

**Oracle FLEXCUBE Core Banking
Security Practices Guide**

Release 11.9.0.0.0

Part No. F30993_01

[May] [2020]



Revision History

Date	Revision No	Reviewer Name	Modified By	Description of Change
11/02/2020	1.0	Yogesh Varshney	Sambhaji Limkar	Initial Version

Table of Contents

1. INTRODUCTION	1-1
1.1 WARNINGS	1-1
1.2 INTRODUCTION	1-1
1.2.1 Purpose	1-1
1.2.2 Audience	1-1
1.2.3 Scope	1-1
1.2.4 General Principles	1-2
1.2.5 Glossary of Icons	1-3
1.2.6 Comments	1-4
2. PRE INSTALLATION STEPS.....	2-5
2.1 SSL ENABLING ON FLEXCUBE	2-5
2.1.1 Introduction	2-5
2.1.2 Install Certificate Services	2-5
3. INSTALLATION.....	3-7
3.1 DATA CENTER PRACTICES.....	3-7
3.1.1 Overview	3-7
3.1.2 Physical System Security	3-7
3.1.3 Minimize the Server Footprint	3-7
3.1.4 Operating System Users and Groups	3-7
3.1.5 Restrict File System Access.....	3-8
3.1.6 Network Perimeter Protection (For Production Mode Only).....	3-8
3.1.7 Network Service Protection (For Production Mode Only)	3-8
3.1.8 Usage of Protected Ports (For Production Mode Only).....	3-8
3.1.9 Installation of Software in Production Mode (For Production Mode Only).....	3-8
3.1.10 Software Updates and Patches	3-9
3.1.11 Usage of Security Appliances and Software	3-9
3.1.12 Configure Security Auditing(For Production Mode Only)	3-9
3.1.13 Separation of concerns	3-9
3.1.14 Separation of concerns (For Production Mode Only)	3-9
3.2 ORACLE DATABASE SECURITY(FOR PRODUCTION MODE ONLY)	3-10
3.2.1 Overview	3-10
3.2.2 Hardening	3-10
3.2.3 Authentication.....	3-10
3.2.4 Authorization	3-10
3.2.5 Secure Database Backups.....	3-11
3.2.6 Separation of Roles	3-11
3.2.7 Advanced Security.....	3-11
3.2.8 Audit.....	3-11
3.3 DATABASE OPERATING ENVIRONMENT SECURITY (FOR PRODUCTION MODE ONLY)	3-14
3.3.1 Overview	3-14
3.3.2 Hardening	3-14
3.3.3 Authentication.....	3-15
3.3.4 Authorization	3-15
3.3.5 Maintenance	3-16
3.4 DATABASE PRACTICES FOR RISK MITIGATION IN PRODUCTION(FOR PRODUCTION MODE ONLY).....	3-16
3.4.1 Production Database – Access Prevention	3-16
3.4.2 Production Database – Data Protection	3-17

