

Security Guide
Oracle Banking Liquidity Management
Release 14.5.3.0.0
Part Number F50376-01
November 2021



Table of Contents

| | |
|---|-----------|
| 1. ABOUT THIS MANUAL..... | 1 |
| 1.1 INTRODUCTION..... | 1 |
| 1.2 SCOPE..... | 1 |
| 1.2.1 <i>Read Sections Completely</i> | 1 |
| 1.2.2 <i>Understand the Purpose of this Guidance</i> | 1 |
| 1.2.3 <i>Limitations</i> | 1 |
| 1.2.4 <i>Test in Non-Production Environment</i> | 1 |
| 2. PREREQUISITE..... | 2 |
| 2.1 OPERATING ENVIRONMENT SECURITY..... | 2 |
| 2.2 NETWORK SECURITY..... | 2 |
| 2.3 ORACLE DATABASE SECURITY..... | 2 |
| 2.3.1 <i>Oracle Banking Liquidity Management Recommended configuration</i> | 2 |
| 2.4 APPLICATION SERVER SECURITY..... | 4 |
| 2.5 CHOICE OF THE SSL CIPHER SUITE..... | 4 |
| 2.6 SECURING THE ORACLE BANKING LIQUIDITY MANAGEMENT APPLICATION..... | 5 |
| 2.6.1 <i>Online Web Application</i> | 5 |
| 2.6.2 <i>API Layer</i> | 8 |
| 2.6.3 <i>Two-way SSL Connection</i> | 10 |
| 3. SECURING ORACLE BANKING LIQUIDITY MANAGEMENT..... | 11 |
| 3.1 DESKTOP SECURITY..... | 11 |
| 3.2 ORACLE BANKING LIQUIDITY MANAGEMENT CONTROLS..... | 11 |
| 3.2.1 <i>Overview</i> | 11 |
| 3.2.2 <i>Disable Logging</i> | 11 |
| 3.2.3 <i>Sign-on Messages</i> | 11 |
| 3.2.4 <i>Authentication & Authorization</i> | 11 |
| 3.2.5 <i>Role Based Access Controls</i> | 12 |
| 3.2.6 <i>Access Controls - Branch Level</i> | 12 |
| 3.2.7 <i>Maker – Checker</i> | 12 |
| 3.2.8 <i>Access Enforcement</i> | 12 |
| 3.2.9 <i>Password Management</i> | 12 |
| 4. GENERAL INFORMATION..... | 14 |
| 4.1 CRYPTOGRAPHY..... | 14 |
| 4.2 SECURITY PATCH..... | 14 |
| 4.3 ORACLE DATABASE SECURITY SUGGESTIONS..... | 14 |
| 4.4 ORACLE SOFTWARE SECURITY ASSURANCE - STANDARDS..... | 14 |
| 4.5 REFERENCES..... | 15 |
| 4.5.1 <i>Datacenter Security considerations</i> | 15 |
| 4.5.2 <i>Database Security considerations</i> | 15 |
| 4.5.3 <i>Security recommendations / practices followed for Database Environment</i> | 15 |
| 4.5.4 <i>Common security considerations</i> | 15 |

1. About this Manual

1.1 Introduction

Purpose:

This guide provides security-related usage and configuration recommendations for Oracle Banking Liquidity Management. 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 IT department or administrators deploying Oracle Banking Liquidity 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 Liquidity Management application.

1.2 Scope

1.2.1 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.

1.2.2 Understand the Purpose of this Guidance

The purpose of the guidance is to provide security-relevant configuration recommendations. It does not imply the suitability or unsuitability of any product for any particular situation, which entails a risk decision.

1.2.3 Limitations

This guide is limited in its scope to security-related issues. This guide does not claim to offer comprehensive configuration guidance. For general configuration and implementation guidance refer to other sources such as Vendor specific sites.

1.2.4 Test in Non-Production Environment

To the extent possible, guidance should be tested in a non-production environment before deployment.

Ensure that any test environment simulates the configuration in which the application will be deployed as closely as possible.

2. Prerequisite

2.1 Operating Environment Security

Please refer the vendor specific documentation for making the environment safer and more secured.

2.2 Network Security

Please refer the vendor specific documentation for making the environment safer and more secured.

2.3 Oracle Database Security

Please refer the Oracle Database Security specification document for making the environment safer and more secured.

