ORACLE®

COMMERCE

Version 11.3

Security Guide

Platform Security Guide

Product version: 11.3
Release date: 04-28-17
Document identifier: ATGCommerceSecurityGuide1704181210

Table of Contents

1 Security Overview

The chapter discusses the Oracle software security assurance initiative and how it relates to the Oracle Commerce Platform.

Oracle Software Security Assurance

Encompassing every phase of the product development lifecycle, Oracle Software Security Assurance (OSSA) is Oracle's methodology for building security into the design, build, testing, and maintenance of its products. Oracle's goal is to ensure that Oracle's products, as well as the customer systems that leverage those products, remain as secure as possible.

Secure Configuration Initiative

Part of this effort is the Secure Configuration Initiative program which ensures that the Oracle products install, out of the box, into a secure state. For more information on the goals of this initiative, see http://www.oracle.com/us/support/assurance/initiative/index.html.

The Security Guides Program

The Secure Configuration Initiative program also includes the Security Guides program, which ensures that all Oracle products have comprehensive documentation on configuring and using the products securely.

The Oracle Commerce Platform Security Guide

Details about security in Oracle Commerce Platform are discussed throughout the Oracle Commerce Platform documentation set. This guide is intended to provide a high-level discussion of security in the Oracle Commerce Platform with links to details in those various documents.

Other Resources

Developers producing customer-facing Web sites using the Oracle Commerce Platform can make use of the security features of the platform. For other insights into secure coding practices and other general methods for making your Web sites secure, consider using information provided by the Open Web Application Security Project (OWASP), see www.owasp.org.

2 Secure Configuration

This chapter discusses security considerations for immediate post-installation configuration of Oracle Commerce Platform. It particularly concentrates on login, SSL, and inter-application communication issues. These considerations are largely invisible to end users, but are critical for application security.

Each section in this chapter provides a brief overview of the security issue under consideration, and provides pointers for more detailed information regarding that issue.

This chapter includes the following sections:

Securing RMI Communications

Oracle Commerce servers can expose certain components to other applications through Java remote method invocation (RMI). To use this feature, you must first write a service according to the RMI specifications, then register your service with the Oracle Commerce RMI server. After registering, other applications can access the service. It is important to insure that these applications access your service in a secure manner.

For more information, see the *Platform Programming Guide*.

You can secure RMI communications by transmitting them over SSL. (For information on non-SSL RMI service implementations, the *Platform Programming Guide*.)

Configuring Keys and Certificates

To use RMI over SSL, configure both public and private keys and wrap the public key in a self-signed certificate. In a production environment, you must create a key store, trust store, and certificate, as described in the *Generating a New Certificate* section of the *Platform Programming Guide*.

For more information about using SSL keys and certificates, and for documentation about the Java Secure Socket Extension (JSSE) APIs, see the Oracle Web site.

Enabling SSL on Oracle Commerce Service Center Applications

Commerce Service Center applications use SSL during user log in. When using HTTPS, you must disable the default access controller. The `<ATG11dir>/Service11.0/Service/Framework/Agent/liveconfig/atg/userprofiling/ProtocolAccessController.properties` file identifies a number of settings required when using HTTPS.

For more detail, see the section *Ensuring HTTPS Connectivity* in the *Commerce Service Center Installation and Programming Guide*, and *Configure SSL* in the *Platform Installation and Configuration Guide*.

Guarding Against Attacks Through Request-Handling Pipeline Configuration

One of the most important tasks for an Oracle Commerce server is handling HTTP requests. The request-handling pipeline represents a potential source of security issues related to denial of service and cross-site attacks, and configuration should be approached carefully.

The Oracle Commerce server extends the basic web server model with Nucleus services that implement the `Servlet` interface, and which are linked in order to process HTTP requests. Each servlet performs a specialized function on a request, then relays the request—sometimes in modified form—to the next servlet in the chain. While each servlet performs a unique service, it often relies on changes that previous servlets made to the request. This chain of servlets is called a *request-handling pipeline*.

For example, a typical request might be processed as follows:

1. Compare the request URI against a list of restricted directories, to make sure that the user has permission to access the specified directory.
2. Translate the request URI into a real file name, taking index files into account when the file name refers to a directory.
3. Given the file name's extension, determine the MIME type of the file.
4. From the MIME type, dispatch the request to the appropriate handler.