3.4.3	<i>Production Database – Release Management</i>	3-18
3.5	APPLICATION SERVER SECURITY	3-19
3.5.1	<i>Overview</i>	3-19
3.5.2	<i>Installation of Oracle Weblogic Server</i>	3-19
3.5.3	<i>Securing the Weblogic Server Installation(For Production Mode Only)</i>	3-19
3.5.4	<i>Securing the Weblogic Security Service(For Production Mode Only)</i>	3-21
4.	POST INSTALLATION	4-23
4.1	DESKTOP SECURITY	4-23
4.1.1	<i>Hardening the Browser</i>	4-23
4.2	BRANCH ENVIRONMENT SECURITY	4-24
4.2.1	<i>OHS Configuration</i>	4-24
4.2.2	<i>Security by Default Issues</i>	4-24
4.2.3	<i>OHS Hardening</i>	4-24
4.3	ORACLE FLEXCUBE CORE BANKING CONTROLS.....	4-27
4.3.1	<i>Overview</i>	4-27
4.3.2	<i>Disable Logging</i>	4-27
4.3.3	<i>Display/Print User Profile</i>	4-27
4.3.4	<i>Clear User Profile</i>	4-28
4.3.5	<i>Change User Password</i>	4-28
4.3.6	<i>Default password Policy</i>	4-28
4.3.7	<i>List of Logged-in Users</i>	4-29
4.3.8	<i>Default Login Type Configurations</i>	4-29
4.3.9	<i>Reporting of Security Violations</i>	4-29
4.3.10	<i>Audit Reports</i>	4-29
4.3.11	<i>Session Timeouts</i>	4-30
4.3.12	<i>Terminal Lockouts</i>	4-30
4.3.13	<i>Enable Request Encryption from Client to OHS Server</i>	4-30
5.	GENERAL	5-1
5.1	OVERVIEW.....	5-1
5.2	VALIDATION	5-4
5.2.1	<i>Client Level Validations:</i>	5-4
5.2.2	<i>Host Level Validations:</i>	5-4
5.3	SESSION MANAGEMENT	5-4
5.3.1	<i>Session Storage</i>	5-5
5.3.2	<i>Session logging</i>	5-5
5.4	PASSWORD MANAGEMENT	5-5
5.4.1	<i>Password Protection</i>	5-5
5.5	EXCEPTION/ERROR HANDLING.....	5-5
5.6	LOGGING	5-6
5.7	ADDITIONAL ORACLE PRODUCTS OFFERING SECURITY(<i>FOR PRODUCTION MODE ONLY</i>).....	5-6
5.8	REFERENCES.....	5-8
5.8.1	<i>Datacenter Security Considerations</i>	5-8
5.8.2	<i>Database Security Considerations</i>	5-8
5.8.3	<i>Security recommendations / practices followed for Database Environment</i>	5-8
5.8.4	<i>Common security considerations</i>	5-9

1. Introduction

1.1 Warnings

- As with any other information system, do not attempt to implement any of the recommendations in this guide without first testing in a non-production environment.
- This document is only a guide containing recommendations. It is not meant to replace well-structured policy or sound judgment. Furthermore this guide does not address site specific optimization, configuration concerns.
- Care must be taken when implementing this guide to address local operational and policy concerns.
- The configuration settings described in the document apply only to the limited scope, version etc. The guidance may not translate gracefully to other systems or versions, although applying vendor updates is always recommended.
- For further details on each suggested setting always refer the vendor specific sites.

1.2 Introduction

1.2.1 Purpose

This document provides security-related usage and configuration recommendations for FLEXCUBE 11.9.0.0.0. This guide may outline procedures required to implement and secure certain features, but it is also not a general-purpose configuration manual.

1.2.2 Audience

This guide is primarily intended for IT department or administrators deploying FLEXCUBE 11.9.0.0.0 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 operation system, network and system administration skills with awareness of vendor/third-party software's and knowledge of FLEXCUBE 11.9.0.0.0 application.

1.2.3 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 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.

Recommendation for Production Environment

The Section marked with “For Production Mode” are configuration recommendations that are not feasible to check on Dev Environments but are recommended to be implemented for Production environment. These recommendations need to be tested in non-production environment before Deployments.

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.

1.2.4 General Principles

The following general principles motivate much of the advice in this guide and should also influence any configuration decisions that are not explicitly addressed.

Encrypt Transmitted Data Whenever Possible

Data transmitted over a network, whether via wire or wirelessly, is susceptible to passive monitoring. Whenever practical mechanisms exist for encrypting this data-in-transit, they should be applied. Even if data is expected to be transmitted only over a local network, it should still be encrypted if possible. Encrypting authentication data, such as passwords, is particularly important.

Encrypt Stored Data Whenever Possible

Data on mobile devices or system is particularly susceptible to compromise due to loss of physical control. Whenever practical solutions exist, they should be employed to protect this data.

Minimize Software to Minimize Vulnerability

The easiest and simplest way to avoid the vulnerabilities in a particular piece of software is to avoid installing the unwanted software altogether.

Leverage Security to Minimize Vulnerability




















Security features should be effectively used to improve a system's resistance to attacks. These features can improve a system's robustness against attack for only the cost of a little effort spent doing configuration.

Grant Least Privilege

Grant the least privilege necessary for users to perform tasks. The more privileges (or capabilities) that a user has, the more opportunities he or she will have to enable the compromise of the system (and be a victim of such a compromise). Similarly, it is possible to restrict the installation of third party apps, and this may be the right balance between security and functionality for some environments.

