CHAPTER 6

Running in a Secure Environment

This chapter describes important information about running GlassFish Server in a secure environment.

This chapter assumes that you are familiar with security features such as authentication, authorization, and certificates. If you are not, see Chapter 1, "Administering System Security."

Instructions for accomplishing the tasks specific to GlassFish Server by using the Administration Console are contained in the Administration Console online help.

The chapter describes the following topics:

- "Determining Your Security Needs" on page 97
- "Installing GlassFish Server in a Secure Environment" on page 98
- "Remove Unused Components" on page 99
- "Run on the Web Profile if Possible" on page 102
- "Securing the GlassFish Server Host" on page 102
- "Securing GlassFish Server" on page 106
- "Securing Applications" on page 109

Determining Your Security Needs

Before you deploy GlassFish Server and your Java EE applications into a production environment, determine your security needs and make sure that you take the appropriate security measures, as described in the following sections:

- "Understand Your Environment" on page 98
- "Read Security Publications" on page 98

Understand Your 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 GlassFish 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 GlassFish Server environment? Should the system administrators have access to all GlassFish Server 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.

Read Security Publications

Read about security issues:

- For the latest information about securing Web servers, Oracle recommends the "Security Practices & Evaluations" information available from the CERT Coordination Center operated by Carnegie Mellon University at http://www.cert.org (`http://www.cert.org/`).

Installing GlassFish Server in a Secure Environment

This section describes recommendations for installing GlassFish Server in a secure environment. The following topic is described:

- "Enable the Secure Administration Feature" on page 99

Enable the Secure Administration Feature

The secure administration feature allows an administrator to secure all administrative communication between the domain administration server (DAS), any remote instances, and administration clients such as the asadmin utility, the administration console, and REST clients. In addition, secure administration helps to prevent DAS-to-DAS and instance-to-instance traffic, and carefully restricts administration-client-to-instance traffic.

When you install GlassFish Server or create a new domain, secure admin is disabled by default. GlassFish Server does not encrypt administrative communication among the system components and does not accept administrative connections from remote hosts. Imposing a heightened level of security is optional.

See Chapter 5, "Managing Administrative Security," for information on enabling the secure administration feature.

Remove Unused Components

Minimize the GlassFish Server installation by removing components that you are not using and do not intend to use.

The Update Tool is a standalone graphical tool bundled withGlassFish Server that you can use to find, install, and remove updates and add-ons on a deployed server instance.

The pkg command is the command-line equivalent to Update Tool. Most of the tasks that can be performed with the graphical Update Tool can be performed from a command line using the pkg tool.

To update or remove installed add-on components, use one of the following commands:

- `install-dir/bin/updatetool`, which starts the Update Tool graphical utility.
- `install-dir/bin/pkg`, a command-line version of the Update Tool.

Removing Installed Components

This section describes how to use the pkg utility to remove an installed component. You can also use the Update Tool to perform this task.

▼ Procedure To Remove an Installed Component

1 **Stop GlassFish Server.**

See "To Stop a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

2 **To ensure that the pkg command can locate the application image, change to the base installation directory for GlassFish Server.**

```
cd install-dir
```