The preceding example shows one of many request-handling configurations. Other configurations might dispatch based on a beginning path such as `/cgi-bin` or move the session-tracking step to be performed only for files with the MIME type `text/session-tracked`.

Because the request-handling pipeline is composed of Nucleus components that are independently configurable, it is easy to modify, giving you the flexibility that enterprise applications often require. For additional information on pipeline configuration, see the *Platform Programming Guide*.

Customizing a Request-Handling Pipeline

The Oracle Commerce installation provides a servlet pipeline that is invoked each time an Oracle Commerce server handles a request. The Dynamo Server Admin also has its own servlet pipeline, which starts with the servlet `/atg/dynamo/servlet/adminpipeline/AdminHandler`. You can construct pipelines used by your own applications, or you can customize existing Oracle Commerce server pipelines.

For more information, see the *Customizing a Request-Handling Pipeline* section of the *Platform Programming Guide*.

Basic HTTP Authentication

The `BasicAuthenticationPipelineServlet` class provides authentication using the Basic HTTP authentication mechanism. A component for this servlet is not included in the standard servlet pipelines, but the class is available for use in servlet pipelines that you might create in your own applications. For enhanced security, it is recommended that you use a secure HTTPS protocol.

For more information, see *Authentication* in the *Platform Programming Guide*.

Request Parameter Validation

The Oracle Commerce Platform provides mechanisms for validating request parameters to protect against cross-site attacks. For example, it can be configured to reject a request with a parameter value that contains a `<script>` tag, because this tag could be used to inject malicious JavaScript code.

Separate mechanisms are required for validating query parameters (which are part of the request URL) and POST parameters (which are part of the body of the request). For more information, see the *Validating Request Parameters to Prevent Cross-Site Attacks* chapter of the *Platform Programming Guide*.

Restricting URL Forwarding

When an HTTP request occurs, some applications execute a forward action that includes information from that request. The `atg/dynamo/Configuration` component sets inclusion or exclusion filters that are called from the `ServletUtil` `checkFowardAllowed` method. By default META-INF and WEB-INF paths are excluded, restricting the paths that can perform forwards to URLs from a request.

You can use the forward properties of the `Configuration` component to modify the forwarded URLs that come from unknown or "unsafe" requests. Use the `forwardExclusion` or `forwardInclusion` properties to identify path forwards that should be prevented or allowed.

Preventing User Interface Redress Attacks (Clickjacking)

User interface redress attack (often referred to as clickjacking) is a hacking technique in which a user is tricked into executing malicious code by clicking an apparently innocuous link or button on a web site. For example, a button might have a hidden script that executes when the button is clicked and transmits personal information about the user.

To protect against clickjacking, the `DynamoHandler` servlet in the request-handling pipeline can add Content Security Policy or `X-Frame-Options` fields to HTTP response headers. Inclusion of these fields prevents site pages from being rendered in frames or iframes, thus ensuring that these pages are not embedded in the pages of another site. For more information, see the entry for `DynamoHandler` in *Appendix C: Request Handling Pipeline Servlets Reference* of the *Platform Programming Guide*.

Browser Caching of Dynamic Pages

Some browsers handle page caching in a way that conflicts with dynamic page requests. Oracle Commerce's browser typer marks page requests from those browsers as `non-cacheable` to override the aggressive caching behavior of some browsers and proxy servers. This approach also helps avoid security exposure caused by proxy servers caching.

For more information on preventing browsers from caching dynamic pages, see the *BrowserTyper* section of the *Platform Programming Guide*.

Setting Access Levels for Properties Files

Oracle Commerce components are configured with plain text properties files. You should set access levels on your properties files so they cannot be altered or viewed by unauthorized users. Only site administrators should have read and write permissions. Oracle Commerce must be invoked from an account with these permissions as well. The properties files that contain sensitive information typically reside in each server's `localconfig` directory, but as a general practice, all Oracle Commerce components should be secured.

For more information, see the *Setting Access Levels for Properties Files* section of the *Platform Installation and Configuration Guide*.