1.2.5 Glossary of Icons

This User Manual may refer to all or some of the following icons:

Icons	Function
	New
	Copy
	Save
	Delete
	Unlock
	Print
	Close
	Re-open
	Reverse
	Template
	Roll-over
	Hold
	Authorize
	Liquidate
	Exit
	Sign-off
	Help
	Add row
	Delete row

Refer the Procedures User Manual for further details about the icons.

1.2.6 Comments

Please provide comments concerning the improvement of this solution through support channel. About *Oracle Software Security assurance* refer below link:

<http://www.oracle.com/us/support/assurance/overview/index.html>

2. Pre Installation Steps

2.1 SSL ENABLING on FLEXCUBE

2.1.1 Introduction

HTTP communications are fine for the average Web server, which just contains informational pages. But if you're thinking about running an application that requires secure transactions, you need to be able to encrypt communications between your Web server and its clients. The most common means is by the use of Secure Sockets Layer (SSL), which uses public key cryptography to protect confidential user information that is transmitted across the Web. Below we will understand how to implement SSL on HTTP Server. By HTTP Server SSL, we imply SSL security on connections established between the client workstations and the HTTP Server.

2.1.2 Install Certificate Services

The basic requirement for signing certificates is that there should be a machine with Certificate services installed. The service is used to issue and manage certificates for a Public Key Infrastructure (PKI).

Prerequisites

Before you install Certificate Services, you should be aware of the system requirements. These will vary depending on the type of Certificate Authority (CA) you are installing.

An Enterprise Root CA requires:

- Active Directory
- Domain Name Service (DNS)
- Transmission Control Protocol / Internet Protocol (TCP/IP).
- Linux server
- Logged in account should be member of Enterprise Admin group.
- OJET server
- OLTP server deployed on WLS12c
- WLS Wallet

Whereas stand-alone CA requires only administrative permissions on the server which you will install the certificate service.

ORACLE FLEXCUBE Installer performs SSL Enabling configurations using Self-Created Dummy Certificates. For production Environment, Valid Certificates from Certificate Authority (CA) needs to be installed. Detail steps to perform SSL Enabling configuration can be referred from "SSL On OJET.docx".



SSL On OJET.docx

Following Ciphers are supported for SSL configuration:
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA

3. Installation

3.1 Data Center Practices

3.1.1 Overview

The following guidelines are recommended to secure the host servers (Application Server, Database Server and others) in an installation of Oracle FLEXCUBE Core Banking.

3.1.2 Physical System Security

It is highly recommended to operate servers in a secured data center to prevent unauthorized users or operating personnel from tampering with the machines.

3.1.3 Minimize the Server Footprint

Each logical software component (Application Server, Database Server etc.) in the installation should preferably operate in a dedicated server. It is not recommended to operate multiple services like mail, FTP, LDAP etc. on the same server, unless absolutely necessary.

It is preferable to customize the operating system installation so that only the minimum set of software components is installed.

Development tools should not be installed on the production servers. In cases where a software package should be compiled and built before installation, it is advisable to perform the build process on a separate machine, following which installation of the binary can be performed on the server.

Samples and demos should not be deployed on a production server, since they are bound to be developed without considering security. Any bugs in such software can be exploited by an attacker resulting in a security incident.

3.1.4 Operating System Users and Groups

It is recommended to minimize the number of user accounts on the host, for easier auditing and management. Besides, it reduces the risk of unauthorized personnel accessing the server.

It is recommended to create user accounts with names that are not easily guessable. There should be at least two system administrator accounts for a server, to ensure backup in the eventuality of one account being locked.

Passwords for all accounts should be strong passwords – this should be enforced by the operating system, for instance, via the **pam** configuration in UNIX. Passwords should not be easy to guess, and neither should they be stored in an insecure media, or written down for easy remembrance.

Passwords should be set to expire periodically; 60-90 days is the recommended period. Passwords for privileged accounts may have a shorter lifecycle.

3.1.5 Restrict File System Access

It is recommended to use a file system that allows maintenance of access rights.

In Windows, NTFS allows for ACLs to be maintained at the most granular level; however, due care should be exercised when granting file system privileges to the “Everyone” group. Similarly, in UNIX like operating systems, privileges should not be granted to the “Nobody” user and group, unless absolutely required.

3.1.6 Network Perimeter Protection (For Production Mode Only)

Firewall rules should be established to ensure that only a required set of services is accessible to machines outside the data centre. Network access can be further restricted to ensure that only certain subnets with trusted machines, and not all machines, can access machines in the data centre.

