Oracle® Banking Branch

Security Guide

14.7.0.0.0

F75183-01

November 2022

Oracle Banking Branch Security Guide, 14.7.0.0.0

F75183-01

Contents

Preface

This guide provides security-related usage and configuration recommendations for Oracle Banking Branch. It may outline procedures required to implement or secure certain features. This guide is not for general-purpose configuration.

This topic contains the following subtopics:

Audience

This guide is primarily intended for IT department or administrators deploying Oracle Banking Branch and third party or vendor software. Some information that may be relevant to IT decision-makers and users of the application are also included.

> **Note:**
>
> Readers are assumed to possess the basic operating system, network, and system administration skills with an awareness of vendor/third-party software and knowledge of the Oracle Banking Branch application.

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 or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Scope

The scope of this guide is as follows:

Table Scope

Scope	Description
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 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 configuration recommendations. It does not imply the suitability or unsuitability of any product for any particular situation, which entails a risk decision.
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.
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.

Acronyms and Abbreviations

The following acronyms and abbreviations are used in this guide:

Table Acronyms and Abbreviations

Acronym/Abbreviation	Description
AES	Advanced Encryption Standard
API	Application Programming Interface
DV	Database Vault
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IdP	Identity Provider
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
JWE	JSON Web Encryption
JWS	JSON Web Signature
JWT	JSON Web Token
LDAP	Lightweight Directory Access Protocol
OAM	Oracle Access Manager
M&A	Mergers and Acquisitions
OAuth	Open Authentication
OIM	Oracle Identity Management
OSSA	Oracle Software Security Assurance
SAML	Security Assertion Markup Language

Table (Cont.) Acronyms and Abbreviations

Acronym/Abbreviation	Description
SDLC	Software Development Lifecycle
SMS	Security Management System
SPOC	Single Point of Contact
SQL	Structured Query Language
SSO	Single Sign-On
SSL	Secure Sockets Layer
TDE	Transparent Data Encryption
TLS	Transport Layer Security
UI	User Interface

List of Topics

This guide is organized into the following topics:

Table List of Topics

Topics	Description
Prerequisite	This topic provides information about prerequisites.
Securing Oracle Banking Branch	This topic provides information about securing Oracle Banking Branch.
General Information	This topic provides general information related to security.

1 Prerequisite

The prerequisites are as follows:

Table 1-1 Prerequisites

Prerequisite	Description/Reference
Operating Environment Security	Refer to the vendor-specific documentation for making the environment more safe and secure.
Network Security	Refer to the vendor-specific documentation for making the environment more safe and secure.
Oracle Database Security	For information on Oracle Database Security, refer to Oracle Database Security.
Application Server Security	For information on Application Server Security, refer to Application Server Security.
Choice of the SSL Cipher Suite	For information on SSL Cipher Suite, refer to SSL Support.
Securing Oracle Banking Branch	For information on Securing Oracle Banking Branch, refer to Securing Oracle Banking Branch Product.

- Oracle Database Security
  The security recommendations are provided for the database used for Oracle Banking Branch products.
- Application Server Security
  The application server of the Oracle Banking Branch needs to be secured.
- SSL Support
  This topic provides the information for SSL Support.
- Securing Oracle Banking Branch Product
  You need to secure the Oracle Banking Branch application through securing the Online Web Application and API Layer exposed to external consumers.

1.1 Oracle Database Security

The security recommendations are provided for the database used for Oracle Banking Branch products.

Refer to the Oracle Database Security specification document for making the environment more safe and secure.

Table 1-2 Recommended Configuration

Configuration	Value	Purpose
`Init.ora`	`REMOTE_OS_AUTHENT=FALSE`	Authentication
`Init.ora`	`TRACE_FILES_PUBLIC=FALSE`	Authorization

Table 1-2 (Cont.) Recommended Configuration

Configuration	Value	Purpose
`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

> **Note:**
>
> To audit the events, log in through *sqlplus* as *SYSTEM* and issue the commands

1.2 Application Server Security

The application server of the Oracle Banking Branch needs to be secured.

