

Oracle® Banking Trade Finance Process Management Development Security Guide



Release 14.8.0.0.0
G28612-01
April 2025

ORACLE®

Copyright © 2007, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Purpose	iv
Audience	iv
Scope	iv
Documentation Accessibility	v
Critical Patches	v
Diversity and Inclusion	v
Related Documents	v
Conventions	vi
Acronyms and Abbreviations	vi

1 How to address the OWASP Top10 in Oracle Banking Trade Finance Process Management

1.1	Injection	1-1
1.2	Broken Authentication and Session Management	1-2
1.2.1	Cryptography Used	1-2
1.3	Cross-Site Scripting (XSS)	1-2
1.4	Insecure Direct Object References	1-2
1.5	Security Misconfiguration	1-3
1.6	Sensitive Data Exposure	1-3
1.7	Missing Function Level Access Control	1-5
1.8	Cross-Site Request Forgery (CSRF)	1-5
1.9	Using Components with Known Vulnerabilities	1-5
1.10	Unvalidated Redirects and Forwards Network Security	1-5

2 Securing API Services

Preface

- [Purpose](#)
This document provides security-related usage and configuration recommendations for Oracle Banking Trade Finance Process Management 14.8.0.0.0. This guide may outline procedures required to implement or secure certain features, but it is also not a general-purpose configuration manual.
- [Audience](#)
- [Scope](#)
- [Documentation Accessibility](#)
- [Critical Patches](#)
- [Diversity and Inclusion](#)
- [Related Documents](#)
- [Conventions](#)
- [Acronyms and Abbreviations](#)

Purpose

This document provides security-related usage and configuration recommendations for Oracle Banking Trade Finance Process Management 14.8.0.0.0. This guide may outline procedures required to implement or secure certain features, but it is also not a general-purpose configuration manual.

Audience

This guide is primarily intended for Developers for Oracle Banking Trade Finance Process Management 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 Trade Finance Process Management application.

Scope

- [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.
- [Understand the Purpose of This Guidance:](#)
The purpose of the guidance is to provide security-relevant code and configuration recommendations.

- [Limitations](#)
This guide is limited in its scope to security-related guideline for developers.

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.

Understand the Purpose of This Guidance:

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

Limitations

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

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Critical Patches

Oracle advises customers to get all their security vulnerability information from the Oracle Critical Patch Update Advisory, which is available at [Critical Patches, Security Alerts and Bulletins](#). All critical patches should be applied in a timely manner to ensure effective security, as strongly recommended by [Oracle Software Security Assurance](#).

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For more information, you can refer to the following documents:

- Oracle Banking Trade Finance Process Management Pre Installation Guide

- Oracle Banking Trade Finance Process Management Services Installation Guide

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Acronyms and Abbreviations

The list of the acronyms and abbreviations that are used in this guide are as follows:

Table 1 Acronyms and Abbreviations

Abbreviation	Description
OBTfPM	Oracle Banking Trade Finance Process Management
XSS	Cross Site Scripting
CSRF	Cross Site Request Forgery
OJET	Oracle JavaScript Extension Toolkit

1

How to address the OWASP Top10 in Oracle Banking Trade Finance Process Management

This topic consists of following sub-topics:

- [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.
- [Broken Authentication and Session Management](#)
- [Cross-Site Scripting \(XSS\)](#)
XSS for Oracle Banking Trade Finance Process Management handled by OJET. Hence application developer's need not to handle specifically.
- [Insecure Direct Object References](#)
- [Security Misconfiguration](#)
- [Sensitive Data Exposure](#)
- [Missing Function Level Access Control](#)
- [Cross-Site Request Forgery \(CSRF\)](#)
Oracle Banking Trade Finance Process Management services are stateless. Oracle Banking Trade Finance Process Management generates JWT upon successful authentication of the users. The generated token works to prevent CSRF.
- [Using Components with Known Vulnerabilities](#)
Source code scanning done using the latest fortify to identify the sources code issue and will provide the proper fix for the reported issues.
- [Unvalidated Redirects and Forwards Network Security](#)
Application uses 302 redirect wherever required. Oracle Banking Trade Finance Process Management uses `response.sendRedirect(newURL);`

1.1 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 Trade Finance Process Management uses Oracle database and it has adequate inbuilt techniques to prevent SQL injections as underlined below:-

1. **Use of prepared statements (parameterized queries)** — Oracle Banking Trade Finance Process Management uses parameterized JPQL/SQL queries with bind variables to construct and execute SQL statements in JAVA.
2. **Escaping all user supplied input**— This third technique is to escape user input before putting it in a query. If it's a concern that rewriting the dynamic queries as prepared statements or stored procedures might break the application or adversely affect

performance, then this might be the best approach for the purpose. However, this methodology is frail compared to using parameterized queries and there's no guarantee that it will prevent all SQL Injection in all situations.

1.2 Broken Authentication and Session Management

In Oracle Banking Trade Finance Process Management, applications are stateless microservices based architecture.