Encrypting Credit Card Numbers in CRS

By default, Oracle Commerce does not apply any encryption to credit card information. Commerce Reference Store extends the credit-card item descriptor so that the credit card number is encrypted using the Triple DES encryption algorithm from the Sun JCE security provider. Out of the box, Commerce does not apply any encryption to credit card information.

Hashing Passwords in LDAP Profile Repositories

Lightweight Directory Access Protocol (LDAP) directories are widely used to store personnel information and other kinds of data. Oracle Commerce's LDAP profile repository is an implementation of the Repository API that enables you to store and access profile data in an LDAP directory.

By default, the Personalization module is configured to use a SQL profile repository, but you can change the configuration to use an LDAP repository instead. Using an LDAP repository enables you to tap into the profile data you already have in an LDAP directory, and to share user information across multiple applications.

Just like the SQL profile repository, the LDAP repository implements the Oracle Commerce repository API to allow you to store, access, modify, and query user profile information. As in the SQL profile repository, repository items are first created as transient items (RAM profiles); they become persistent after they are added to the database.

For complete information about LDAP repository concepts, architecture, and code, see the *LDAP Repositories* chapter in the *Repository Guide*.

It is important to note, however, that the LDAP repository implementation is not specific to user profiles in any way. Since an LDAP directory can be used to store any kind of data (people, groups, mailing lists, documents, printers, etc.), you could use the LDAP repository to expose any of that data in an Oracle Commerce application.

Scenarios module and LDAP Repositories: You cannot use scenarios with an LDAP profile repository, because the LDAP repository is not currently powerful enough to express all the data relationships required by the Scenarios module. If you want to run scenarios, you must use either a SQL repository or a composite repository to store all profile information.

Creating the LDAP Profile Repository Component

The LDAP profile repository is a component of class `atg.adapter.ldap.LDAPRepository`. Create and configure an instance of this component as described in the *LDAP Repositories* chapter of the *Repository Guide*.

Configuring the Personalization Module to use the LDAP Repository

By default, the Personalization module is configured to use a SQL database to store profiles. To use an LDAP directory instead, you need to configure Personalization module components to work with the LDAP repository.

For more information, see the *Configuring the Personalization Module to use the LDAP Repository* section of the *Personalization Programming Guide*.

LDAP Password Encryption

The `passwordHasher` property of the `/atg/userprofiling/PropertyManager` component points to a password hasher component that handles password encryption.

For more information, see the *LDAP Password Encryption* section of the *Repository Guide*.

For LDAP servers other than Oracle Directory Server, you may need to create your own `PasswordHasher` implementation, if none of the `PasswordHasher` implementations included in the Oracle Commerce Core Platform meet your requirements.

For more information, see the *Password Hashing* section in the *Customizing Application Security* chapter of the *Platform Programming Guide* for more information about Oracle Commerce's `PasswordHasher` implementations.

For more information, see *User Profiling Tools* in the *Personalization Programming Guide*.

LDAP Profile Repository Definition File

For a sample LDAP profile repository definition file, see the *Sample LDAP Profile Repository Definition File* section of the *Platform Programming Guide*.

Securing LDAP Repositories

The Oracle Commerce LDAP Repository is an implementation of the Repository API that enables you to store and access profile data in an LDAP (Lightweight Directory Access Protocol) directory. The LDAP repository is similar in functionality to the SQL repository, as described earlier in this guide. While by default Oracle

Commerce Scenario Personalization is configured to use an SQL profile repository, you can change the configuration to use an LDAP repository instead.

See the *Personalization Programming Guide* for information about configuring Oracle Commerce to use an LDAP profile repository. LDAP directories are widely used to store personnel information and other kinds of data. LDAP repository lets you to tap into the profile data you already have in an LDAP directory, and to share user information across multiple applications.

Also, you can configure Oracle Commerce's application security scheme to use an LDAP repository, rather than an SQL repository. See the *Managing Access Control* chapter in the *Platform Programming Guide* for more information.

Just like the SQL repository, the LDAP repository implements the Oracle Commerce Repository API to allow you to store, access, modify, and query user profile information. As in the SQL repository, repository items are first created as transient items (RAM profiles); they become persistent after they are added to the database.

It is important to note, however, that the LDAP repository implementation is not specific to user profiles in any way. Because an LDAP directory can be used to store any kind of data—people, groups, mailing lists, documents, printers—you can use the LDAP repository to expose any of that data in Oracle Commerce.