Refer to the Oracle Web Logic Security specification document for making the environment more safe and secure.

Oracle Banking Branch products supports the following authentication schemes for the online web application:

- Standard LDAP Directory (e.g. OUD/AD/Embedded Weblogic)
- 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 Branch products solution supports the following authentication scheme for the API layer:

- OAuth (CLIENT CREDENTIALS) with OAM
- OAuth (CLIENT CREDENTIALS) without OAM

In case the customer does not have OAM, they can use OAUTH without OAM or it is expected that the customer has an enterprise API Management Layer that protects Oracle Banking Branch products's API layer with the same controls (i.e. OAuth).

Support for Secure Transformation of Data (SSL)

The Oracle Banking Branch products are 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.

1.3 SSL Support

This topic provides the information for SSL Support.

This topic contains the following subtopics:

- SSL Setup
  This topic provides the information for SSL Setup.
- Choice of the SSL cipher suite
  This topic describes about choice of the SSL cipher suite.
- Product configurations for SSL
  This topic provides the information for Product configurations for SSL.

SSL Setup

Refer to SSL Configurations Setup Guide for the setup details.

Choice of the SSL cipher suite

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

The default configuration of the 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_RSA_WITH_AES_256_CBC_SHA</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, which 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).

Product Configurations for SSL

Refer to **Section 12.4** in Oracle Banking Microservices Platform Foundation Installation Guide.

1.4 Securing Oracle Banking Branch Product

You need to secure the Oracle Banking Branch application through securing the Online Web Application and API Layer exposed to external consumers.

This topic contains the following subtopics:

- Online Web Application
  The Online Web Application can be secured through the industry standards and policies.
- API Security
  This topic describes about API Security.
- Two-way SSL Connection
  A two-way SSL is used when the server needs to authenticate the client.

1.4.1 Online Web Application

The Online Web Application can be secured through the industry standards and policies.

Authentication and authorization to requests to access the Online Web Application (appshell) are controlled using the below industry standard approaches:

- Standard LDAP Directory authentication
- SSO with OAM
- SSO with other External SSO Agents
- SAML with the Oracle Banking Branch products application acting as the service provider

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

JWTs are an open, industry-standard *RFC 7519* method for representing claims securely between two parties. 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.

Table 1-3 Features

Feature	Description
No Session to Manage (stateless)	The JWT is a self-contained token that has authentication information, expire time information, and other user-defined claims digitally signed.
Portable	A single token can be used with multiple backends.
No Cookies Required	It is 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 be easily separated into its own server.

In addition, the following policies are followed for JWT:

Table 1-4 Policies

Policy	Description
Token Store	To increase security and better usability, every authentication/refresh request is secured by a 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 the *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. The platform security module validates the claims mentioned in the Claims during the process.

Table 1-4 (Cont.) Policies

Policy	Description
Token Expiry	Platform security module invalidates the token if the client submits after the Expiration Time.
Logout	While the user calls the logout operation, the 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.

Table 1-5 Claims

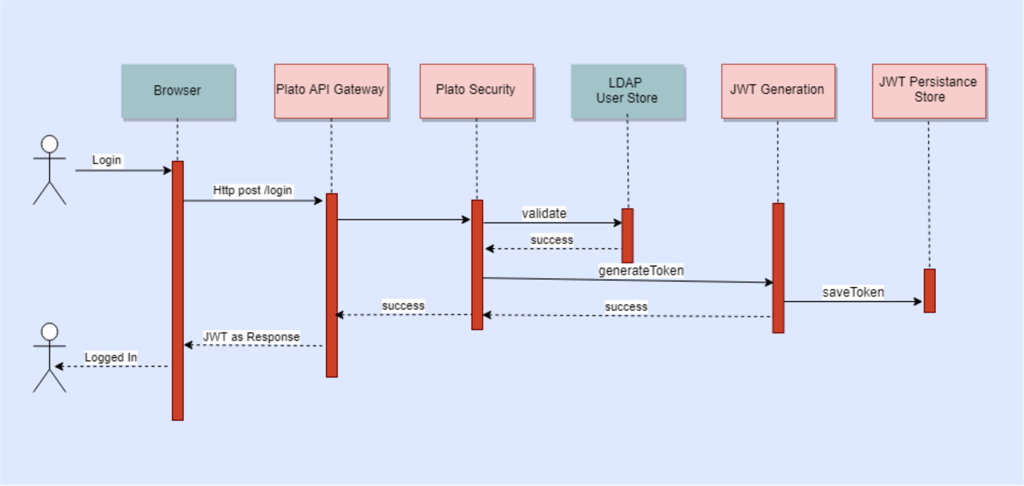
Claim Name	Description	Mandatory	Type
`Iss`	Issuer	Yes	Registered
`Sub`	Subject	Yes	Registered
`Aud`	Audience	No	Registered
`Exp`	Expiration Time	Yes	Registered
`Nbf`	Not Before	No	Registered
`Iat`	Issued At	Yes	Registered
`Jti`	JWT Id	Yes	Registered
`Tid`	Tenant Id	Yes	Private

