Security Measures Guide
Oracle Banking Trade Finance Process Management
Release 14.6.1.0.0

Part Number F61853-01
August 2022

ORACLE
FINANCIAL SERVICES

Table of Contents

ORACLE

1. About this Manual

1.1 Purpose

This guide is designed to help user to quickly get familiar with the Security Measure of Oracle Banking Microservices Architecture products. It provides an overview of the Security Measure and takes you through the various security features that Oracle Banking Microservices Architecture product offers.

1.2 Audience

This guide is primarily intended for Developers for Oracle Banking Microservices Architecture product and third party or vendor software's. Some information may be relevant to IT decision makers and users of the application are also included. Readers are assumed to possess basic operating system, network, and system administration skills with awareness of vendor/third-party software's and knowledge of Oracle Banking Microservices Architecture products.

1.3 Scope

1.3.1 Read Sections Completely

Each section should be read and understood completely. Instructions should never be blindly applied. Relevant discussion may occur immediately after instructions for an action, so be sure to read whole sections before beginning implementation.

1.3.2 Understand the Purpose of this Guidance

The purpose of the guidance is to provide security-relevant code and configuration recommendations.

1.3.3 Limitations

This guide is limited in its scope to security-related guideline for developers.

2. Overview

Controlled access to the system is a basic parameter that determines the robustness of the security in banking software. In Oracle Banking Microservices Architecture products, we have employed a multi-pronged approach to ensure that this application is secure.

3. OBTFM- Product Controls

3.1 Authentication

First, only authorized users can access the system with the help of a unique User ID and a password. Each unique user authenticated by the system must maintained in the LDAP server.

3.2 Role Based Access Controls

It is likely that users working in the same department at the same level of hierarchy need to have similar user profiles. In such cases, you can define a Role Profile that includes access rights to the functions that are common to a group of users. A user can be linked to a Role Profile by which you give the user access rights to all the functions in the Role Profile.

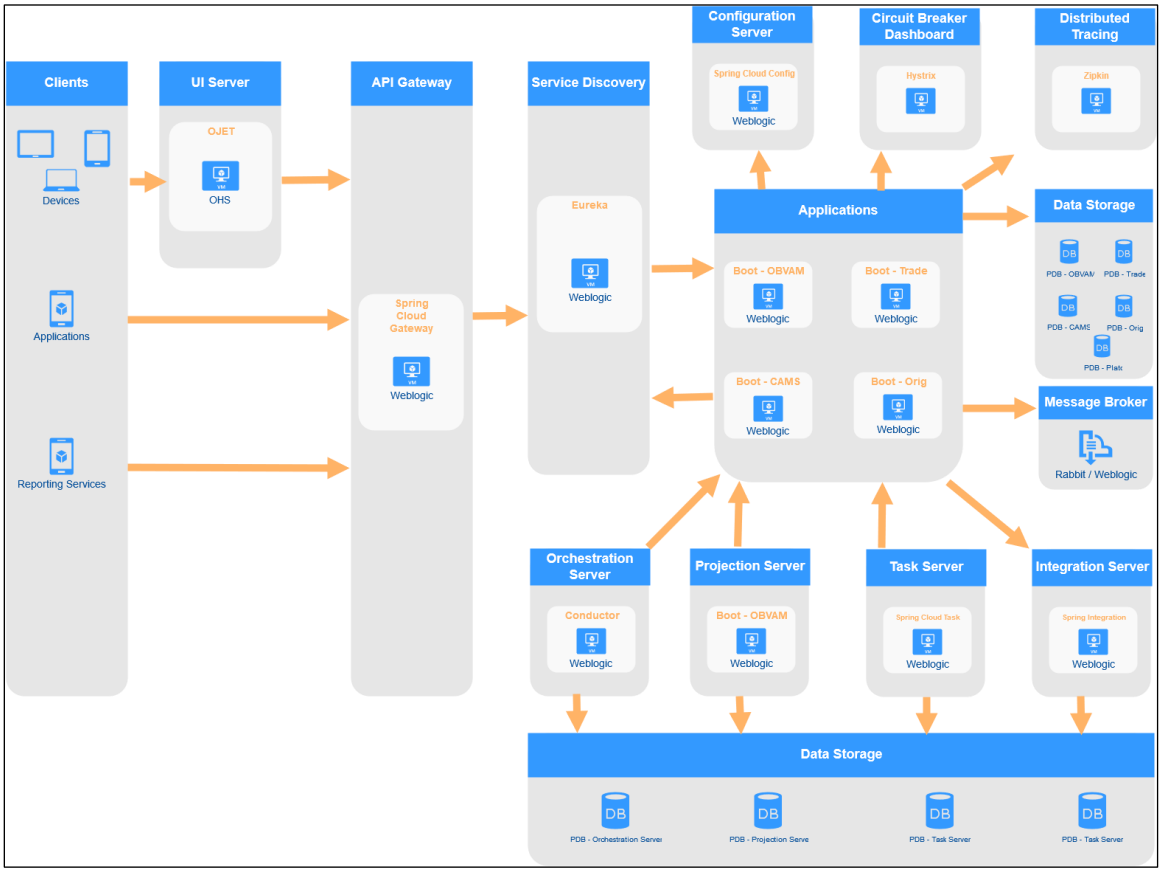
Application-level access has implemented via the Security Management System (SMS) module. SMS supports “ROLE BASED” access of Screens and different types of operations.