See the *Platform Programming Guide* for an introduction to LDAP terminology, architecture, and concepts.

InitialContextEnvironment

For details on the component that specifies the JNDI environment properties used to create a JNDI `InitialDirContext` to point to your LDAP directory server see the */atg/adapter/ldap/InitialContextEnvironment* section of the *Repository Guide*. You must configure this component to point to your LDAP directory server.

Password Hashing

The `passwordHasher` property of the `/atg/userprofiling/PropertyManager` component points to a password hasher component that handles password encoding.

For more information, see the *LDAP Password Encryption* section of the *Personalization Programming Guide*.

For LDAP servers other than Oracle Directory Server, you might need to create your own `PasswordHasher` implementation, if none of the `PasswordHasher` implementations included in the Oracle Commerce Core Platform meet your requirements.

See the *Working with User Profiles* chapter of the *Personalization Programming Guide* for more information about configuring the `PropertyManager` component.

For detail on password encoding and encryption for Web services, see the *Web Services for Personalization* and *Scenarios* sections of the *Personalization Programming Guide*.

Encrypted Properties in Nucleus Components

If you decide to encrypt sensitive information that is stored in properties files with a symmetrical/asymmetrical encryption method, you must be able to access the encrypted information. A Base64 encoding method should not be used.

For more information, see the *Decoding Encrypted Properties in Nucleus Components* section of the *Platform Programming Guide*.

Single Sign-On for External CRM Applications

CRM applications can call on Commerce Service Center for ticket creation and modification, and solution search. See the *Commerce Service Center Installation and Programming Guide* for more information.

Single Sign-on

CRM integration provides single sign-on capability so that users do not have to log in to Commerce Service Center each time they create a new ticket or modify a ticket from an external CRM application. The single sign-on capability is provided using the `LoginAgentUser` Web service (to authenticate the identity of the user for whom the service was called).

The `LoginAgentUser` Web service is very similar to the standard DPS `Userprofiling` `LoginUser` Web service except that it can handle the Customer Service Agent user subtype whereas the `LoginUser` Web service can only handle the base user type (Internal User). For any application that uses the Customer Service Agent user subtype, the `LoginAgentUser` Web service is required and the `LoginUser` Web service will not work. If client-side encryption is used, then the agent versions of the `CanClientEncryptPasswords`, `GetPassWordHashKey,` and `GetPassWordHashAlgorithm` web services should be used.

LoginAgentUser Web Service

The `LoginAgentUser` Web service authenticates the identity of the agent or user for which the service was called, returning the agent/user profile ID if authentication is successful.

For more information, see the *LoginAgentUser Web Service* section of the *Commerce Service Center Installation and Programming Guide*.

CanClientEncryptAgentPasswords Web Service

The `CanClientEncryptAgentPasswords` Web service is used as part of the optional client-side encryption feature that you can use with the `LoginAgentUser` Web service. It is a utility service that checks to see if a client is configured and able to encrypt passwords for sending via an Oracle Commerce Web service. Currently, only the `LoginUser` and `LoginAgentUser` Web services can handle passwords encrypted by the client.

For more information, see the *CanClientEncryptAgentPasswords Web Service* section of the *Commerce Service Center Installation and Programming Guide*.

Note: See the *Personalization Programming Guide* for more information about using client-side password encryption.

GetAgentPasswordHashKey Web Service

The `GetAgentPasswordHashKey` Web service is used as part of the optional client-side encryption feature. It returns a temporary hash key used by the client to encrypt a password for a single authentication call. The client

encodes the password with this hash key. The server then stores the hash key in the current session so you do not have to return it to the `LoginAgentUser` service.

For more information, see the *GetAgentPasswordHashKey Web Service* section of the *Commerce Service Center Installation and Programming Guide*.

Note: See the *Personalization Programming Guide* for more information about using client-side password encryption.

GetAgentPasswordHashAlgorithm Web Service

The `GetAgentPasswordHashAlgorithm` Web service is used as part of the optional client-side encryption feature. It returns the name of the algorithm that the containing application on the server uses to hash passwords (for example; MD5, SHA, or SSHA). Along with a hash key, this service allows the client to encrypt the password and pass it to the `LoginAgentUser` or Web service over unsecured transport mechanisms.