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

LDAP Authentication

The flow is depicted below:

Figure 1-1 LDAP Authentication



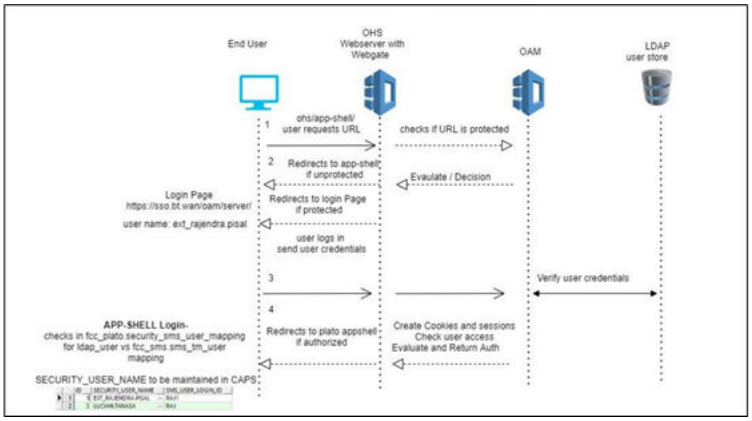
The security flow is as follows:

- The user has presented the standard login page for the Oracle Banking Branch products.
- 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.

OAM Based SSO

The flow is depicted below:

Figure 1-2 OAM Based SSO



The security flow is as follows:

- The online UI is protected on OAM.
- The client requests protected resource. OAM presents the 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 in the Database, and returns the same.
- The UI layer uses this token to maintain the state and conduct subsequent invocations.

Product configuration:

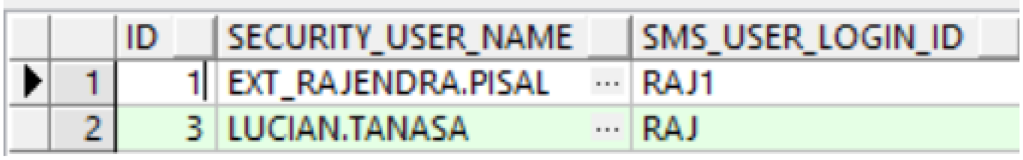
The following parameters need to be set to enable a successful integration with OAM as SSO in Oracle Banking Branch products:

PLATO.SECURITY_CONFIG table
USER_HEADER_ATTRIBUTE_KEY,IS_SSO_CONFIGURED,USER_MAPPING_REQUIRED to be set as true.

