ensuring the Security of Your Production Environment

About This Document

This document highlights essential security measures for you to consider before you deploy WebLogic Server into a production environment. It is organized as follows:

- Chapter 1, "Determining Your Security Needs," provides a list of questions and additional resources for determining the level of security your WebLogic Server deployment needs.
- Chapter 2, "Ensuring the Security of Your Production Environment," provides a list of essential security actions to ensure that your production is secured from intrusions.

Audience

This document focuses on security issues that are directly related to WebLogic Server. It assumes that you have taken additional measures to secure the operating systems, networking facilities, databases, and hardware in your production environment.

The document is written for Application Architects, Application Administrators, and Server Administrators.

In addition to setting security goals and designing the overall security architecture for their organizations, **Application Architects** evaluate WebLogic Server security features and determine how to best implement them. Application Architects have in-depth knowledge of Java programming, Java security, and network security, as well as knowledge of security systems and leading-edge security technologies and tools.

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

Server Administrators work closely with Application Architects to design a security scheme for the server and the applications running on the server, to identify potential security risks, and to propose security configurations that prevent security problems.

Related responsibilities may include maintaining critical production systems, setting up and configuring security realms, implementing authentication and authorization schemes for server and application resources, upgrading security features, and maintaining security provider databases. Server Administrators have in-depth knowledge of the Java security architecture, including Web application and EJB security, Public Key security, and SSL.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation.

How to Print this Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the ProductName documentation CD. You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format.

If you do not have the Adobe Acrobat Reader installed, you can download it for free from the Adobe Web site at `http://www.adobe.com/`.

Contact Us!

Your feedback on BEA documentation is important to us. Send us e-mail at docsupport@bea.com if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the documentation.

In your e-mail message, please indicate the software name and version you are using, as well as the title and document date of your documentation. If you have any questions about this version of BEA WebLogic Server, or if you have problems installing and running BEA WebLogic Server, contact BEA Customer Support through BEA WebSupport at http://www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address

- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and filenames and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*

Convention	Item
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed. *Example*: buildobjclient [-v] [-o name] [-f *file-list*]... [-l *file-list*]...
|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	Indicates one of the following in a command line: • That an argument can be repeated several times in a command line • That the statement omits additional optional arguments • That you can enter additional parameters, values, or other information The ellipsis itself should never be typed. *Example*: buildobjclient [-v] [-o name] [-f *file-list*]... [-l *file-list*]...
. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

Determining Your Security Needs

Before you deploy WebLogic Server and your J2EE applications into a production environment, determine your security needs and make sure that you have taken the appropriate security measures, as described in the following sections:

- "Understand Your Environment" on page 1-1
- "Hire Security Consultants or Use Diagnostic Software" on page 1-2
- "Read Security Publications" on page 1-2
- "Install WebLogic Server in a Secure Manner" on page 1-2

Understand Your Environment

To better understand your security needs, ask yourself the following questions:

- Which resources am I protecting?

  There are many resources in the production environment that can be protected including information in databases accessed by WebLogic Server and the availability, performance, 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 to 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. BEA partners offer services and products that can help you to secure a WebLogic Server production environment. See the BEA Partner's Page at http://www.bea.com/partners.

Read Security Publications

Read about security issues:

- For the latest information about securing Web servers, BEA recommends the "Security Practices & Evaluations" information available from the CERT™ Coordination Center operated by Carnegie Mellon University.
- For BEA security advisories, refer to the BEA Advisories & Notifications page on the dev2dev Web site at http://dev2dev.bea.com/advisories. Here, you can download security-related patches and register to receive notifications of newly available security advisories.

  Report possible security issues in BEA products to `secalert@bea.com`.