Oracle Banking Microservices Architecture products support dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights. Apart from the role-based access control particular functions, products can be restricted for user as described below.

3.3 Branch Level Access Controls

Roles are granted to a user for each branch that they need access to separately.

Deployment Architecture Diagram



4. Validation

4.1 Secure Transformation of Data (SSL)

A two-way SSL is used when the server needs to authenticate the client. In a two-way SSL connection, the client verifies the identity of the server and then passes its identity certificate to the server. The server then validates the identity certificate of the client before completing the SSL handshake.

If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic.

Below configuration must be ensured in weblogic.xml within the deployed application ear.

- Cookies are set with Http only as true
- Cookie secure flag set to true
- Cookie path to refer to deployed application

4.2 Sign-On messages

The below table shows the general Sign-On messages which would be displayed to the user during invalid authentication.

Message	Explanation
User Authentication Failed	An incorrect user ID or password was entered.
User Status is Disabled. Please contact your System Administrator	The user profile has been disabled due to number of dormancy days allowed for the user has exceeded the dormancy days configured in the system.
User Status is Locked. Please contact your System Administrator	The user profile has been locked due to an excessive number of attempts to login, using an incorrect user ID or password. The number of attempts could have matched either the successive or cumulative number of login failures (configured for the system).

4.3 CSRF Token Validation

System identifies the request using the JWT short live issued during the login. The XMLHttpRequest object sets a custom HTTP Authorization header in the request with JWT, with the header value being the Cross-site request forgery token; the server then verifies for the presence of such a header and the Cross-site request forgery token. This serves as a protection at endpoints used for XMLHttpRequest requests, since only XMLHttpRequest objects can set HTTP headers.

4.4 Cross-Site Scripting (XSS)

XSS for Oracle Banking Microservices Architecture product handled by OJET. Hence the application developer’s need not to handle specifically

4.5 Clickjacking/Frame-bursting

Oracle JET handles clickjacking/Frame-bursting attack. Oracle Banking Microservices Architecture product uses the X-Frame-Options HTTP response header to indicate whether or not a browser should be allowed to render a page in a <frame> or <iframe>. This is used to avoid Clickjacking attacks, by ensuring that the content is not embedded into other sites.

Evidence:

response.addHeader("X-FRAME-OPTIONS", "DENY");

4.6 CACHE Control in Servlet and jsp

There are three basic HTTP response headers that prevent a page from being cached to disk. Different browsers handle them in slightly different ways, so they need to be used in combination to ensure all browsers do not cache the specific page. These headers are "Expires", "Pragma" and "Cache-control". In addition, these headers can either be sent directly by the server or placed in the HTML code as HTTP-EQUIV META tags within the HEAD section. The "Expire" header gives a date at which point the page should expire and no longer be cached. Internet Explorer supports a date of "0" for immediately and any negative number for already expired. The "Pragma: no-cache" header indicates that the page should not be cached.
uses below code to prevent cache control.

```
response.setHeader( "Pragma", "no-cache" );
response.setHeader( "Cache-Control", "no-cache" );
response.setHeader( "Cache-Control", "no-store" );
response.setDateHeader( "Expires", -1 );
```

4.7 SECURE RANDOM INSTEAD OF RANDOM

The application uses a SecureRandom class to generate random number wherever required.