ID	KEY	VALUE
1	USER_HEADER_ATTRIBUTE_KEY	userId

ID	KEY	VALUE
2	USER_HEADER_ATTRIBUTE_REQUIRED	Y
3	IS_SSO_CONFIGURED	true
4	USER_MAPPING_REQUIRED	true

Figure 1-3 PLATO.SECURITY_SMS_USER_MAPPING table

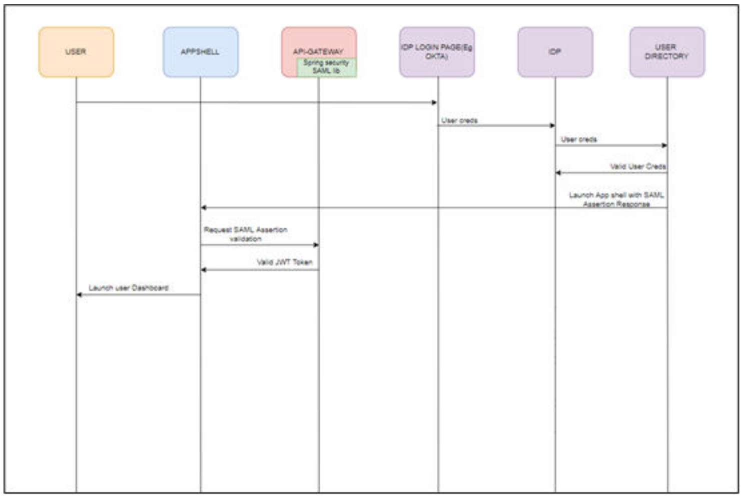


		ID	SECURITY_USER_NAME	SMS_USER_LOGIN_ID
▶	1	1	EXT_RAJENDRA.PISAL ···	RAJ1
	2	3	LUCIAN.TANASA ···	RAJ

SAML Authentication

The flow is depicted below:

Figure 1-4 IDP Initiated SAML Authentication



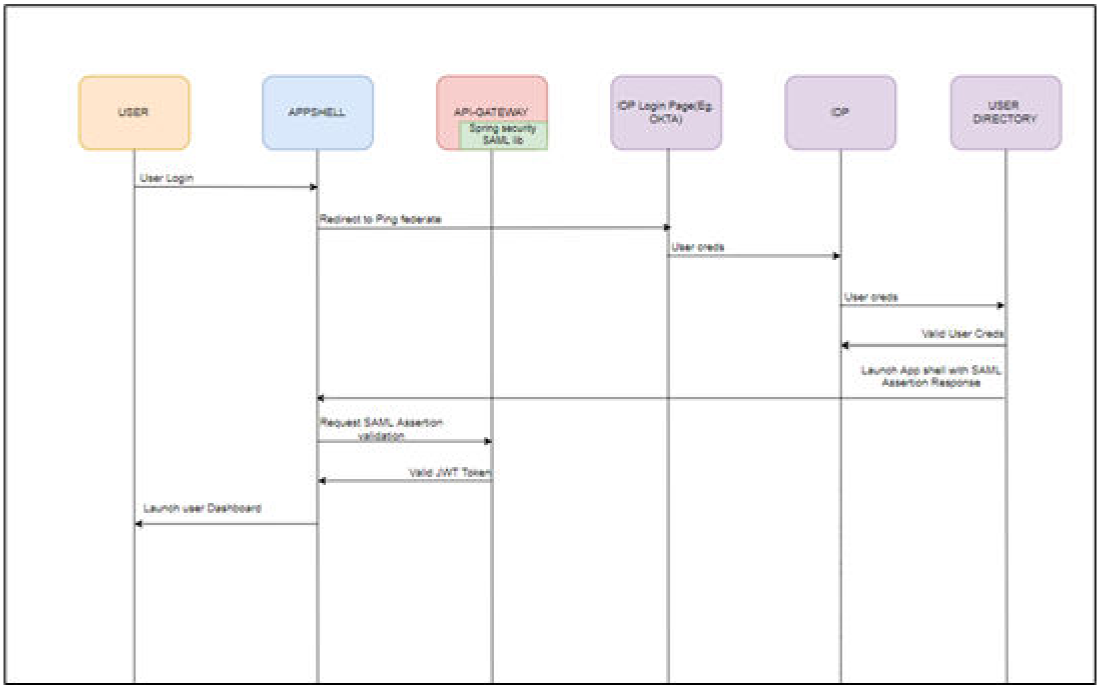
The security flow is as follows:

- The Identity Provider (IdP) is external to the Oracle Banking Branch Application (For example, OKTA) with the Oracle Banking Branch application acting as the Service Provider.
- Client requests protected resource from Oracle Banking Branch. 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 in the Database, and returns the same.
- It is possible to configure an external service to do the SAML Verification instead of the API Gateway using the EXTERNAL_SSO_VALIDATION_URL parameter in the SECURITY_CONFIG table in PLATO-SECURITY schema.