For more information, see the *GetAgentPasswordHashKey Web Service* section of the *Commerce Service Center Installation and Programming Guide*.

Note: See the *Personalization Programming Guide* for more information about user profiling and password encryption. Follow the directions contained within the *Personalization Programming Guide* for using the Web service to log in before issuing a request to create a ticket or to perform a search using Commerce Service Center. Agents and users do not need to log in to modify the external ticket identification number since that is handled automatically through the Web service call for updating the external ticket identification number and external system name of an existing ticket.

Enabling Java Security Manager for Applications

The Java Security Manager is a java class that can define a security policy for an application. Use it with an application to determine, before performing a possibly unsafe or sensitive operation, what operation is being attempted in a security context. The application can then allow or disallow the operation. For more information, set the *Enabling Java Security Manager for Applications* section of the *Platform Installation and Configuration Guide*.

3 Security Features

This chapter discusses key security features in the Oracle Commerce. These features include mechanisms for authentication, access control, and security audit.

The section below identifies places in the Oracle Commerce Suite documentation that discuss these security features in more detail.

This chapter includes the following sections:

Configuring and Using Authentication

These topics relate to configuration options involving user authentication.

Configuring Security Related to User Profiles

This area includes use of different authentication mechanisms, password expiration, password rule checks, password hashing, and securing cookies with hash keys.

For more information, see the *Personalization Programming Guide*, *Working with User Profiles*.

Customizing Portal Authentication

Oracle Commerce Portal features default user authentication pages, which you can customize (or replace with different ones). The default set of authentication pages includes login and logout forms, and an access denied page. The authentication configuration allows pages to be assigned on a device-specific and community-specific basis.

For more information, see the *Portal Development Guide, Customizing Portal Authentication*.

Configuring Authentication for the Dynamo Server Admin

By default, Dynamo Server Admin requires password authentication to run.

For more information, see the *Platform Programming Guide, Developing and Assembling Nucleus-Based Applications*.

Configuring the Request-Handling Pipeline to Manage Authentication

The BasicAuthenticationPipelineServlet class provides authentication using the Basic HTTP authentication mechanism.

For more information, see the *Platform Programming Guide, Request Handling with Servlet Pipelines*, *Request Handling Pipeline Servlets Reference*.

Authentication for REST Web Services

Before you can use the Oracle Commerce Core Platform REST Web Services, you must log in to open an authorized HTTP session. When the server receives a login request for a valid user account, it authenticates the user and return a session identifier if the authentication is successful.

For more information, see the *Web Services Guide*, *Using REST Web Services*, and the *Web Services Guide*, *Security for REST Web Services*.

Configuring Single Sign On Authentication

The Oracle Commerce can be configured to use Single Sign On (SSO), using Oracle Access Management (OAM). This feature enables the Oracle Commerce Business Control Center and the Oracle Commerce Workbench to share logins, so that when a user logs into one of these environments, that user is automatically also logged into the other environment as well. You can use OAM SSO to authenticate internal users of Oracle Commerce Business Control Center, and Oracle Commerce Workbench using a single authentication step.

Single Sign On Authentication can also be configured using Commerce Single Sign-On. Commerce Single Sign-On ensures that when a user logs into either the Business Control Center or the Workbench, that user is automatically also logged into the other environment.

For more information, see *Appendix D: Using Oracle Access Management for Single Sign On* in the *Platform Installation and Configuration Guide* and the *Commerce Single Sign-On* chapter of the *Platform-Guided Search Integration Guide*.

Configuring and Using Access Control

These topics relate to configuration options involving user access control.

Core Access-Control Facilities for Users, Groups, Roles, Privileges, and Access Control Lists (ACLs)

User account security is managed through the `atg.security` API. Using this API, you can manage persistent user accounts, look up user identities and associate them with roles, manage access control lists, and tie

together multiple security systems running against the same user account database or authentication mechanisms.

The Security Services Interface is a set of fast, flexible APIs that you can use in an application to provide security for the application's features. The Security Management Interface enables programmers to configure account and privilege information with minimal programming.

For more information, see the *Platform Programming Guide*, *Managing Access Control*.