This topic contains following sub-topics:

- [Cryptography Used](#)

1.2.1 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 Trade Finance Process Management platform service leverages HS-512 hashing algorithm with random salt for JWT.

1.3 Cross-Site Scripting (XSS)

XSS for Oracle Banking Trade Finance Process Management handled by OJET. Hence application developer's need not to handle specifically.

1.4 Insecure Direct Object References

1. Use of Prepared Statements (Parameterized Queries):

Oracle Banking Trade Finance Process Management uses parameterized JPQL/SQL queries with bind variables to construct and execute SQL statements in JAVA.

2. Input Validation

Oracle Banking Trade Finance Process Management is a web based application, the request data from browser to server will be passed using request headers and request parameters. All the request fields coming from the client are validated using whitelist validation to prevent cross site scripting.

User defined methods used for input validation which checks each character of the request field with a range of allowed characters. In addition, OJET framework handles the input attribute validations. User defined methods `escapeJavaScript()`, `escapeHTML()` and `escapeURL()` will sanitize the output data before flushing it into client browser. `escapeJavaScript()` will escape all characters except immune JavaScript characters and alphanumeric characters in the ASCII character set. All other characters are encoded

using the `\xHH` or `\uHHHH` notation for representing ASCII or Unicode sequences. `escapeHTML()` will escape the characters with equivalent HTML entities obtained from the lookup map. Lookup map will have entities such as amp, quot, lt, gt etc. `escapeURL()` will encode the URL using `URLEncoder` class. Whitelist validation is also used to restrict Image/signature/excel upload and to check rights for every operation performed by user.

3. Field Validation

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 Trade Finance Process Management has free-text fields, which takes all data, entered by the user, as a String.

4. Restriction on Blacklist Characters

Blacklisted characters on Oracle Banking Trade Finance Process Management handled by OJET. Hence application developer's need not to handle specifically.

1.5 Security Misconfiguration

1. Configuration Files

Configuration files are securely placed inside the Classes folder of the WEB-INF folder which is not publicly accessible.

2. 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.

3. BI Publisher Reports – Generation and Access

The application uses a sandbox for placing the generated reports file into a sandbox area. The sandbox is placed in a specified location (the location will be specified in the properties file) on the server. The application validates if the user has explicit Rights to generate Reports.

1.6 Sensitive Data Exposure

1. Secure Transformation of Data (SSL):

The Oracle Banking Trade Finance Process Management allows a deployer to configure the application such that all HTTP connections to the application are over SSL/TLS. In other words, all HTTP traffic in the clear will be prohibited; only HTTPS traffic will be allowed. It is mandatory to enable this option in a production environment, especially when WebLogic Server acts as the SSL terminator.

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.

In order to establish a two-way SSL connection, need to have two certificates, one for the server and the other for client. This is required for de-centralized setup of application.

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

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

```
<wls: session-descriptor>
<wls: cookie-http-only>true</wls: cookie-http-only>
</wls: session-descriptor>

<wls: session-descriptor>
<wls: cookie-secure>true</wls: cookie-secure>
<wls: url-rewriting-enabled>false</wls: url-rewriting-enabled>
</wls: session-descriptor>
```

Always make sure Cookies are set with always Auth Flag enabled by default for WebLogic server

2. Sign-On Messages

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

Message	Element
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).

3. 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.

4. Clickjacking/Frame-Bursting

Oracle JET handles clickjacking/Frame-bursting attack. Oracle Banking Trade Finance Process Management 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

1.7 Missing Function Level Access Control

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.

Roles are granted to a user at the level of each branch, thereby controlling what functions the user can perform in which branch.

Oracle Banking Trade Finance Process Management solutions supports dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights.

1.8 Cross-Site Request Forgery (CSRF)

Oracle Banking Trade Finance Process Management services are stateless. Oracle Banking Trade Finance Process Management generates JWT upon successful authentication of the users. The generated token works to prevent CSRF.

1.9 Using Components with Known Vulnerabilities

Source code scanning done using the latest fortify to identify the sources code issue and will provide the proper fix for the reported issues.

3rd party libraries scanning for every release has been done to validate if any security issues rise for any of the components or not. Update the 3PL with latest security patch or upgraded to latest version

1.10 Unvalidated Redirects and Forwards Network Security

Application uses 302 redirect wherever required. Oracle Banking Trade Finance Process Management uses `response.sendRedirect(newURL);`

2

Securing API Services

Different applications deployed on disparate platforms and using different infrastructure need to be able to communicate and integrate seamlessly with Oracle Banking Trade Finance Process Management in order to exchange data. The Oracle Banking Trade Finance Process Management Service API Gateway will cater to these integration needs.

The integration needs supported by the Gateway can be broadly categorized from the perspective of the Gateway as follows:

- Inbound application integration – used when any external system needs to add, modify or query information within Oracle Banking Trade Finance Process Management.
- Outbound application integration – used when any external system needs to be accessed for processing transactions within Oracle Banking Trade Finance Process Management.