3 **Obtain a list of all your installed components. (The following list is for example purposes only and might not match your installed components.)**

```
install-dir/bin/pkg list
```

```
NAME (PUBLISHER)                              VERSION         STATE      UFIX
felix                                         3.0.7-0         installed  ----
glassfish-appclient                           3.1-39          installed  ----
glassfish-bundled-jdk (release.release.sun.com) 1.6.0.23-5.1    installed  ----
glassfish-cluster                             3.1-39          installed  ----
glassfish-cmp                                 3.1-39          installed  ----
glassfish-common                              3.1-39          installed  ----
glassfish-common-full                         3.1-39          installed  ----
glassfish-corba                               3.1.0-23        installed  ----
glassfish-corba-base                          3.1.0-23        installed  ----
glassfish-ejb                                 3.1-39          installed  ----
glassfish-ejb-lite                            3.1-39          installed  ----
glassfish-full-incorporation                  3.1-39          installed  ----
glassfish-full-profile                        3.1-39          installed  ----
glassfish-grizzly                             1.9.28-1        installed  ----
glassfish-grizzly-full                        1.9.28-1        installed  ----
glassfish-gui                                 3.1-39          installed  ----
glassfish-ha                                  3.1-39          installed  ----
glassfish-hk2                                 3.1-39          installed  ----
glassfish-javahelp                            2.0.2-1         installed  ----
glassfish-jca                                 3.1-39          installed  ----
glassfish-jcdi                                3.1-39          installed  ----
glassfish-jdbc                                3.1-39          installed  ----
glassfish-jms                                 3.1-39          installed  ----
glassfish-jpa                                 3.1-39          installed  ----
glassfish-jsf                                 2.1.0-10        installed  ----
glassfish-jta                                 3.1-39          installed  ----
glassfish-jts                                 3.1-39          installed  ----
glassfish-management                          3.1-39          installed  ----
glassfish-nucleus                             3.1-39          installed  ----
glassfish-registration                        3.1-39          installed  ----
glassfish-upgrade                             3.1-39          installed  ----
glassfish-web                                 3.1-39          installed  ----
glassfish-web-incorporation                   3.1-39          installed  ----
glassfish-web-profile                         3.1-39          installed  ----
javadb-client                                 10.6.2.1-1      installed  ----
javadb-common                                 10.6.2.1-1      installed  ----
javadb-core                                   10.6.2.1-1      installed  ----
javaee-firstcup-tutorial                      2.0.2-6         installed  ----
javaee-javadocs                               3.1-39          installed  ----
javaee-samples-build                          1.0-4           installed  ----
javaee-samples-full                           1.0-4           installed  ----
javaee-samples-web                            1.0-4           installed  ----
javaee-sdk-full-profile                       3.1-39          installed  ----
javaee-tutorial                               6.0.1-10        installed  u---
jersey                                        1.5-1.0         installed  ----
metro                                         2.1-25          installed  ----
mq-bin-exe                                    4.5-26.1        installed  ----
```

```
mq-bin-sh                                     4.5-26.1        installed  ----
mq-config-gf                                  4.5-26.1        installed  ----
mq-core                                       4.5-26.1        installed  ----
mq-locale                                     4.5-26.1        installed  ----
mq-server                                     4.5-26.1        installed  ----
pkg (dev.glassfish.org)                       1.122.2-50.2809 installed  ----
pkg-java                                      1.122-50.2809   installed  ----
python2.4-minimal (dev.glassfish.org)         2.4.4.0-50.2809 installed  ----
sdk-branding-full                             3.1-39          installed  ----
shoal                                         1.5.28-0        installed  ----
updatetool (dev.glassfish.org)                2.3.3-50.2809   installed  ----
wxpython2.8-minimal (dev.glassfish.org)       2.8.10.1-50.2809 installed  ----
```

4 **Uninstall the component that you want to remove from your system.**

```
pkg uninstall package-name
```

For example:

```
pkg uninstall metro
```

5 **Start GlassFish Server.**

See "To Start a Domain" in *Oracle GlassFish Server 3.1 Administration Guide*.

Remove Services You Are Not Using

Consider removing services that you are not using. For example, if applications are not using messaging, then consider removing the JMS from the server. Also consider removing EJB Container, JCA, and so forth.

Note – There is always a potential of making mistakes when deleting components from the GlassFish Server installation. Therefore, Oracle recommends testing your changes in a secure development environment before implementing them in a production environment.

The Updatetool and the Administration Console both provide descriptions of each installed component. In addition, the Updatetool also describes dependencies. You can use this information to decide whether you need to keep these components installed.

Before you remove a component, use the `asadmin list-<component>-resources` subcommand or the Administration Console to make sure that resources of a given type, for example JMS, are not in use. For example, you might use the `asadmin list-jms-resources` subcommand to make sure that JMS resources are not currently in use:

```
D:\glassfish3\glassfish\bin>asadmin list-jms-resources

Nothing to list

Command list-jms-resources executed successfully.
```

Run on the Web Profile if Possible

If your applications can run on the Web Profile, use that instead of the Full Platform.

Java EE 6 introduced the concept of profiles. A profile is a collection of Java EE technologies and APIs that address specific developer communities and application types.