Install WebLogic Server in a Secure Manner

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

- Do not install the WebLogic Server sample applications and XML Spy. When installing WebLogic Server, select the Custom option and unclick the Samples option. At the end of the WebLogic Server installation, unclick the Install XML Spy option.
- Delete the following files or directories from the WebLogic Server installation:

  Note: There is always a potential of making mistakes when deleting executables, files, and directories from the WebLogic Server installation. Therefore, BEA recommends testing your changes in a secure, development environment before implementing them in a production environment.

  – Run with the JRE instead of the Java SDK. The Javasoft SDK offers a JRE download and installation. When installing WebLogic Server, use the Configuration Wizard and select the JRE option. This option eliminates the Java compiler and other development tools.

  – When using JRockit, delete the software components of the Java SDK that are not in the JRockit JRE.

  – Delete development tools such as the Configuration Wizard, WebLogic Builder and the jCOM tools if you don’t plan to use them in production.

  – Delete the Pointbase database which is included for evaluation purposes and it is not supported in the production environments.

  – Delete the MedRec sample providers JAR file. This file is installed even if you chose not to install the sample applications.

Ensuring the Security of Your Production Environment

BEA recommends that you implement the following actions to ensure the security of your production environment:

- "Securing the WebLogic Server Host" on page 2-1
- "Securing Network Connections" on page 2-7
- "Securing Your Database" on page 2-10
- "Securing the WebLogic Security Service" on page 2-10
- "Securing Applications" on page 2-17