2.3.1 Oracle Banking Liquidity Management Recommended configuration

This section contains security recommendations for the Database used for Oracle Banking Liquidity Management Application.

| | | |
|-------------------|-------------------------------------|----------------|
| Init.ora | REMOTE_OS_AUTHENT=FALSE | Authentication |
| Init.ora | TRACE_FILES_PUBLIC=FALSE | Authorization |
| Init.ora | REMOTE_OS_ROLES=FALSE | Authorization |
| Init.ora | O7_DICTIONARY_ACCESSIBILITY = FALSE | Authorization |
| Init.ora | AUDIT_TRAIL = OS | Audit |
| Init.ora | AUDIT_FILE_DEST = E:\logs\db\audit | Audit |
| To audit sessions | SQL> audit session; | Audit |

| | | |
|-------------------------|---|-------|
| To audit schema changes | SQL> audit user; | Audit |
| To audit other events | SQL> AUDIT DATABASE LINK; -- Audit create or drop database links SQL> AUDIT PUBLIC DATABASE LINK; -- Audit create or drop public database links SQL> AUDIT SYSTEM AUDIT; -- Audit statements themselves SQL> AUDIT ALTER ANY ROLE by ACCESS; -- Audit alter any role statements SQL> AUDIT ALTER DATABASE by ACCESS; -- Audit alter database statements SQL> AUDIT ALTER SYSTEM by ACCESS; -- Audit alter system statements SQL> AUDIT CREATE ROLE by ACCESS; -- Audit create role statements SQL> AUDIT DROP ANY ROLE by ACCESS; -- Audit drop any role statements SQL> AUDIT PROFILE by ACCESS; -- Audit changes to profiles SQL> AUDIT PUBLIC SYNONYM by ACCESS; -- Audit public synonyms statements SQL> AUDIT SYSDBA by ACCESS; -- Audit SYSDBA privileges SQL> AUDIT SYSOPER by ACCESS; -- Audit SYSOPER privileges SQL> AUDIT SYSTEM GRANT by ACCESS; -- Audit System grant privileges | Audit |

To audit the events, login through sqlplus as SYSTEM and issue the commands.

2.4 Application Server Security

Please refer the Oracle Web Logic Security specification document for making the environment safer and more secured.

Support for LDAP (External Password Authentication)

Oracle Banking Liquidity Management Solution supports the following authentication schemes for the online web application

- Standard LDAP Directory (e.g. OUD/AD)
- SSO with OAM (Oracle Access Manager – Part of the Oracle Identity Management Suite)
- SAML assertions with a Service Provider protecting the resource and an Identity Provider.

Oracle Banking Liquidity Management Solution supports the following authentication scheme for the API layer

- OAuth (CLIENT CREDENTIALS) with OAM

In case the customer does not have OAM, it is expected that the customer has an enterprise API Management Layer that protects Oracle Banking Liquidity Management's API layer with the same controls (i.e. OAuth)

Support for SSL (Secure Transformation of Data)

The Oracle Banking Liquidity Management to be configured 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 highly recommended to enable this option in a production environment, especially when WebLogic Server acts as the SSL terminator.

2.5 Choice of the SSL cipher suite

Oracle WebLogic Server allows for SSL clients to initiate a SSL connection with a null cipher suite. The null cipher suite does not employ any bulk encryption algorithm thus resulting in transmission of all data in clear text, over the wire.

The default configuration of Oracle WebLogic Server is to disable the null cipher suite. Ensure that the usage of the null cipher suite is disabled, preventing any client from negotiating an insecure SSL connection.

Furthermore, for installations having regulatory requirements requiring the use of only 'high' cipher suites, Oracle WebLogic Server can be configured to support only certain cipher suites. The restriction can be done in config.xml of the WebLogic domain. Provided below is an example config.xml restricting the cipher suites to those supporting 128-bit symmetric keys or higher, and using RSA for key exchange.

```
....
<ssl>
  <enabled>true</enabled>
  <ciphersuite>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</ciphersuite>
</ssl>
....
```

- Configuration of WebLogic Server to support the above defined cipher suites might also require an additional command line argument to be passed to WebLogic Server, so that a FIPS 140-2 compliant crypto module is utilized. This is done by adding - **Dweblogic.security.SSL.nojce=true** as a JVM argument.
- The restriction on cipher suites needs to be performed for every managed server.
- The order of cipher suites is important – Oracle WebLogic Server chooses the first available cipher suite in the list, that is also supported by the client.
- Cipher suites with RC4 are enabled despite it being second best to AES. This is primarily for older clients that do not support AES (for instance, Microsoft Internet Explorer 6, 7 and 8 on Windows XP).