4.8 Injection

Injection flaws occur when an application sends untrusted data to an interpreter. Injection flaws are very prevalent, particularly in legacy code. They are often found in SQL, LDAP, Xpath, or SQL queries; OS commands; XML parsers, SMTP Headers, program arguments, etc. Injection flaws are easy to discover when examining code.

Oracle Banking Microservices Architecture product uses Oracle Database and it has adequate inbuilt techniques to prevent SQL injections as underlined below: -

1. **Use of parameterized queries** - Oracle Banking Microservices Architecture products uses queries with bind variables to construct and execute SQL statements in JAVA.

Evidence: -

```
query =entityManager.createquery("select obj from Country obj where obj.countryId = ?");
query.setString(1,countryId);
```

4.9 Field Validations

Field level validations exist for all mandatory fields. Database too had limits on the type and the length of data. Blacklisted characters are not allowed in the mandatory fields. Nevertheless, Oracle Banking Microservices Architecture products has free-text fields, which takes all data, entered by the user, as a String.

4.10 Restriction on Blacklist Characters

Below table shows the list of bad characters which should not be allowed in URL path, but the application's operations require many of the below characters to be passed in the request. So, Oracle Banking Microservices Architecture product will encode the below bad characters before sending them through the URL and same will be decoded at the server to prevent the hacker from modifying the request.

Bad URL Characters (Unsafe Characters)	
&	//
<	./
>	/.
;	/*
\"	*.
\'	~
%	\
)	25%
(	%25u
+	%25U
,	%00-%1f, %7f-%ff
" " (space)	%00-%1f and %7f-%ff
-	%25u and %25U

4.11 Unhandled Exception

Virtual Pages takes care of it at application level.

All unhandled exceptions are handled via an OOPs Page. This page suggests the user to contact the system administrator.

5. Session Management

Oracle Banking Microservices Architecture product doesn't maintain the state of the client side. It's a pure stateless micro services base platform.

5.1 Cryptography Used

PCI council defines **Strong Cryptography** as:

Cryptography based on industry-tested and accepted algorithms, along with strong key lengths and proper key-management practices. Cryptography is a method to protect data and includes both encryption (which is reversible) and hashing (which is not reversible, or "one way"). SHA-1 is an example of an industry-tested and accepted hashing algorithm. Examples of industry-tested and accepted standards and algorithms for encryption include AES (128 bits and higher), TDES (minimum double-length keys), RSA (1024 bits and higher), ECC (160 bits and higher), and ElGamal (1024 bits and higher).

Encryption algorithm: The application leverages AES encryption algorithm to store sensitive information into properties file. This algorithm uses 256 bit secret key for encryption and decryption which would be stored at property file.

Hashing algorithm: Oracle Banking Microservices Architecture product platform service leverages HS-512 hashing algorithm with random salt for JWT.

5.2 Session Logging

Unsuccessful attempt to login is stored in the database with timestamp. Invalid and expired tokens submitted to the application are categorized as authentication failures and the same are logged.

6. Password Management

6.1 Password Protection

Passwords are hashed using SHA-512 algorithm and same will be stored in database table. This is applicable for

a) Credentials used for integration with other applications which will happen through OAUTH.
b) Credentials to access the LDAP server (where user credentials are stored)

6.2 Password protection over transmission from browser to database

Passwords are protected using SSL over transmission from browser to database.

7. Exception/Error handling

7.1 Exception Handling in Java

Different types of exceptions can rise in application. Java exceptions handled using try catch blocks available in java. Sometimes we use the Throw statement to throw an exception which is caught by the catch block. Caught exceptions will be written into the log files for the debug purpose whenever required. Whenever any exception occurs in application, proper information used to send to the front-end user by showing alert.

8. Logging

Oracle Banking Microservices Architecture products implement accurate logging to identify any security threats in the system which when closely watched can help identify a security attack before it is too late.

Oracle Banking Microservices Architecture Security Measures Guide

Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway

Goregaon (East)

Mumbai, Maharashtra 400 063

India

Worldwide Inquiries:

Phone: +91 22 6718 3000

Fax: +91 22 6718 3001

https://www.oracle.com/industries/financial-services/index.html