Securing the WebLogic Server Host

A WebLogic Server production environment is only as secure as the security of the machine on which it is running. Therefore, it is important that you lock down the physical machine, the operating system, and all other software that is installed on the host machine. The following are suggestions for locking down a WebLogic Server host in a production environment. Also check with the manufacturer of the machine and operating system for recommended security measures.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
Physically secure the hardware.	Keep your hardware in a secured area to prevent unauthorized users from tampering with the deployment machine or its network connections.
Secure networking services that the operating system provides.	Have an expert review network services such as the e-mail program or directory service to ensure that a malicious attacker cannot access the operating system or system-level commands. Avoid sharing file systems with other machines in the enterprise network.
Use a file system that can prevent unauthorized access.	Make sure that the file system on each WebLogic Server host can prevent unauthorized access to protected resources. For example, on a Windows computer, use only NTFS.
Limit the number of user accounts on the host machine.	Avoid creating more user accounts than you need on WebLogic Server hosts, and limit the file access privileges granted to each account. Ideally, the host machine would have two user accounts with system administrator privileges and one user with sufficient privileges to run WebLogic Server. 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 user or intruder with read access to files and directories can defeat any security mechanisms you establish with WebLogic Server authentication and authorization schemes.
Create no fewer than two user accounts with system administrator privileges.	One of the system administrator users should be created when the domain is created. You can create the other user(s). Having at least two system administrator user accounts ensures that one user maintains account access in case another user becomes locked out by a dictionary attack.
For your system administrator user accounts, choose names that are not obvious.	For additional security, avoid choosing an obvious name such as “system”, “admin”, or “administrator” for your system administrator user accounts.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
Safeguard passwords.	The passwords for user accounts on production machines should be difficult to guess and should be guarded carefully. Set a policy to expire passwords periodically. Never code passwords in client applications.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
On each host computer, give only one user account access to WebLogic resources.	On each WebLogic Server host computer, use the operating system to establish a special user account (for example, `wls_owner`) specifically to run WebLogic Server. Grant to this operating-system (OS) user account the following privileges: • Access privileges only to the BEA Home directory, the WebLogic Server product directory tree, and your domain directories. The **BEA Home directory** is a repository for common files that are used by multiple BEA products installed on the same machine. The WebLogic Server **product installation directory** contains all the WebLogic Server software components that you choose to install on your system, including program files. A **domain directory** contains the configuration files, security files, log files, J2EE applications, and other J2EE resources for a single WebLogic domain. If you install multiple domains on a WebLogic Server host computer, each domain directory must be protected. By default, the BEA installation program places all BEA files and your domain directories in a single directory tree, whose top directory is named `bea`. All WebLogic Server files are a subdirectory of this directory tree (`bea\weblogic810`), and your domain files are in other subdirectories (`bea\user_projects\domains\`*`domain1`*, `bea\user_projects\domains\`*`domain2`*, ...). You can, however, locate the WebLogic Server product installation directory and your domain directories outside the BEA Home directory. For more information, refer to "Selecting Directories for the WebLogic Platform Installation" in the *Installation Guide*. • Grant read, write, and execute privileges within the BEA Home directory, the WebLogic Server product directory tree, and your domain directories. No other OS user should have read, write, or execute access to BEA files and your domain files. This protection limits the ability of other applications executing on the same machine as WebLogic Server to access BEA 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.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
Run WebLogic Server Windows services under the special OS user account.	On the Windows platform, you can run a WebLogic Server instance as a Windows service. This causes the server instance to start automatically each time you boot the Windows computer. To **set up** a WebLogic Server instance to run as a Windows service, you must log in to the Windows computer with a user account that has privileges to modify the Windows registry. For more information, refer to "Setting Up a WebLogic Server Instance as a Windows Service" in *Configuring and Managing WebLogic Server*. You do not need these administrator-level privileges to **run** a WebLogic Server instance as a Windows service. Instead, the Windows service should run under the special OS user account that you created for running WebLogic Server. To ensure that the WebLogic Server instance runs under the special OS user account, provide the username and password on the Windows service's Properties page. For more information, refer to "Verifying the User Account Under Which the Service Runs" in *Configuring and Managing WebLogic Server*.
To bind to protected ports on UNIX, configure WebLogic Server to switch user IDs or use Network Address Translation (NAT) software.	On UNIX systems, only processes that run under a privileged user account (in most cases, root) can bind to ports lower than 1024. However, long-running processes like WebLogic Server should not run under these privileged accounts. Instead, you can do either of the following: • For each WebLogic Server instance that needs access to privileged ports, configure the server to start under a 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. See "Binding to Protected Ports on UNIX" in the *Administration Console Online Help*. • Start WebLogic Server instances from a non-privileged account and configure your firewall to use Network Address Translation (NAT) software to map protected ports to unprotected ones. BEA does not provide NAT software.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
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 and sample software on a production machine.	Do not install development tools on production machines. Keeping development tools off the production machine reduces the leverage intruders have should they get partial access to a WebLogic Server production machine. Do not install the WebLogic Server sample applications on a production machine. When the BEA installation program asks whether you want a Typical Installation or Custom Installation: 1. Choose Custom Installation. Then click Next. 2. On the Choose Components page, remove the check mark from the Server Examples check box. Then click Next. Complete the remaining pages of the BEA installation program.
Enable security auditing.	If the operating system on which WebLogic Server runs supports security auditing of file and directory access, BEA recommends using audit logging to track any denied directory or file access violations. Administrators should ensure that sufficient disk space is available for the audit log.
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 operation-system service packs and security patches.	Refer to the vendor of your operating system for a list of recommended service packs and security-related patches.

Table 2-1 Securing the WebLogic Server Host

Security Action	Description
Apply the latest BEA service packs and implement the latest security advisories.	If you are responsible for security related issues at your site, register on the BEA Advisories & Notifications page at http://dev2dev.bea.com/advisories to receive notifications of newly available security advisories. Remedies recommended in the security advisories are posted on the Advisories & Notifications page. In addition, you are advised to apply each service pack as it is released. Service packs include a roll-up of all bug fixes for each version of the product, as well as each of the previously released service packs. You can download service packs from http://commerce.bea.com/downloads. Report possible security issues in BEA products to `secalert@bea.com`.
Do not run WebLogic Server in Development mode in a production environment.	Production mode sets the server to run with settings that are more secure and appropriate for a production environment.
Protect Java Transaction API (JTA) transaction logs.	As described in Monitoring Transactions, the transaction log consists of multiple files. You should use operating system-specific methods to protect these files from tampering that might prevent transaction recovery from taking place. For JTA migration, all servers in the same cluster must have permission to write, delete, and read from the transaction log files but access by users from a different domain or cluster should be disallowed.