2.6 Securing the Oracle Banking Liquidity Management Application

Securing the Oracle Banking Liquidity Management Application includes securing

- A) The Online Web Application and
- B) The API Layer exposed to external consumers

2.6.1 Online Web Application

Access to the online web application is granted only via the following methods

- Standard LDAP Directory authentication
- SSO with OAM and
- SAML with the Oracle Banking Liquidity Management application acting as the service provider

In addition to the authentication, the Oracle Banking Liquidity Management online web application uses JWT (JSON Web Tokens) to maintain the state for authenticated users.

JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties. JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed.

- **No Session to Manage (stateless):** The JWT is a self-contained token which has authentication information, expire time information, and other user defined claims digitally signed.
- **Portable:** A single token can be used with multiple backend.
- **No Cookies Required, So It's Very Mobile Friendly**
- **Good Performance:** It reduces the network round trip time.
- **Decoupled/Decentralized:** The token can be generated anywhere. Authentication can happen on the resource server, or easily separated into its own server.

In addition, the following policies are followed for JWT,

- **Token Store:** To increase the security and better usability, every authentication/refresh request is secured by random unique key. The generated token and the secure key are persisted in the table, so that during the horizontal scaling of the servers, any API gateway instance can serve for the request.
- **Cipher strength:** Platform security module hashes the JWT footer with HS512 algorithm.

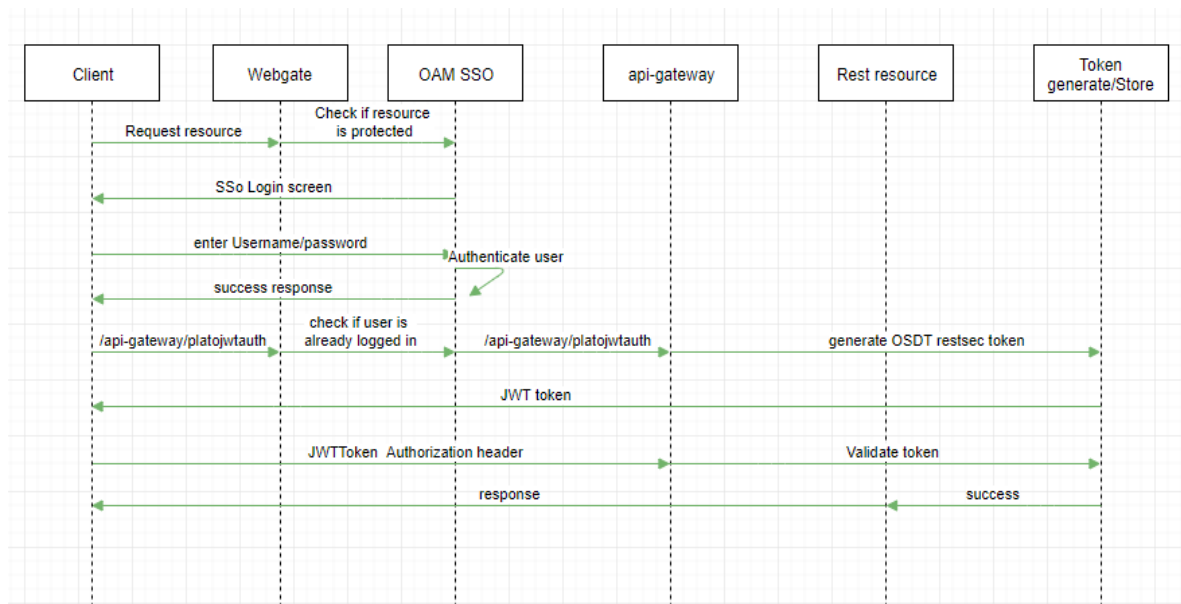
- **Refresh Token:** Users are allowed to get the new token any time before expiring the existing token.
- **Claims:** The JWT Claims Set represents a JSON object whose members are the claims conveyed by the JWT. Platform security module validates below claims during the process.