Configuring Access Control for Secured Repositories

The Oracle Commerce secured repository system works in conjunction with the Oracle Commerce Security System to provide fine-grained access control to repository item descriptors, individual repository items, and individual properties through Access Control List (ACL) settings.

For more information, see the *Repository Guide*, *Secured Repositories*.

Configuring Access Control for Assets, Projects, and Workflows in Content Administration

Access to assets, projects, and workflows in Oracle Commerce Content Administration is highly configurable.

For more information, see the *Content Administration Programming Guide*, *Managing User Access and Security*.

Configuring the Access Control System for Oracle Commerce Service

Service security allows you to configure different levels of access to internal users. By creating service security, you can configure access to different elements of the Service application.

For more information, see the *Commerce Service Center Installation and Programming Guide*, *Defining Oracle Commerce Service Security*.

Configuring Access Control for Commerce Service Center Agents

When Commerce Service Center is installed, it is preconfigured with various access rights, global roles, and access controllers. These elements are used to restrict access to certain pages in Commerce Service Center.

For more information, see the *Commerce Service Center Installation and Programming Guide*, *Setting Up Internal Access Control*.

Configuring Access Control for the Oracle Commerce Business Control Center

The Oracle Commerce Business Control Center provides various levels of security, which you can use to control access to the entire UI, to specific activities, or to assets managed within it.

For more information, see the *Business Control Center Administration and Development Guide, Oracle Commerce Business Control Center Security*.

Creating Organizations and Roles

In addition to setting up profiles for individual users (customers who are site visitors, or other types of site users such as administrators), you can set up additional profiles for abstract entities called "organizations" and "roles" and use them to create a multi-level organization of site users grouped by function.

For more information, see the *Personalization Guide for Business Users, Setting Up Visitor Profiles*.

Configuring Users and Roles in Merchandising

Without modification after installation, user access to the Merchandising application is limited to an evaluation Content Administration publishing user account, the administrative user, and the Merchandising user who is given access to all non-administrative parts of the Oracle Commerce Business Control Center.

For more information, see the *Merchandising Administration Guide, Configuring Merchandising*.

Configuring Access Control for Site Administration

Site Administration allows you to share data as well as configure and maintain the sharing relationships between sites. When working in a multisite environment, you can configure sites to share data. Data such as Nucleus components or data objects can be identified as shareable types. Site administrators combine shareable types and sites into a site group, where the shareable types are used by all sites in the group. Access to this data is controlled by managing roles using Site Administration in the Oracle Commerce Business Control Center.

See the *Multisite Administration Guide, Sharing Data*.

Using the Dynamo User Directory to Control Access to Organizations, Roles, and Principals

The Dynamo User Directory allows you to assign access rights to repository items.

For more information, see the *Personalization Programming Guide*, *Working with the Dynamo User Directory*.

Setting up Access Control for Scenarios and Workflows

You can grant or deny access to the features of the Scenarios module by displaying or hiding menu items in the main ACC window.

For more information, see the *Personalization Programming Guide*, *Setting Up Security Access for Scenarios* and *Setting Up Security Access for Workflows*.

Configuring Access Control for Oracle Commerce Portal

Oracle Commerce Portal is subject to the security settings specified within the Oracle Commerce Core Platform. Portal Application Framework security settings are handled primarily from the administrator interface, although additional tags and methods are available to further maintain portal security from within individual gears and the Portal Application Framework itself.

For more information, see the *Portal Administration Guide, Portal Access Control*, and the *Portal Development Guide, Portal Security*.

Access Control for SOAP and REST Web services

The Oracle Commerce Core Platform SOAP and REST Web Services use the underlying security system of the Oracle Commerce Core Platform.

For more information, see the *Web Services Guide*.

Access Control Servlet in Commerce Reference Store and Commerce Service Center

The Access Control Servlet can allow or deny access to a page or group of pages based on criteria such as membership in a group or satisfaction of a targeting rule.

For more information, see the *Commerce Service Center Installation and Programming Guide* and the *Commerce Reference Store IUA Overview*.

OrderLookup Servlet Bean in Commerce

The `/atg/commerce/order/OrderLookup` servlet bean retrieves one or more `Order` objects, depending on the supplied input parameters. `OrderLookup` has a security feature that allows the current user to view only her own orders. By default, this feature is enabled for `/atg/commerce/order/OrderLookup`. To disable the feature, set the `enableSecurity` property to false.