Oracle Financial Services does not recommend exposing the application server hosting Oracle FLEXCUBE Core Banking to the Internet.

3.1.7 Network Service Protection (For Production Mode Only)

Network services installed on the server should be enabled only to serve the primary business function(s) that the server must provide. Disable all services that are not needed to serve a justified business need.

Review the network services (like mail and directory services) running on the servers to ensure that they are adequately protected from abuse by an attacker.

Also review and limit the network file shares on the servers, to reduce the risk of an attack on the file system. It is recommended to share files and directories on servers only to trusted machines in the network.

3.1.8 Usage of Protected Ports (For Production Mode Only)

It is not recommended to execute long processes like application servers and database servers under the root account, since a compromise of such processes will result in an attacker gaining elevated privileges.

Therefore, limit the use of protected ports (port numbers less than 1024 on UNIX like operating systems), since they require the use of a privileged user account (in most cases, this is only the root account). Consider the use of NAT to map protected ports to unprotected ones.

3.1.9 Installation of Software in Production Mode (For Production Mode Only)

It is highly recommended to install production builds of any software on production servers. For example, Oracle WebLogic Server should be installed in the production mode, as opposed to the default of development mode. The Oracle Database Server should be installed with options required for production usage (for instance, do not install the sample schemas).

Moreover, it is highly recommended to refer to the manuals and documentation provided by the software supplier, for installing and operating such software securely in a production environment.

3.1.10 Software Updates and Patches

It is recommended to subscribe to security bulletins and advisories published by software vendors to ensure that critical servers are always up to date.

Oracle Financial Services recommends that patches be tested to ensure that they do not conflict with the normal operation of the system.

3.1.11 Usage of Security Appliances and Software

Consider the usage of security appliances and software to monitor and ensure that the production environment continues to be secure after the process of server preparation.

Intrusion Detection Systems can be employed to monitor for security sensitive changes in the system and alert personnel. Antivirus scanners can be used to prevent the server(s) from being compromised. Note that, although UNIX like operating systems may have better defences against viruses (and other malware), consider running antivirus scanners on servers regardless of the OS.

3.1.12 Configure Security Auditing(For Production Mode Only)

Most server operating systems (Linux OS with kernel version 2.6 onwards, IBM AIX, Microsoft Windows Server 2008 etc.) allow for auditing file and directory access. Oracle Financial Services recommends enabling this feature in order to track file system access violations. It is not recommended to enable audit for normal file access operations; audits should preferably contain records of violations to reduce the amount of noise in the logs.

Administrators should ensure sufficient disk space for the audit log. Additionally, administrators should factor the increase on server load due to auditing being enabled.

3.1.13 Separation of concerns

It is not recommended to perform development of any kind on a production machine. The standard practice is to establish a separate development environment for developers, isolated from the testing/staging and production environments. Additional environments can be created for other purposes (for instance, a post-production support environment).

3.1.14 Separation of concerns (For Production Mode Only)

Back-ups should be taken regularly. This will minimize downtime if there is an emergency. Access to the application areas should not be at the operating system level. On-line archival of redologs should be set up from the date of going live. It is recommended that:

- Backup of all database related files viz., data files, control files, redo logs, archived files, init.ora, config.ora etc should be taken at the end of the day.
- The tape can be recycled every week by having day-specific tapes.
- On-line backup of archived redo-log files onto a media to achieve to the point recovery in case of crash, shutdown etc.(recycled every day)
- Complete export of database and soft base should be done at least once in a week and this can be stored off-site (media can be recycled in odd and even numbers).

- Complete backup of the Oracle directory (excluding the database related files) to be taken once in a month. This media can be recycled bimonthly.
- When the database is huge, incremental exports and on-line table space backups are recommended.

The above strategy may be improvised by the Oracle DBA, depending on the local needs. The backup operations are to be logged and tapes to be archived in fireproof storages.

3.2 Oracle Database Security (For Production Mode Only)

3.2.1 Overview

This section contains security recommendations for the Database.

3.2.2 Hardening

Review database links in both production environments. Unwanted links need to be dropped.

3.2.3 Authentication

Middle-tier applications logon to the database through application schemas rather than end-user accounts. Some individuals (IT Administrators) may require direct access to the application database via their own schema.