Securing Network Connections

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

Table 2-2 Securing Network Connections

Security Action	Description
Use hardware and software to create firewalls.	A firewall limits traffic between two networks. Firewalls can be a combination of software and hardware, including routers and dedicated gateway machines. They employ filters that allow or disallow traffic to pass based on the protocol, the service requested, routing information, packet content, and the origin and destination hosts or networks. They can also limit access to authenticated users only. The WebLogic Security Service supports the use of third-party Identity Assertion providers, which perform perimeter-based authentication (Web server, firewall, VPN) and handle multiple security token types/protocols (SOAP, IIOP-CSIv2). For more information, refer to "Perimeter Authentication" in *Introduction to WebLogic Security*. For more information about using firewalls with WebLogic Server, refer to "Security Options for Cluster Architectures" in *Using WebLogic Server Clusters*.
Use WebLogic Server connection filters.	Instead of, or in addition to, using hardware and third-party software to create firewalls, consider using WebLogic Server connection filters to limit network traffic based on protocols, IP addresses, and DNS node names. Connection filters are most appropriate when the machines in a WebLogic Server domain can access each other without going through a firewall. For example, you might use a firewall to limit traffic from outside the network, and then use WebLogic Server connection filters to limit traffic behind the firewall. See “Configuring Connection Filtering” in *Managing WebLogic Security*.

Table 2-2 Securing Network Connections

Security Action	Description
Use a domain-wide Administration Port for administrative traffic.	An Administration Port limits all administrative traffic between server instances in a WebLogic Server domain to a single port. When the server is run without an Adminstrative Port, an application can inadvertently transmit confidential server configuration on the wire in clear-text. Running the server with an Administration Port significantly reduces the chances of this happening. Furthermore, having an Administrative Port configured is helpful should a denial-of-service attack occur because the resources for handling requests for, and the limitations on Administration Port requests are separate from those of the rest of the server. When used in conjunction with a connection filter, you can specify that a WebLogic Server instance accepts administrative requests only from a known set of machines or subnets and only on a single port. Enabling the Administration Port requires clients to interact with the Administration Console using SSL which protects sensitive data from being sniffed on the wire by an attacker and protects against some cross site scripting attacks. You enable the domain-wide Administration Port in the Administration Console on the *DomainName* →Configuration →General tab. See "Enabling the Domain-Wide Administration Port" in the *Administration Console Online Help*.

Table 2-2 Securing Network Connections

Security Action	Description
Enable the Administrative channel	The Administration channel must be enabled to ensure that inter-server administrative communication is secured. Without the Adminstration channel, some key administrative messages will be passed in clear-text allowing the capture, modification, deletion, and replay of messages. See "Administration Port and Administration Channel" in *Configuring and Managing WebLogic Server*.
Secure the embedded LDAP port.	To protect the embedded LDAP port against brute force attacks, close off the embedded LDAP listen port using a connection filter in a single server configuration. While this does not protect the embedded LDAP port in a multiple server configuration, the built-in implementation supports filtering based on the source IP address which should be used to allow access only from servers that are part of the domain. As a result, only the machines in the domain can access the LDAP port. For more information on using connection filters, see "Using Network Connection Filters" in *Programming WebLogic Security*.