| Claim Name | Description | Mandatory | Type |
|------------|-----------------|-----------|------------|
| iss | Issuer | Yes | Registered |
| sub | Subject | Yes | Registered |
| aud | Audience | No | Registered |
| exp | Expiration Time | Yes | Registered |
| nbf | Not Before | No | Registered |
| iat | Issued At | Yes | Registered |
| jti | JWT Id | Yes | Registered |
| tid | Tenant Id | Yes | Private |

- **Token Expiry:** Platform security module invalidates the token, if the client submits after the Expiration time. In addition, token becomes invalid, if the user password changed after the token issuance.
- **Logout:**
While user calls the logout operation, platform security module clears the issued token and deletes the record from the table as well. The old token no longer will be used for any purpose.

The various security flows for the online web application are depicted below

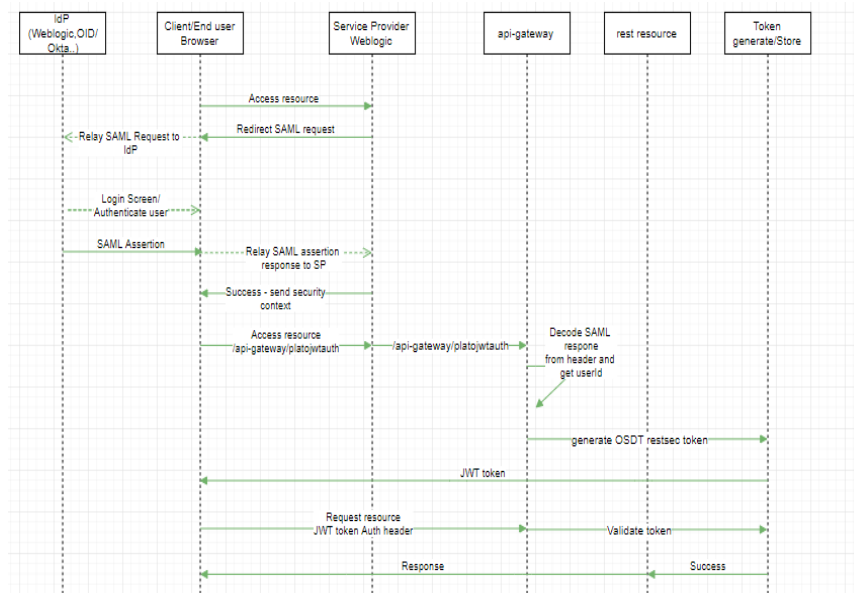
OAM Based SSO



- The online UI is protected on OAM.
- Client requests protected resource. OAM presents SSO Login screen

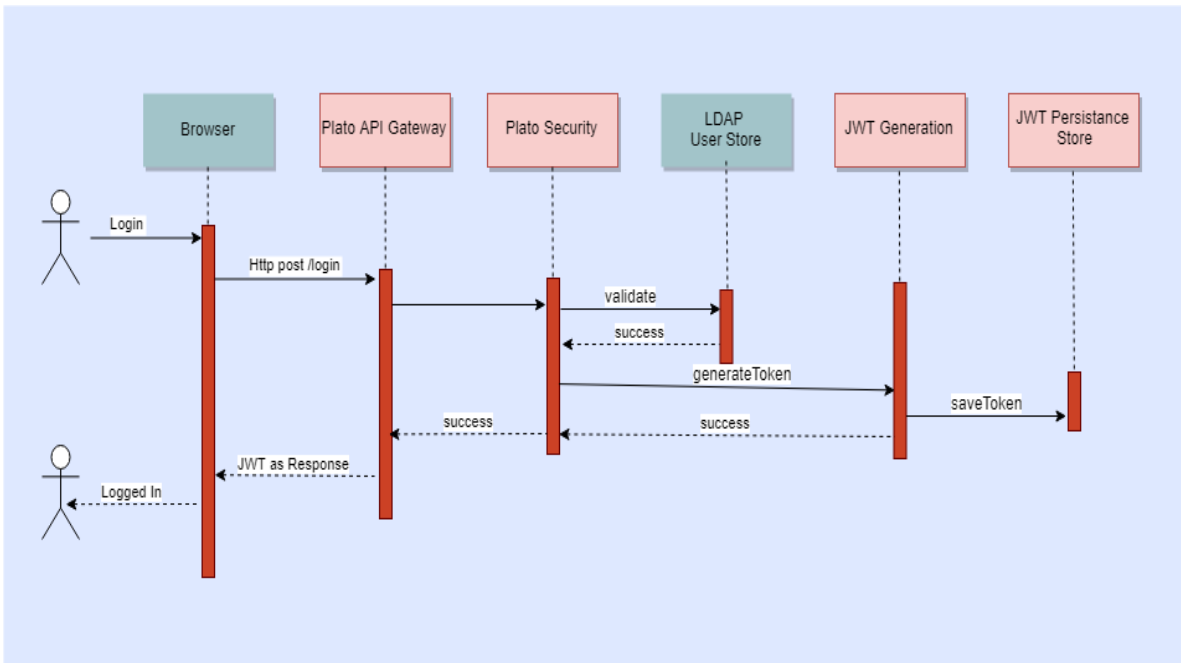
- Client enters a user id and password. In case of success, OAM sets the corresponding user profile details in the security context
- The request is routed to the Gateway which extracts the profile details from the security context
- The API Gateway creates a JWT token (Utilizing Oracle's Security Developer Toolkit part of Oracle's Platform Security Services), persists it in the Database and returns the same.
- The UI layer uses this token to maintain state and conduct subsequent invocations

