BEA WebLogic Portal™

Security

Version 8.1 Service Pack 5
Document Revised: June 2006

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

WebLogic Portal Security

WebLogic Portal Security

Overview

This guide includes guidelines for securing your portal applications and application resources.

BEA WebLogic Portal® security leverages BEA WebLogic Server security features. The WebLogic Server, in turn, supports and enhances J2EE security services. Like other J2EE components, the security services are based on standardized, modular components. WebLogic Server implements these Java security service methods according to the standard, and adds extensions that handle many details of application behavior automatically, without requiring additional programming. For more information about WebLogic Server security, see the WebLogic Server Security documentation.

This chapter includes the following sections:

Authentication

Authentication is the verification of identity, answering the question: Is the user who they say they are?

Authentication is typically performed using a login process, during which the user supplies a credential, such as a username and password combination. Once the user is authenticated, a set of identities (such as the username and the user's group membership) is associated with the user. These identities are also referred to as *principals*.

Authentication Providers

Authentication providers verify the identity of users. Authentication providers also remember, transport, and make available that identity information to various components of a system through *subjects* (containers for authentication information) when needed. During the authentication process, a principal validation provider provides additional security protections for the principals contained in the subject by signing and verifying the authenticity of those principals.

WebLogic Server supplies its own authentication providers, which can access user and group data in its embedded LDAP server, external LDAP stores, and external Relational Database Management Systems (RDBMSs). WebLogic Portal and WebLogic Server use the embedded LDAP server as the default authentication provider, but WebLogic Portal also provides its own RDBMS Authenticator.

WebLogic Portal uses WebLogic Server for authentication, whether you are using one or more WebLogic authentication providers, one or more custom authentication providers configured for use with WebLogic Server, or a combination of WebLogic and custom providers.

Identity Assertion Providers and Single Sign-On

An identity assertion provider is a specific type of authentication provider that allows users or system processes to assert their identity using *tokens* (representations of security-related information). Identity assertion providers enable perimeter authentication and support *single sign-on* (SSO). For example, an identity assertion provider can generate a token from a digital certificate, and that token can be passed around the system so that users are not asked to sign on more than once.

You can use an identity assertion provider in place of an authentication provider if you create a LoginModule for the identity assertion provider, or in addition to an authentication provider if you want to use the authentication provider LoginModule.

Web applications often consist of many different components, each of which may have its own authentication scheme or user registry. SSO enables users of these applications to authenticate only once to obtain access to all components of the application. Once users are authenticated in one site that participates in a SSO configuration, they are automatically logged into the other sites in the SSO configuration.

The WSRP Identity Asserter provides this functionality.

Authorization

Authorization is the control of access to resources, answering the question: Does the user have access to this protected resource? Interactions between users and resources are controlled based on user identity or other information.

In WebLogic Portal (as in WebLogic Server), roles control access to portal resources, J2EE resources, and administrative tools, so users can access only the resources and tools that their assigned roles allow. WebLogic Portal uses WebLogic Server roles to enable you to dynamically match users to roles at login.

Authorization Providers

An authorization provider controls the interactions between users and resources to ensure confidentiality. Like a LoginModule for an authentication provider, an Access Decision is the component of an authorization provider that determines if access to a resource is allowed. Specifically, an Access Decision determines whether a subject has permission to perform the specified action on a WebLogic resource. A runtime call to the isAccessAllowed method makes this determination, based on the subject's security roles.

It returns one of the following values:

`PERMIT`—Indicates that the requested access is permitted

`DENY`—Indicates that the requested access is explicitly denied

`ABSTAIN`—Indicates that no explicit decision was rendered

Role Mapping Providers

Role mapping is the process by which principals (users or groups) are dynamically mapped to security roles at runtime. A role mapping provider determines which security roles apply to the subject when the subject is attempting to perform an operation on a resource. Because this

operation usually involves gaining access to the resource, role mapping providers are typically used with authorization providers.

The default role mapping provider is the WebLogic Role Mapper. It stores role policies in the embedded LDAP server.

Roles and Role Policies

A security role is a privilege granted to users or groups based on specific conditions. Roles are used to determine whether to grant or deny access to resources, and to determine which capabilities on those resources are available to the user. Granting a role to a user or group confers the defined access privileges to that user or group, as long as the user or group is granted the role. Any number of users or groups can be granted a single role.

Roles are computed and granted to users or groups dynamically, based on role policies, which consist of a role name and a role definition. Role policies are dynamic, and can be based on username, group membership, user profile property values, session and request attributes, and date and time functions.