Securing Your Database

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

Securing the WebLogic Security Service

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

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Apply the latest BEA service packs and implement the latest security advisories.	If you are responsible for security related issues at your site, register on the BEA Advisories & Notifications page at http://dev2dev.bea.com/advisories to receive notifications of newly available security advisories. Remedies recommended in our security advisories are posted on the Advisories & Notifications page. In addition, you are advised to apply each service pack as it is released. Service packs include a roll-up of all bug fixes for each version of the product, as well as each of the previously released service packs. Download service packs from http://commerce.bea.com/downloads. Report possible security issues in BEA products to `secalert@bea.com`.
Deploy production-ready security providers to the security realm.	The WebLogic Security Service uses a pluggable architecture in which you can deploy multiple security providers, each of which handles a specific aspect of security. By default WebLogic Server includes its own security providers that provide a complete security solution. If you have purchased or written your own security providers: • Make sure that you have deployed and configured them properly. You can verify which security providers are currently deployed in the Administration Console under the Security →Realms →*RealmName* →Providers folder. • Make sure that the realm in which you deployed your security providers is the default (active) realm. You activate a realm in the Administration Console by clicking on the name of the Security realm folder in the left pane and then specifying Domain Wide Security Settings in the right pane. • Refer to "Customizing the Default Security Configuration" in *Managing WebLogic Security*.

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Use SSL, but do not use the demonstration digital certificates in a production environment.	To prevent sensitive data from being compromised, secure data transfers by using the SSL and the HTTPS protocol (HTTP over the Secure Sockets Layer (SSL)) rather than the HTTP protocol. WebLogic Server includes a set of demonstration private keys, digital certificates, and trusted certificate authorities that are for development only. Everyone who downloads WebLogic Server has the private keys for these digital certificates. Do not use the demonstration identity and trust. Refer to "Configuring Keystores and SSL" in the *Administration Console Online Help*.
Enable maximum-strength encryption.	The version of WebLogic Server that you download supports 512-bit keys and 40-bit bulk encryption. If you want to use a version that supports maximum-strength encryption (1024-bit keys with 128-bit bulk encryption), contact your BEA sales representative. Because of export restrictions, this version of WebLogic Server is available only for customers in specific countries.
Make sure that WebLogic Server enforces security constraints on digital certificates.	When communicating via 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` This option could be located in a startup script, or, if you use the Node Manager to start Managed Servers, in the Administration Console on the Servers →*ServerName* →Configuration →Remote Start Options tab. In your development environment, you might have disabled the enforcement of security constraints to work around incompatibilities with demonstration digital certificates that WebLogic Server provided in releases prior to 7.0 Service Pack 2. Make sure you enable this feature in your production environment.

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Verify that host name verification is enabled to avoid man-in-the-middle attacks.	By default, the WebLogic SSL implementation validates that the host to which a connection is made is the intended or authorized party. However, during the implementation of WebLogic Server at your site, you might have disabled host name verification. You can enable host name verification in the Administration Console by going to the Servers →*ServerName* →Configuration →Keystores & SSL tab and then clicking Show next to Advanced Options. Refer to "Using Host Name Verification" in *Managing WebLogic Security*. *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.

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Restrict the size and the time limit of requests on external channels to prevent denial of service attacks.	To prevent some denial of service 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. BEA recommends that you: • set the size limit of requests on internal channels so that a Managed Server is able to accept messages from the Administration Server; • restrict the size and time limits of requests on external channels; • configure internal channels so that they are only accessible internally and not externally. To do the above, configure these settings for the HTTP, T3, and IIOP protocols in the Administration Console under the Servers →*ServerName* →Protocols tab. Refer to the following tasks in the *Administration Console Online Help*: • Configuring the HTTP Protocol • Configuring the T3 Protocol • Enabling and Configuring the IIOP Protocol *Background Information:* A denial of service attack leaves a Web site running but unusable. Hackers deplete or delete one or more critical resources of the Web site. To perpetrate a denial of service 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.
Set the number of sockets allowed to a server.	To prevent some denial of service attacks, limit the number of sockets allowed to a server. This ensures that the number of file descriptors allowed by the operating system limits is not exceeded. You can configure this setting using the MaxOpenSockCount flag. Refer to the following task in the *Administration Console Online Help*: Servers-->Configuration-->Tuning