SAML Authentication



- The Identity Provider is external to the Oracle Banking Liquidity Management Application (e.g. OKTA) with the Oracle Banking Liquidity Management application acting as the Service Provider
- Client requests protected resource from Oracle Banking Liquidity Management. The Idp presents a configured login screen to the user
- Client enters a user id and password. In case of success, the Idp sets the corresponding user profile details in the security context
- The request is routed to the Gateway which extracts the profile details by decoding the SAML response
- The API Gateway creates a JWT token (Utilizing Oracle's Security Developer Toolkit part of Oracle's Platform Security Services), persists it in the Database and returns the same.

LDAP Authentication



- The user is presented the standard login page for the Oracle Banking Liquidity Management application
- The user enters a user id and password. The credentials are validated against a standard LDAP store.
- If successful, the API Gateway generates a JWT token (Utilizing Oracle's Security Developer Toolkit part of Oracle's Platform Security Services), persists it in the Database and returns the same.

2.6.2 API Layer

The Oracle Banking Liquidity Management application provides an API Layer (also known as the Service API Layer) which is used by external consumers to access Oracle Banking Liquidity Management's functionality.

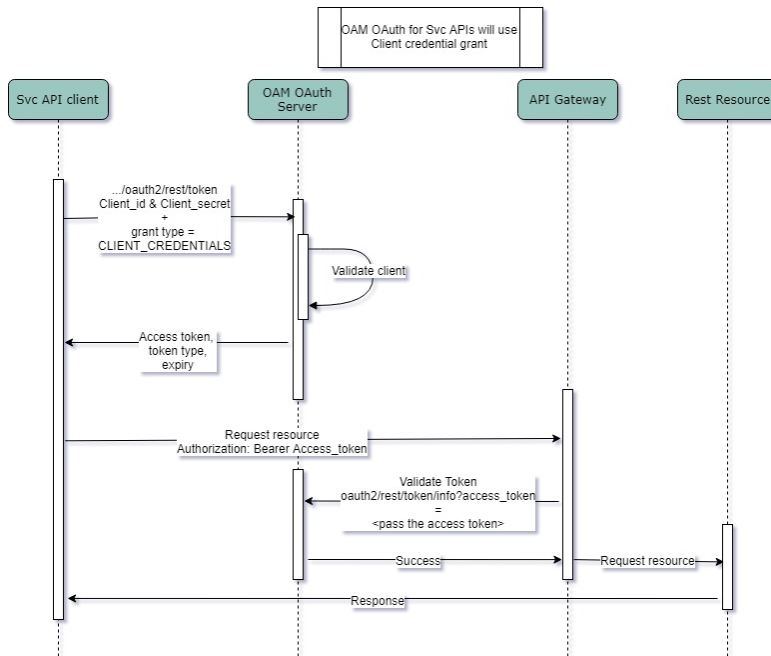
Access to this API layer is granted only via the following methods