For more information, see the *OrderLookup* section of the *Guide to Setting Up a Store*.

Creating Secure Tasks

When working with sensitive information, you may not want to store property values in clear text or temporary files. Oracle Commerce can collect data from these tasks and store it in a credential store.

For more information, see *Creating Secure Tasks* in the *CIM Developer's Guide*.

User Segment Sharing

The user segment sharing feature allows a content administrator to choose a user segment that has been defined in the Business Control Center as a trigger for a cartridge in Experience Manager. When configuring the user segment sharing feature, you must specify an Oracle Commerce Platform server to act as the user segment server. This server responds to Workbench requests for the list of user segments defined in the Business Control Center. This call must be secure to prevent unwanted access to the user segment data. By default, user segment security is enabled by the `/atg/rest/security/RequestCredentialAccessController` component. However, you must add security credentials to both the user segment server and the Workbench to complete the security configuration. For detailed information on this feature and its configuration, see the *User Segment Sharing* chapter in the *Platform-Guided Search Integration Guide*.

Security Credentials for Oracle Commerce Workbench

In order to submit schema configuration to the Endeca Configuration Repository, the `ConfigImportDocumentSubmitter` component in the Platform-Guided Search integration must access the security credentials for the Oracle Commerce Workbench. You create these credentials in CIM when you set up Oracle Commerce, and information for accessing the credential store is stored in properties of the `/atg/`

`endeca/ApplicationConfiguration` component. For more information, see the *Platform-Guided Search Integration Guide*.

Creating Unique Credentials for Oracle Commerce Servers

For Oracle Commerce servers that share an OPSS installation, you can set up a separate credential wallet for any given server, allowing you to have a unique set of credentials for that particular server. This allows you to limit the credentials that are stored to only those credentials that the server needs to do its tasks. For more details, see the *Managing Access Control* chapter of the *Platform Programming Guide*.

Configuring and Using Security Audit

These topics relate to configuring and using security audit to bolster software security.

Logging System Events and Collecting Data in Oracle Commerce Portal

Oracle Commerce includes three different systems for sending, receiving, and recording messages generated by components: Logging, Data Collection, and Recorders. Oracle Commerce Logging provides a convenient way to log system messages.

For more information, see the *Portal Development Guide*, *Logging and Data Collection*.

Audit Logging of Actions by Commerce Service Center Agents

Commerce Service Center uses audit logging to record actions performed by Commerce Service Center agents in the agent audit repository.

For more information, see the *Commerce Service Center Installation and Programming Guide, Programming Oracle Commerce Service Center*.

Audit Trail for Pricing in Commerce Service Center

Commerce Pricing Calculators provide options for audit trails which can be used for security related purposes.

For more information, see the *Commerce Service Center Installation and Programming Guide, Commerce Pricing Calculators*.

Recording an Audit Trail for Scenario Activity

For the purposes of managing your company's relationship with its site visitors, it is useful to be able to track what happens as a result of the elements within a scenario. For example, if you set up a scenario that sends a promotional e-mail to new members offering them a discount on a product, it is helpful to keep a record of the members to whom the e-mail is sent. Information such as this can help you monitor the success of your promotions, and it also allows you to provide better customer service.

For more information, see the *Personalization Guide for Business Users, Creating Scenarios* and the *Personalization Programming Guide*.

User Event Logging

When you send events to the Oracle Commerce logging system, you can record useful information about the operation of your Web application. The Personalization module's logging system handles page requests (from URLs), user events (such as new session, login, and so on), and content viewed from content repositories.

For more information, see the *Personalization Programming Guide*, *Personalization Module Logging*.

Content Event Logging

Oracle Commerce Personalization lets you customize content for specific users and events, including event logging.

For more information, see the *Page Developer's Guide, Serving Targeted Content with Oracle Commerce Servlet Beans*.

Login Attempt Logging

Oracle Commerce includes a login auditing feature that records each attempt to log into its administration applications. The login auditing feature writes information about each attempt in a file on your Web application server's file system. Use the audit log file to help detect unauthorized attempts to access your administration applications.

For more information, see the *Platform Installation and Configuration Guide*, *Recording Login Attempts*.

Index