SP Initiated SAML Authentication

Figure 1-5   SP Initiated SAML Authentication



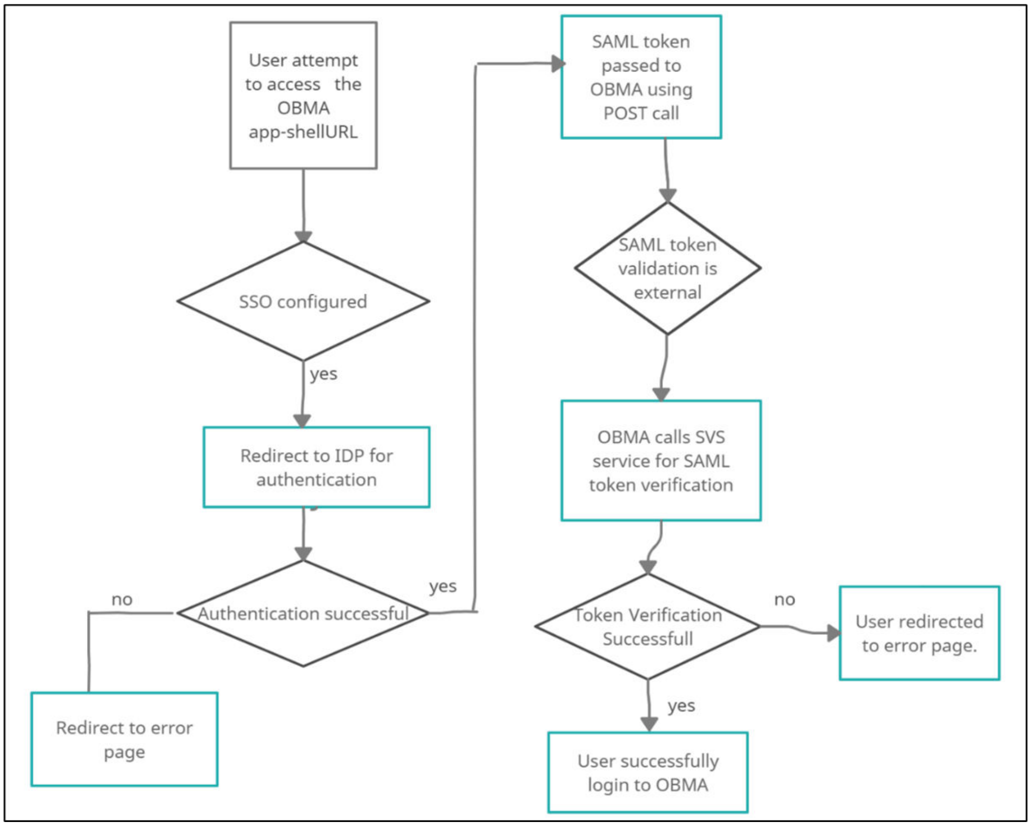
The security flow is as follows:

- The user initiates a call to the Oracle Banking Branch application and is redirected to the federate login page of the bank.
- The Identity Provider is external to the Oracle Banking Branch products (example OKTA) with the Oracle Banking Branch products acting as the Service Provider.
- 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 in the Database, and returns the same.

SAML SSO Implementation

It is possible to configure an external service to do the SAML Verification instead of the API Gateway.