- OAuth with OAM (Oracle Access Manager)
- OAuth without OAM

As stated before, in case the customer does not have OAM, an enterprise API Management layer should be implemented to protect the service API(s)

OAuth with OAM

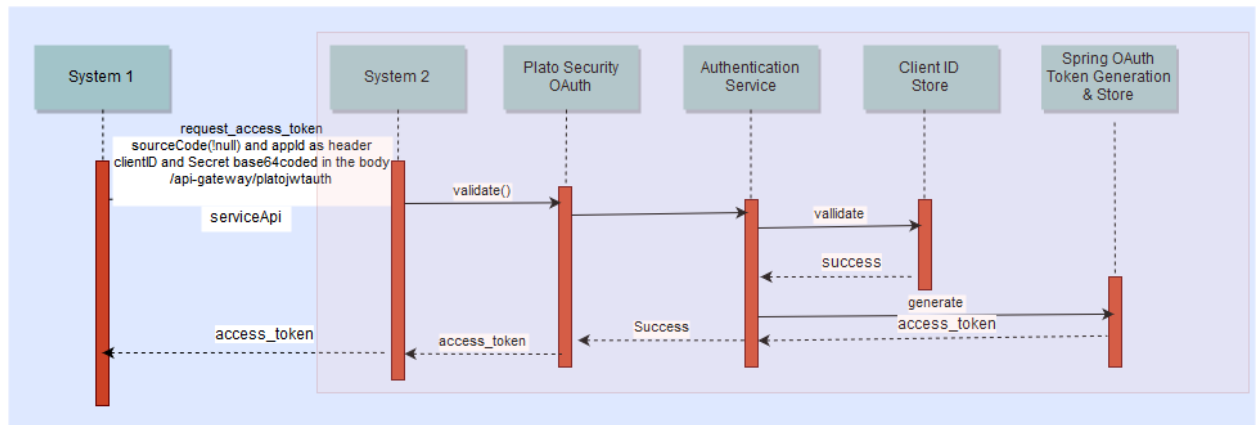
The flow is depicted below



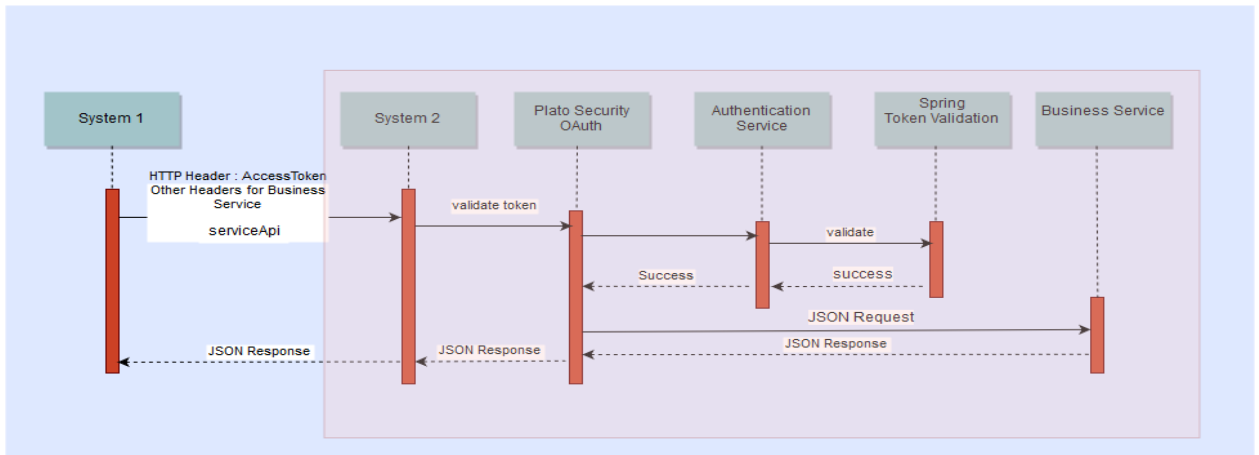
- API clients pass the client id & client secret and grant type as CLIENT CREDENTIALS, to get the access token , using the below endpoint
 - `/oauth2/rest/token`
- API Clients will pass the access token in the Authorization Header as Bearer token in their subsequent calls to access the Service APIs.
- API Gateway validates the client access token on OAM Authorization server
- If valid, it passes the request on to the Svc APIs and gets the response.
- The client can choose to get a new token (refresh) before the expiry of the current token. In case the token expires, they will pass the client Id and client secret to get a new token.