The following profiles are implemented through the distributions of GlassFish Server:

- Full Platform –The full Java EE platform is designed for developers who require the full set of Java EE APIs for enterprise application development, and is installed when you install GlassFish Server. This profile is also installed as part of the Java EE 6 SDK installation.
- Web Profile –This profile contains Web technologies that are a subset of the full Java platform, and is designed for developers who do not require the full set of Java EE APIs. This profile is also installed with Java EE 6 Web Profile SDK.

For the list of APIs in each profile, see "Java EE 6 Standards" in *Oracle GlassFish Server 3.1 Release Notes*.

Securing the GlassFish Server Host

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

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

Note – The domain and server configuration files should be accessible only by the operating system users who configure or execute GlassFish Server.

TABLE 6–1 Securing the GlassFish Server Host

Security Action	Description
Physically secure the hardware.	Keep your hardware in a secured area to prevent unauthorized operating system users from tampering with the deployment machine or its network connections.
Log out of the Administration Console before navigating to a non-secure site.	If you are logged on to the Administration Console, be sure to log out completely before browsing to an unknown or non-secure Web site.

TABLE 6–1 Securing the GlassFish Server Host *(Continued)*

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.
Use a file system that can prevent unauthorized access.	Make sure that the file system on each GlassFish Serverhost can prevent unauthorized access to protected resources. For example, on a Windows computer, use only NTFS.
Set file access permissions for data stored on disk.	Set operating system file access permissions to restrict access to data stored on disk. This data includes, but is not limited to, the following: The database files. GlassFish Server includes an implementation of Java DB (formerly known as Derby), however, you can use any JDBC-compliant database. The directory and filename location of a private keystore, such as keystore.jks The directory and filename location of a Root Certificate Authority (CA) keystore, such as cacerts.jks. For example, operating systems provide utilities such as umask and chmod to set the file access permissions. At a minimum, consider using "umask 066", which denies read and write permission to Group and Others.

TABLE 6–1 Securing the GlassFish Server Host *(Continued)*

Security Action	Description
Limit the number of user accounts on the host machine.	Avoid creating more user accounts than you need on 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 GlassFish Server. Having two system administrator users provides a back up at all times. The GlassFish Server user should be a restricted user, not a system administrator user. One of the system administrator users can always create a new GlassFish Server user if needed. Important: Domain and server configuration files should be accessible only by the operating system users who configure or execute GlassFish Server. Review active user accounts regularly and when personnel leave. Background Information: 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 authentication and authorization schemes.
For your system administrator user accounts, choose names that are not obvious.	For additional security, avoid choosing an obvious name such as "system," "admin," or "administrator" for your system administrator user accounts.
Safeguard passwords.	The passwords for user accounts on production machines should be difficult to guess and should be guarded carefully. Set a policy to expire passwords periodically. Never code passwords in client applications. Do not deploy an application that can be accessed with the default username admin and no password.

TABLE 6–1 Securing the GlassFish Server Host *(Continued)*

Security Action	Description
Safeguard password files	The `-passwordfile` option of the `asadmin` command specifies the name of a file that contains password entries in a specific format. These password entries are stored in clear text in the password file, and rely on file system mechanisms for protection. To provide additional security, create a password alias.
Use a password alias	A password alias stores a password in encrypted form in the domain keystore, providing a clear-text alias name to use instead of the password. To provide additional security, use the `create-password-alias` subcommand to create an alias for the password. The password for which the alias is created is stored in an encrypted form. Then, specify the alias in the entry for the password in the password file as follows: In password files and the domain configuration file, use the form ${alias=alias-name} to refer to the encrypted password.
Do not run GlassFish Server as root	GlassFish Servershould run only as an unprivileged user, never as root. The directory structure in which GlassFish Server is located, including all files, should be protected from access by unprivileged users. Taking these steps helps ensure that unprivileged users cannot insert code that can potentially be executed by GlassFish Server server.
Consider use PAM Realm	The use of a PAM Realm requires GlassFish Server to run as an account that has read-access to a shadow password file or the equivalent, and therefore may not be suitable in your environment.
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.

TABLE 6–1 Securing the GlassFish Server Host *(Continued)*

Security Action	Description
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 production machine.
Enable security auditing.	If the operating system on which GlassFish Server runs supports security auditing of file and directory access, Oracle recommends using audit logging to track any denied directory or file access violations. Administrators should ensure that sufficient disk space is available for the audit log.
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.
Apply the latest maintenance packs and critical patch updates.	Refer to the vendor of your operating system for a list of maintenance packs and critical patch updates.