Roles can be scoped to specific WebLogic resources within a single application in a WebLogic Server domain (unlike groups, which are always scoped to an entire WebLogic Server domain).

For additional information, see Securing WebLogic Resources in the see the WebLogic Server Security documentation.

Security Policies

Security policies answer the question: Who has access to a WebLogic resource? A security policy is created when you define an association between a WebLogic resource and one or more users, groups, or roles. Hence, a role policy defines a role and a security policy defines an authorization constraint associated with that role.

BEA recommends basing security policies on roles rather than users or groups. Basing security policies on roles enables you to manage access based on a role that a user or group is granted, which is a more flexible method of management. Security policies are kept in the authorization provider's store, which is the embedded LDAP server by default.

If a security policy is based on a user or group, the user or group must be defined in the user store for the authentication provider that is configured in the default security realm. If a security policy is based on a role, the role must be defined in the store for the role mapping provider that is configured in the default security realm.

For additional information, see Securing WebLogic Resources in the see the WebLogic Server Security documentation.

Deployment Descriptors

The J2EE platform defines a means of establishing security contracts in a document known as the deployment descriptor. Deployment descriptors restrict access to resources that can be accessed directly using a URL. You must use deployment descriptors to secure portlet resources including JSPs and page flows, which could otherwise be accessed directly by anyone that knows the URL to those resources. Roles defined in deployment descriptors are based on users and groups.

For information on securing portal application resources using deployment descriptors, see "Preventing Direct Access to Portlet Resources" on page 1-8.

Portal application resources can also be protected based on runtime constraints, such as the time of day and user profile property values. For information on setting up these runtime constraints using role-based security policies, see the WebLogic Portal online help.

Auditing

Auditing is the process whereby information about operating requests and the outcomes of those requests is collected, stored, and distributed in order to prevent repudiation. In other words, auditing provides an electronic trail of computer activity. In the WebLogic Server security architecture, an Auditing provider is used to provide auditing services.

If configured, the WebLogic Security Framework calls an Auditing provider before and after the performance of security operations (such as authentication or authorization). The decision to audit a particular event is made by the Auditing provider itself and can be based on specific audit criteria and/or severity levels.

You can use either the WebLogic Auditing provider or a custom Auditing provider in a security realm. Although an Auditing provider is configured per security realm, each server writes auditing data to its own log file in the server directory. By default, all auditing information recorded by the WebLogic Auditing provider is saved in the following file:

```
WL_HOME\yourdomain\yourserver\DefaultAuditRecorder.log.
```

By writing a custom Auditing provider, however, you can send the records containing audit information to any one of various output repositories, such as an LDAP server, database, or a simple file.

To configure an Auditing provider and enable auditing, see Configuring a WebLogic Auditing Provider in the WebLogic Server Security documentation.

WebLogic Portal-Specific Security Extensions

This section provides a high level overview of the WebLogic Portal-specific extensions to role-based access control. You can control access to portal resources for two categories of users:

- Portal visitors, whose access to portals and portal resources you control using visitor entitlements. Visitor access is determined based on visitor entitlement roles.
- Portal administrators, whose capabilities to manage portal resources you control using delegated administration. Administrative access is determined based on delegated administration roles.

Visitor Entitlements

Use visitor entitlements to determine who may access the resources in a portal application and what they may do with those resources. Access is based on the role assigned to a portal visitor, allowing for flexible management of portal resources.

For example, if you have an Employee Review portlet that only managers should be able to access, you can create a Managers visitor entitlement role, and set entitlements so that only users who are assigned that role can view the portlet.

A portal visitor is assigned a role based on their username, group membership, user profile property values, session and request attributes, and date and time functions. For example, the Gold Member role could be assigned to visitors who are part of the frequent flyer program and have flown more than 50,000 miles in the previous year. Roles are assigned to visitors dynamically when they log in to the site. As this example highlights, visitor entitlements also enable personalization.

If no visitor entitlement roles exist, the default behavior is to allow access to the portal and portal resources to all visitors.

Delegated Administration

Delegated administration provides secure administrative access to the WebLogic Portal Administration Console. Using the WebLogic Portal Administration Console, administrators can create and manage portals, desktops, shells, books, pages, layouts, look and feels, and portlets. A delegated administration role is a classification of a portal administrator based on their username, group membership, user profile property values, session and request attributes, and date and time functions.

In your organization, you might want different administrators to have different access privileges to various administration tasks and resources. You can use delegated administration to propagate WebLogic Portal Administration Console access privileges within a hierarchy of roles.