Figure 1-6 SAML SSO Implementation


User attempt to access the OBMA app-shellURL
SSO configured
yes
Redirect to IDP for authentication
Authentication successful
no
yes
Redirect to error page
SAML token passed to OBMA using POST call
SAML token validation is external
OBMA calls SVS service for SAML token verification
Token Verification Successfull
no
User redirected to error page.
yes
User successfully login to OBMA

Steps to achieve SSO-SAML Authentication are as follows:

- Bank user will try to access the Oracle Banking Branch app-shell URL.
- Oracle Banking Branch will check if the IS_SSO_CONFIGURED parameter is set to true in the SECURITY_CONFIG table.
- If the IS_SSO_CONFIGURED parameter is true the user will be redirected to the IDP for authentication.
- On successful authentication, IDP will generate the SAML token and pass the token to the Oracle Banking Branch assertion consumer service URL in the body of POST method through EXTERNAL_SSO_KEY parameter.
- Oracle Banking Branch will receive the token and check if the SSO_SERVICE_PROVIDER is set to EXTERNAL in the SECURITY_CONFIG table.
- If SSO_SERVICE_PROVIDER is EXTERNAL, Oracle Banking Branch would make a HTTP Post call to SVS using the EXTERNAL_SSO_VALIDATION_URL configured in the SECURITY_CONFIG table for SAML token validation. Oracle Banking Branch will pass the SAML token through EXTERNAL_SSO_TOKEN_KEY parameter in the body of the POST to SVS.
- SVS will return an XML response with IsValid tag as TRUE or FALSE. If the tag value is TRUE, Oracle Banking Branch would generate a JWT token using the user id from the <subject> </subject> tag of the SVS response and allow the user to login.

- In case of failure, Oracle Banking Branch would give login error to the user.

Product Configurations Required:

The following parameters needs to be configured in the SECURITY_CONFIG table in the PLATO-SECURITY schema to enable SAML SSO.

Table 1-6 SECURITY_CONFIG

Key	Value
IS_SSO_CONFIGURED	True
JWT_EXP_SECONDS	JWT expiry time
JWT_ALGORITHM	HS512
EXTERNAL_SSO_VALIDATION_URL	SVS URL
EXTERNAL_SSO_KEY	Parameter in which the SAML token will be passed to Oracle Banking Microservices Architecture from IDP after user authentication.
SSO_SERVICE_PROVIDER	EXTERNAL
EXTERNAL_SSO_TOKEN_KEY	Parameter in which the SAML token will be passed to SVS URL for token validation.
HEADERS	Request headers for making HTTP call to SVS URL

FCUBS integration with Oracle Banking Branch as SSO Provider

Refer to Launching Oracle Banking Branch from UBS section in the Installation Guide.

1.4.2 API Security

This topic describes about API Security.

Refer to API Security for the detailed explanation.

1.4.3 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. To establish a two-way SSL connection, the user must have certificates for:

- Server
- Client

The 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 Securing Oracle Banking Branch

You can use various programs available within Oracle Banking Branch to help in the maintenance of security and secure the desktop operating system.

Desktop Security

Refer to the vendor-specific relevant sections for securing the desktop operating system. In addition, refer to the browser-specific security settings mentioned in the vendor-specific docs.

Refer to the client browser setting required for Oracle Banking Branch.

Oracle Banking Branch Controls

This section describes the various programs available within Oracle Banking Branch 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.

Table 2-1 Oracle Banking Branch Products Controls

Control	Description
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 all users in the system.
Sign-on Messages	• **Message** - User Authentication Failed/Invalid Login **Explanation** - An incorrect user ID or password was entered. • **Message** - User Status is Locked. Please contact your System Administrator **Explanation** - 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 number of login failures (configured for the system).
Authentication and 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 Branch operations 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 the Security Management Service module of the Oracle Banking Branch.