Securing GlassFish Server

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

TABLE 6–2 Securing GlassFish Server

Security Action	Description
Enable Secure Admin.	The secure administration feature allows an administrator to secure all administrative communication between the domain administration server (DAS), any remote instances, and administration clients such as the `asadmin` utility, the administration console, and REST clients. In addition, secure administration helps to prevent DAS-to-DAS and instance-to-instance traffic, and carefully restricts administration-client-to-instance traffic. The secure administration feature provides a secure environment, in which you can be confident that rogue users or processes cannot intercept or corrupt administration traffic or impersonate legitimate GlassFish Server components. See Chapter 5, "Managing Administrative Security"
Protect the `.asadminpass` file	If you create a domain with the `--savelogin` option, `create-domain` saves the administration user name and password in the `.asadminpass` file in the user's home directory. Make sure that this file remains protected. Information stored in this file will be used by `asadmin` commands to manage this domain.
Deploy production-ready security providers to the security realm.	Java Authorization Contract for Containers (JACC) is the part of the Java EE specification that defines an interface for pluggable authorization providers. This enables you to set up third-party plug-in modules to perform authorization. By default, the GlassFish Server provides a simple, file-based authorization engine that complies with the JACC specification. You can also specify additional third-party JACC providers. If you have purchased or written your own security providers, make sure that you have deployed and configured them properly.

TABLE 6–2 Securing GlassFish Server *(Continued)*

Security Action	Description
Use SSL, but do not use the self-signed certificates in a production environment.	To prevent sensitive data from being compromised, secure data transfers by using HTTPS. By default, GlassFish Server uses self-signed certificates. The self-signed certificates that GlassFish Server uses might not be trusted by clients by default because a certificate authority does not vouch for the authenticity of the certificate. You can instead use your own certificates, as described in "Using Your Own Certificates" on page 90.
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, restrict the size of a message as well as the maximum time it takes a message to arrive. The default setting for maximum post size is 2097152 bytes and 900 seconds for the request timeout.
Enable authentication and authorization auditing.	Auditing is the process of recording key security events in your GlassFish Server environment. You use audit modules to develop an audit trail of all authentication and authorization decisions. To enable audit logging, two steps are required: 1. On the Security page, select the Audit Logging Enabled checkbox to enable audit logging. 2. Set the `auditOn` property for the active audit module to true. 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.
Set logging for security and SSL messages.	Consider setting module log levels for table.javax.enterprise.system.ssl.security and javax.enterprise.system.core.security. You can set a level from Severe to Finest (the default is Info), but be aware that the finer logging levels may produce a large log file. By default, GlassFish Server logging messages are recorded in the server log, and you can set the file rotation limit, as described in `rotate-log`(1.)
Ensure that you have correctly assigned users to the correct groups.	Make sure you have assigned the desired set of users to the right groups. In particular, make sure that users assigned to the asadmin group need to be members of that group.

TABLE 6–2 Securing GlassFish Server (Continued)

Security Action	Description
Create no fewer than two user accounts in the asadmin group.	The user admin is created when you install GlassFish Server. For production environments, create at least one other account in the asadmin group in case one account password is compromised. When creating asadmin users give them unique names that cannot be easily guessed.
Assign a password to the admin account.	By default, GlassFish Server includes a single account for user "admin" and an empty password. For production environments this default is inherently unsecure, and you should set a password for admin.

Securing Applications

Although much of the responsibility for securing the GlassFish Server resources in a domain fall within the scope of the server, some security responsibilities lie within the scope of individual applications. For some security options, GlassFish Server 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 6–3 Securing Applications

Security Action	Description
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. To do this, set the `deploy` subcommand `-precompilejsp` option to true for the component. When set to true, the `deploy` and `redeploy` subcommands `-precompilejsp` option compiles JSPs during deploy time. If set to false (the default), JSPs are compiled during runtime.

TABLE 6–3 Securing Applications *(Continued)*

Security Action	Description
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.
Examine applications for security.	There are instances where an application can lead to a security vulnerability. 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 GlassFish 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 GlassFish Server and untrusted classes are being run, the Java security manager may be useful. See "Enabling and Disabling the Security Manager" in *Oracle GlassFish Server 3.1 Application Development Guide*.
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 (`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.