For example, a system administrator might have access to every feature in the WebLogic Portal Administration Console. The system administrator might then create a portal administrator role for administrators who are allowed to manage instances of portal resources in specific desktop views of your portal. The system administrator might also create a library administrator role for administrators who are allowed to manage your portal resource library.

Delegated administration roles are mapped to administrative functions on portal resources using security policies. Given the appropriate privileges, administrators can delegate both the privilege to administer a given resource capability and the privilege for the delegatee to delegate further.

Protecting User Accounts

An important aspect of authentication is how to handle login failures. You can configure the number of login attempts that are allowed, as well as how to terminate the login process after a specified number of unsuccessful user authentication attempts has been made.

Authentication provider properties, including the User Lockout Manager, are described in View Security Provider Properties.

User lockout attributes, including the lockout threshold and the lockout duration, are controlled using the WebLogic Server Administration Console, as described in Protecting User Accounts in the WebLogic Server document Managing WebLogic Security.

You can also unlock a locked account, as described in Unlocking a User Account in the WebLogic Server document Managing WebLogic Security.

Getting Started with Portal Administration contains a high-level description of portal administration and how to log in to a portal as a default user. Default users are listed in the Securing Portal Applications Overview topic.

For additional information on default users, groups, and roles, see Users, Groups, and Roles Preconfigured in a Platform Domain in the WebLogic Platform document Security in WebLogic Platform 8.1. You must manage and protect the passwords for these accounts.

Note: BEA recommends that you change all administrator passwords from their default values, especially in a production environment. For additional information on managing passwords for user and administrator accounts, see Ensuring the Security of Your Production Environment in the WebLogic Server document Securing a Production Environment.

Securing a Production Environment: Guidance for Administrators

This section highlights considerations you must be aware of to ensure secure operation of your production WebLogic Portal environment.

Administrators must adhere to the guidance in Securing a Production Environment in the WebLogic Server Security documentation to ensure that all aspects of the WebLogic Server environment, including hosts, network connections, databases, and applications have been properly secured.

Additional guidance for ensuring a secure WebLogic Platform environment is provided in Managing WebLogic Platform Security.

Preventing Direct Access to Portlet Resources

You can control visitor access to portal resources using visitor entitlements in the WebLogic Portal Administration Console. However, you must also use deployment descriptors to secure the JSPs and page flows contained in a portlet; otherwise a malicious user can access those resources directly if they know the correct URL.

You must use J2EE security to prevent direct access to JSPs and page flows; otherwise, a user can access those resources directly by entering the correct URL.

Note: Descriptor security is only intended to prevent direct access to the JSP or page flow using a URL; it is not used when a portal renders a portlet.

Scoped roles are defined in their respective deployment descriptors.

- Enterprise-application-scoped roles are defined in `application.xml` and `weblogic-application.xml`
- Web-application-scoped roles which apply to resources within a project are defined in `web.xml` and `weblogic.xml`
- EJB-scoped roles which apply only to resources within an EJB are defined in `ejb-jar.xml` and `weblogic-ejb-jar.xml`

An example URL to a JSP is:

```
http://emp_app/employmentPortal/portlets/hr/vpSalaries.jsp
```

To prevent direct access to portal resources, add a security entry in your portal Web project's `/WEB-INF/web.xml` file. For example:

```
<!-- Use declarative security to block direct address to portlets -->
    <security-constraint>
        <display-name>Default Portlet Security Constraints</display-name>
        <web-resource-collection>
            <web-resource-name>Portlet Directory</web-resource-name>
            <url-pattern>/portlets/*</url-pattern>
            <http-method>GET</http-method>
            <http-method>POST</http-method>
        </web-resource-collection>
        <auth-constraint>
            <role-name>Admin</role-name>
        </auth-constraint>
        <user-data-constraint>
            <transport-guarantee>NONE</transport-guarantee>
        </user-data-constraint>
    </security-constraint>
```

This security entry in the `web.xml` file protects all files in the portal web project's `/portlet` directory and its subdirectories from being directly accessed by a request URL.

These protected resources are still displayed in entitled portlets, but only for users entitled to access those portlets.

Warning: A <url-pattern> of `/portlets/*.jsp` is not legal syntax and does not protect subdirectories.

Resources such as images, which do not require security restrictions, must be stored in unsecured directories outside of the `/portlets` directory.

Avoiding Direct Editing of Portal Resource Definition Labels

Although the capability to edit a definition label exists, BEA strongly discourages you from doing so, once you have created the portal resource using WebLogic Workshop. Modifying the

definition label could have unintended implications, including exposing a protected resource, or breaking WSRP (which uses the definition label as the portlet handle).