Table 2-1 (Cont.) Oracle Banking Branch Products Controls

Control	Description
Role-Based Access Controls	The role-based access controls are: • Application level access has been implemented via the Security Management System (SMS) module. • SMS supports “ROLE BASED” access of screens and different types of operations. • Oracle Banking Branch 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.
Access Controls - Branch Level	SMS provides branch-level access through the roles provided for the user at a particular branch.
Maker – Checker	The application supports dual control methodology, wherein every operation performed has to be authorized by another user with the requisite rights.
Access Enforcement	Access management in Oracle Banking Branch can be done in two steps: • **Branch level:** In such a case the user cannot view even the menu list of the Oracle Banking Branch when the user tries to log in into the restricted branch. Thus, no transactions could be performed. • **Roles wise:** As described above based on the user-roles mapping, the user can access different functions of the Oracle Banking Branch. For example, a bank clerk will have access to customer creation, account opening, term-deposits opening, and liquidation screens, but will not have access to the User Creation function activity.
Password Management	The Oracle Banking Branch 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 (For example, in WebLogic Embedded-LDAP, the user and password rules can be configured via the admin console of the WebLogic). If OIM/OAM is configured, password management and policy rules can be set on OIM. The Identity Provider (IdP) in case of SAML takes care of the password policies. Certain user password related parameters should be defined at the system 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. **Note:** For more information on Password Management, refer to Password Policies in this guide.

Password Policies

To enable password validation criteria, there is a flag given in the SECURITY_CONFIG table is PASSWORD_VALIDATION_FLAG which has to be set as Y.

Table 2-2   SECURITY_PASSWORD_VAL_CONFIG

Property	Value	Description
MIN_PSWD_LEN	Any integer	Minimum password length required
MAX_PSWD_LEN	Any integer	Maximum password length allowed
MIN_PSWD_AGE	Any integer	Not used currently
MAX_PSWD_AGE	Any integer	Not used currently
FLAG_UPPER_CHAR	Y/N	Y- Uppercase characters required
NUM_MAND_UPPER	Integer	Minimum uppercase characters required. Checked only if FLAG_UPPER_CHAR is set to Y
FLAG_LOWER_CHAR	Y/N	Y- Lowercase characters required
NUM_MAND_LOWER	Integer	Minimum lowercase characters required. Checked only if FLAG_LOWER_CHAR is set to Y
FLAG_SPECIAL_CHAR	Y/N	Y- Special characters required
NUM_MAND_SPECIAL	Integer	Minimum special characters required. Checked only if FLAG_SPECIAL_CHAR is set to Y
FLAG_NUMERIC_CHAR	Y/N	Y- Numeric characters required
NUM_MAND_NUMERIC	Integer	Minimum numeric characters required. Checked only if FLAG_ NUMERIC_CHAR is set to Y

3

General Information

The general information about security includes standards, patches, suggestions, and references.

Cryptography

Oracle Banking Branch uses cryptography to protect sensitive data.

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

Security Patch

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

Oracle Database Security Suggestions

Table 3-1 Oracle Database Security Suggestions

Security Suggestion	Description
Access Control	Database Vault (DV) Provides enterprises with protection from 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 addresses Access privilege by separating responsibilities.
Data Protection	Advance Security provides the most advanced encryption capabilities for protecting sensitive information without requiring any change to the application. TDE is a 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 the client using encryption, Network encryption using SSL/TLS.
Monitoring and Compliance	Audit Vault (AV) transparently collects and consolidates audit data from multiple databases across the enterprise, does provide valuable insight into who did what with which data and when including privileged 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 the activity causing alerts, in addition, database audit settings are centrally managed and monitored.

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 the integration of OSSA methodologies and processes into its SDLC.

4 References

For more information, see these Oracle resources:

Table 4-1 References

Reference	Link/URL
Datacenter Security Considerations	Understanding WebLogic Server Deployment (oracle.com)
Database Security Considerations	• https://www.oracle.com/security/database-security/ • https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/database-security-guide.pdf
Security Recommendations / Practices Followed for Database Environment	• https://docs.oracle.com/en/database/oracle/oracle-database/19/security.html • https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/index.html
Common Security Considerations	• https://www.oracle.com/database/technologies/high-availability/fusion-middleware-maa.html • https://www.oracle.com/a/tech/docs/tip4847-maa-best-practices-for-database.pdf • https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/perfm/basics.html#GUID-178B107B-10E9-4563-BCA4-E06E14F5D3FF • https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/lockd/securing-production-environment-oracle-weblogic-server.pdf • https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/secmg/index.html • https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-server/12.2.1.4/index.html