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Configure user lockouts and login time limits to prevent attacks on user accounts.	By default, the WebLogic Security Service provides maximum security against dictionary 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. You verify or change these settings in the Administration Console on the Security →Realms →*RealmName* →User Lockouts tab. Refer to "Protecting User Accounts" in the *Administration Console Online Help*. *Background Information:* In a dictionary 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 attacks.
If you use multiple Authentication providers, set the JAAS control flag.	If a security realm has multiple Authentication providers configured, configure the order and precedence of **each** provider by setting the JAAS control flags. You set the JAAS control flag in the Administration Console on the Security →Realms →*RealmName* →Providers →Authentication → *AuthenticatorName* →General tab. Refer to "Setting the JAAS Control Flag" in the *Administration Console Online Help*.

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Enable security auditing.	Auditing is the process of recording key security events in your WebLogic Server environment. When the Auditing provider that the WebLogic Security Service provides is enabled, it logs events in `DomainName\DefaultAuditRecorder.log` You enable auditing of administration changes in the Administration Console on the *DomainName* →Configuration →General tab. See “Configuration Auditing” in the *Administration Console Online Help*. You enable an Auditing provider in the Administration Console on the Security →Realms →*RealmName* →Providers →Auditing page. Refer to "Configuring a WebLogic Auditing Provider" in the *Administration Console Online Help*. **Note:** Using an Auditing provider might adversely affect the performance of WebLogic Server even if only a few events are logged. Review the auditing records periodically to detect security breaches and attempted breaches. Noting repeated failed logon attempts or a surprising pattern of security events can prevent serious problems.
Ensure that you have correctly assigned users and groups to the default WebLogic Server security roles.	By default, all WebLogic resources are protected by security policies that are based on a default set of security roles. Make sure you have assigned the desired set of users and groups to these default security roles. Refer to "Security Roles" in the *Securing WebLogic Resources* guide.

Table 2-3 Securing the WebLogic Security Service

Security Action	Description
Create no fewer than two user accounts with system administrator privileges.	One of the system administrator users should be created when the domain is created. Create other user(s) and assign them the Admin security role. When creating system administrator users give them unique names that cannot be easily guessed. Having at least two system administrator user accounts helps to ensure that one user maintains account access in case another user becomes locked out by a dictionary/brute force attack.
Consider preventing WebLogic Server from sending its name and version number in HTTP responses.	By default, when an instance of WebLogic Server responds to an HTTP request, its HTTP response header includes the server's name and WebLogic Server version number. This poses a potential security risk if an attacker knows about a vulnerability in the specific version of WebLogic Server. To prevent a WebLogic Server instance from sending its name and version number, disable the Send Server Header attribute in the Administration Console. The attribute is located on the Server → *ServerName* →Configuration →Protocols →HTTP tab under the Advanced Options section.

Securing Applications

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

Table 2-4 Securing Applications

Security Action	Description
Determine which technique secures your Web applications and EJBs.	By default, each Web application and EJB uses deployment descriptors (XML files) to declare its secured resources and the security roles that can access the secured resources. Instead of declaring security in Web application and EJB deployment descriptors, you can use the Administration Console to set security policies that secure access to Web applications and EJBs. This technique provides a single, centralized location from which to manage security for all Web applications and EJBs. You can combine these two techniques and configure WebLogic Server to copy security configurations from existing deployment descriptors upon the initial deployment of a URL (Web) or EJB resource. Once these security configurations are copied, the Administration Console can be used for subsequent updates. Refer to "Security Roles" in *Securing WebLogic Resources*.
Use JSP comment tags instead of HTML comment tags.	Comments in JSP files that are not meant for the end user should use the JSP syntax of `<%/* ... */%>` instead of the HTML syntax `<!-- ... -->`. The JSP comments are deleted when the JSP is compiled and therefore cannot be viewed.
Do not install uncompiled JSPs and other source code on the production machine.	Always keep source code off of the production machine. Getting access to your source code allows an intruder to find security holes. Consider precompiling JSPs and installing only the compiled JSPs on the production machine. For information about precompiling JSPs, refer to "Precompiling JSPs" in *Programming WebLogic JSP*.
Configure your applications to use SSL.	Set the `transport-guarantee` to `CONFIDENTIAL` in the `user-data-constraint` element of the `web.xml` file. Refer to "security-constraint" in *Developing Web Applications for WebLogic Server*.