OAuth without OAM

The flow for token generation is depicted below:



The flow for accessing svc is depicted below:



- API clients pass the client id & client secret in the body and other required headers, to get the access token, using the below endpoint:
<http://<<hostname>>:<<port>>/api-gateway/platojwtauth/>
- API Clients will pass the access token in the Authorization Header as Bearer token in their subsequent calls to access the Service APIs.
- API Gateway validates the client access token on Authorization server
- If valid, it passes the request on to the Svc APIs and gets the response.
- The client can choose to get a new token (refresh) before the expiry of the current token. In case the token expires, they will pass the client Id and client secret to get a new token.
- Also, an additional facility of increasing the token is provided.

2.6.3 Two-way SSL Connection

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, must have two certificates, one for the server and the other for client.

3. Securing Oracle Banking Liquidity Management

3.1 Desktop Security

Please refer the vendor specific relevant sections for securing the Desktops Operating system. Also do refer the Browser specific security settings mentioned in the vendor specific docs.

Refer the client browser setting required for Oracle Banking Liquidity Management.

3.2 Oracle Banking Liquidity Management Controls

3.2.1 Overview

This chapter describes the various programs available within Oracle Banking Liquidity Management, to help in the maintenance of security.

Access to the system is possible only if the user logs in with a valid ID and the correct password. The activities of the users can be reviewed by the Security Officer in the Event Log and the Violation Log reports.

3.2.2 Disable Logging

It is recommended that the debug logging facility of the application be turned off, once the system is in production. This is achieved by updating the logback.xml file of the application.

The above described practice does not disable logging performed by the application in the database tier. This can be disabled by running the lockdown scripts provided. The lockdown scripts will disable logging across all modules and across all users in the system.

3.2.3 Sign-on Messages

| Message | Explanation |
|---|---|
| User Authentication Failed/Invalid Login | An incorrect user ID or password was entered. |
| User Status is Locked. Please contact your System Administrator | The user profile has been disabled 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). |

3.2.4 Authentication & Authorization

Only authenticated users can access the system. Secondly, a user should have access rights to execute a function. The user profile of a user contains the User ID and the functions to which the user has access. Oracle Banking Liquidity Management operation such as new, copy, query, unlock and so on will be enabled based on function rights available for the user. The function rights will be checked for each operation performed by the user, in Security Management Service module of Oracle Banking Liquidity Management.

3.2.5 Role Based Access Controls

- 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 Liquidity Management supports dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights.
- SMS provides an option to map multiple roles for a user in a given branch. Allowed operations are mapped to the roles and SMS authorizes the user based on it.

3.2.6 Access Controls - Branch Level

- SMS provides the branch level access through the roles provided for the user at a particular branch

3.2.7 Maker – Checker

Application supports dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights.

3.2.8 Access Enforcement

Access management in Oracle Banking Liquidity Management can be done in two steps.

- **Branch level:** In such a case the user cannot view even the menu list of the Oracle Banking Liquidity Management when he tries to login into the restricted branch. Thus, no transactions could be performed
- **Roles wise:** As described above basing on the user-roles mapping, the user can access different functions of Oracle Banking Liquidity Management. For an example, a bank clerk will have access to customer creation, account opening, term-deposits opening and liquidation screens, but he will not have access to User Creation function activity.

3.2.9 Password Management

The Oracle Banking Liquidity Management application relies on external password management and does not store any credentials. If an external LDAP is used, password management and policy rules can be set on that. If OIM/OAM is configured, password management and policy rules can be set on OIM. The IdP (Identity Provider) in case of SAML takes care of the password policies.

Certain user password related parameters should be defined at the bank level. These parameters will apply to all the users of the system. Examples of such parameters are the number of invalid login attempts after which a user-id should be disabled, the maximum and minimum length for a password, the number of previous passwords that should not be used, the interval at which the password should be changed by every user, etc.

3.2.9.1 Invalid Logins

In Oracle Banking Liquidity Management user should specify the allowable number of times an invalid login attempt is made by a user. Each user accesses the system through a unique User ID and password. While logging on to the system, if either the User Id or the Password is wrong, it amounts to an invalid login attempt.

