Development Security Guide
Oracle Banking Electronic Data Exchange for Corporates
Release 14.5.0.0.0

Part No. F46661-01

September 2021

ORACLE

Development Security Guide

September 2021

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

www.oracle.com/financialservices/

Table of Contents

1. Preface

1.1 Intended Audience

This document is intended for the following audience:

- Customers
- Partners

1.2 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.

1.3 Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit

http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

1.4 Structure

This manual is organized into the following categories:

Preface gives information on the intended audience. It also describes the overall structure of the User Manual.

The subsequent chapters describes following details:

- Introduction
- Preferences & Database
- Configuration / Installation.

1.5 Related Information Sources

For more information on Oracle Banking Electronic Data Exchange for Corporates Release 14.5.0.0.0, refer to the following documents:

- Oracle Banking Electronic Data Exchange for Corporates Installation Manuals

2. Database Setup

2.1 Introduction

Purpose:

This document provides security-related usage and configuration recommendations for Oracle Banking Electronic Data Exchange for Corporates 14.5.0.0.0. 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 Electronic Data Exchange for Corporates and third party or vendor software's. Some information may be relevant to IT decision makers and users of the application are 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 Electronic Data Exchange for Corporates application.

2.2 Scope

2.2.1 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.

2.2.2 Understand the Purpose of this Guidance

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

2.2.3 Limitations

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

3. How to address the OWASP Top10 in Oracle Banking Electronic Data Exchange for Corporates

3.1 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 Electronic Data Exchange for Corporates 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 Electronic Data Exchange for Corporates uses parameterized JPQL/SQL queries with bind variables to construct and execute SQL statements in JAVA.

3.2 Cross-Site Scripting (XSS)

XSS for Oracle Banking Electronic Data Exchange for Corporates handled by OJET. Hence application developer's need not to handle specifically

3.3 Insecure Direct Object References

1. **Use of prepared statements (parameterized queries)**

   Oracle Banking Electronic Data Exchange for Corporates uses parameterized JPQL/SQL queries with bind variables to construct and execute SQL statements in JAVA.

2. **Input Validation**

   Oracle Banking Electronic Data Exchange for Corporates 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 white list 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.

White list 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 Electronic Data Exchange for Corporates 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 Electronic Data Exchange for Corporates handled by OJET. Hence application developer's need not to handle specifically.

3.4 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.

3.5 Sensitive Data Exposure

1. **Secure Transformation of Data (SSL)**

The Oracle Banking Electronic Data Exchange for Corporates 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 as 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	Explanation
User Authentication Failed	An incorrect user ID or password was entered.

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 Electronic Data Exchange for Corporates 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.

3.6 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 Electronic Data Exchange for Corporates solutions supports dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights.

3.7 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.

Home