Table 2-4 Securing Applications

Security Action	Description
Use SSL to guarantee the integrity of transactional data	As described in Writing Applications that Use SSL in *Programming WebLogic Security*, BEA WebLogic Server provides Secure Sockets Layer (SSL) support for encrypting data transmitted between WebLogic Server clients and servers, Java clients, Web browsers, and other servers. For applications, the use of SSL is important to guarantee the integrity of transactional data transmitted by RMI calls. Applications can make remote calls using RMI to invoke EJBs, remote objects such as JDBC resources, and so forth. When these RMI invocations are made under the scope of a transaction, the transaction context propagates with the RMI call. Therefore, applications that make remote calls under the scope of a transaction should use SSL so that the transaction context, application data, and security context are not subject to tampering that could affect data integrity.
Do not use the `Servlet` servlet.	BEA does not recommend using the `Servlet` servlet in a production environment. Instead, map servlets to URIs explicitly. Remove all existing mappings between WebLogic servlets and the `Servlet` servlet from all Web applications before using the applications in a production environment. For information on mapping servlets, refer to "Configuring Servlets" in *Developing Web Applications for WebLogic Server*.
Do not leave `FileServlet` as the default servlet in a production environment.	BEA does not recommend using the `FileServlet` servlet as the default servlet a production environment. For information on setting up a default servlet, refer to "Setting Up a Default Servlet" in *Developing Web Applications for WebLogic Server*.
Verify all WebLogic security policies.	In WebLogic Server 7.0, security policies replace ACLs and answer the question “who has access” to a WebLogic resource. Make sure that you have not removed security policies from WebLogic resources, and make sure that your security role assignments provide users the kind of access that you intend. Refer to Securing WebLogic Resources.

Table 2-4 Securing Applications

Security Action	Description
Examine applications for security	There are instances where an application can lead to a security vulnerability. Many of these instances are defined by third-party organizations such as Open Web Application Security project (see http://www.owasp.org/documentation/topten for a list of common problems). Of particular concern is code that uses Java native interface (JNI) because Java positions native code outside of the scope of Java security. If Java native code behaves errantly, it is only constrained by the operating system. That is, the Java native code can do anything WebLogic Server itself can do. This vulnerability is further complicated by the fact that buffer overflow errors are common in native code and can introduce 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 unknown classes are being run, the Java security manager may be useful. To enable the Java security manager for a server instance, use the following Java options when starting the server: `-Djava.security.manager` `-Djava.security.policy[=]=`*`filename`* Refer to "Using the Java Security Manager to Protect WebLogic Resources" in *Programming WebLogic Security*.
Replace HTML special characters when servlets or JSPs return user-supplied data.	The ability to return user-supplied data can present a security vulnerability called **cross-site scripting**, which can be exploited to steal a user's security authorization. For a detailed description of cross-site scripting, refer to "Understanding Malicious Content Mitigation for Web Developers" (a CERT security advisory) at http://www.cert.org/tech_tips/malicious_code_mitigation.html. To remove the security vulnerability, before you return data that a user has supplied, scan the data for HTML special characters. If you find any such characters, replace them with their HTML entity or character reference. Replacing the characters prevents the browser from executing the user-supplied data as HTML. See "Securing User-Supplied Data in JSPs" in *Programming WebLogic JSP* and "Securing Client Input in Servlets" in *Programming WebLogic HTTP Servlets*.