By default, the allowable number of cumulative invalid attempts is six, and the allowable number of consecutive invalid attempts is three. These default values can be changed and specify the allowable number of attempts in each case. An allowable number for cumulative attempts are between 6 and 99, and for consecutive (successive) attempts are between 3 and 5.

When authentication of credentials is unsuccessful due to an incorrect user ID, then the user id will not be logged in the audit logs. In case the user id is correct and the password is wrong, the attempt is logged in the audit log and the successive and cumulative failure count is incremented. When the user id and password are correct, this is logged into the audit logs.

3.2.9.2 Specifying Parameter

Dormancy Days

Oracle Banking Liquidity Management allows you to automatically disable the profile of all the users who have not logged into the system for a pre-defined period of time. A user ID is considered dormant if the difference between the last login date and the current date is equal to or greater than the number of 'Dormancy Days' that has been specified. This is reckoned in calendar days i.e. inclusive of holidays. All dormant users (whose home branch is same as the current branch) are disabled during the end of day run at the current branch.

4. General Information

4.1 Cryptography

Oracle Banking Liquidity Management uses cryptography to protect the sensitive data.

For encryption, AES, which is considered to be of gold standard, is used. It produces a key size of 256 bits when it comes to symmetric key encryption.

4.2 Security patch

Security patches needs to be applied whenever it's available for the applicable product version.

4.3 Oracle Database Security Suggestions

Access Control

Database Vault (DV) Provides enterprises with protection from the insider threats and in advantage leakage of sensitive application data. Access to application data by users and administrators is controlled using DV realms, command rules and multi factor authorization. DV also address Access privilege by separating responsibilities.

Data Protection

Advance Security provides the most advance encryption capabilities for protecting sensitive information without requiring any change to the application. TDE is native database solution that is completely transparent to the existing applications.

Advance Security also provides strong protection for data in transit by using network encryption capabilities. Features like Easy to deploy, Ensure secure by default to accept communication from client using encryption, Network encryption using SSL/TLS.

Monitoring and Compliance

Audit Vault (AV) transparently collects and consolidate audit data from multiple databases across the enterprise, does provide valuable insight into who did what with which data & when including privilege users. The integrity of the audit data is ensured using controls including DV, Advance Security. Access to AV data is strictly controlled. It also does provide graphical summaries of activity causing alerts, in addition database audit setting are centrally managed and monitored.

4.4 Oracle Software Security Assurance - Standards

Every acquired organization must complete the Mergers and Acquisitions (M&A) Security Integration process. The issues identified during this review must be addressed according to the agreed upon M&A remediation plan. The acquired organization must complete SPOC assignments and plan integration of OSSA methodologies and processes into its SDLC.

4.5 References

4.5.1 Datacenter Security considerations

Please refer to the following links to understand Datacenter Security considerations

http://docs.oracle.com/cd/B14099_19/core.1012/b13999/rectop.htm

4.5.2 Database Security considerations

Please refer the below links to understand more on Database Security considerations recommended to be followed

<http://www.oracle.com/us/products/database/security/overview/index.html>

<http://www.oracle.com/technetwork/database/security/twp-security-checklist-database-1-132870.pdf>

4.5.3 Security recommendations / practices followed for Database Environment

Please refer the below mentioned links to understand more on Security recommendations / practices followed for Database Environment

http://docs.oracle.com/cd/B28359_01/network.111/b28531/guidelines.htm

4.5.4 Common security considerations

Please refer below links to understand some of the common security considerations to be followed

http://docs.oracle.com/cd/B14099_19/core.1012/b28654.pdf

http://docs.oracle.com/cd/E14899_01/doc.9102/e14761/tuningforappserver.htm

http://docs.oracle.com/cd/E13222_01/wls/docs81b/lockdown/practices.html

http://docs.oracle.com/cd/E23943_01/web.1111/e14529/security.htm

<http://www.oracle.com/us/solutions/oos/weblogic-server/overview/index.html>



Security 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>

Copyright © 2018, 2021, Oracle and/or its affiliates. All rights reserved.

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. 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 failsafe, 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.

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.

This software or hardware and documentation may provide access to or information on 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. 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.