This setting prevents the database from using an insecure logon protocol. Make sure init.ora contains:

REMOTE_OS_AUTHENT=FALSE

Following an installation, the application database instance contains default, open schemas with default passwords. These accounts and corresponding passwords are well-known, and they should be changed, especially for a database to be used in a production environment.

Use the SQL*Plus PASSWORD command to change a password:

SQL> PASSWORD <SCHEMA>

Metalink Patch note 4926128 contains a SQL script that will list all open accounts with default password in your database.

In addition, the password to the default accounts like SYS, SYSTEM etc. should be complex and securely stored by the bank.

3.2.4 Authorization

The init.ora parameter `_TRACE_FILES_PUBLIC` grants file system read access to anyone who has activated SQL tracing. Set this to its default value of *False*.

_TRACE_FILES_PUBLIC=FALSE

Set the init.ora parameter REMOTE_OS_ROLES to *False* to prevent insecure remote roles.

REMOTE_OS_ROLES=FALSE

Set O7_DICTIONARY_ACCESSIBILITY to *False* to prevent users with Select ANY privilege from reading data dictionary tables. False is the default for the 10g database.

O7_DICTIONARY_ACCESSIBILITY = FALSE

3.2.5 Secure Database Backups

RMAN secure backup should be used to ensure that the backups stolen from your system cannot be restored in another remote system. Additionally, data masking - a feature offered by Oracle Enterprise Manager – can be used to move data from your production environment to a test environment. Both these are very crucial steps towards securing confidential customer data.

The database backups should be stored for the required period as per the regulations and bank's history retention policies. These backups should be securely stored and access should be controlled to authorized users only.

3.2.6 Separation of Roles

It is vital to ensure that roles and responsibilities of database administrators and application users/administrators are clearly segregated. Database administrators should not be allowed to view or access customer data. Oracle Database vault helps to achieve this separation of duty by creating different realms, factors and rule sets. It can enforce policies that prevent a DBA from accessing an application realm. The product has a set of configuration policies that can be directly implemented with database vault. Implementation specific requirements can be imposed over and above these.

3.2.7 Advanced Security

Oracle Advanced Security provides industry standards-based data privacy, integrity, authentication, single sign-on, and access authorization in a variety of ways. Sensitive information that is stored in your database or that travels over enterprise networks and the Internet can be protected by encryption algorithms. An encryption algorithm transforms information into a form that cannot be deciphered without a decryption key. Oracle Advanced Security supports multiple industry standard encryption algorithms such as RC4, DES3 and Triple-DES. To ensure the integrity of data packets during transmission, Oracle Advanced Security can generate a cryptographically secure message digest using MD5 or SHA-1 hashing algorithms and include it with each message sent across a network.

3.2.8 Audit

This section describes the auditing capabilities available in Oracle database. These recommendations should not have a measurable performance impact.

In `init.ora`, set `AUDIT_TRAIL` to `DB`, `OS` or `TRUE`. Consult with the Applications Database Administrator before setting this value to `TRUE`. When set to `OS`, the database stores its audit records on the file system:

```
AUDIT_TRAIL = DB
```

Restart the database for these parameters to take effect.

Note: The database generates some audit records by default, whether or not `AUDIT_TRAIL` is enabled. For example, Oracle automatically creates an operating system file as an audit record when a user logs in as `SYSDBA` or as `INTERNAL`.

Monitoring and auditing database sessions, provides valuable information on database activity and is the only way to identify certain types of attacks (for example, password guessing attacks on an application schema). By auditing database sessions, suspicious connections to highly privileged schemas may be identified.

To audit sessions, login through `sqlplus` as `SYSTEM` and issue the following command:

```
SQL> audit session;
```

Audit any changes to the standard `FCFLEXCUBE` database schema or creation of new schemas. As rare events, these changes may indicate inappropriate or malicious activity.

To audit schema changes, login through `sqlplus` as `SYSTEM` and issue the following command:

```
SQL> audit user;
```

To complete the recommended auditing, enable three other audit events: *create database link*, *alter system* and *system audit*. The remaining audit options generate significant entries of little value. Auditing these other actions provides little meaningful information.

To audit the other events, login through `sqlplus` as `SYSTEM` and issue the following commands:

```
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
```


Connections to the database as well as SYSDBA and SYSOPER actions (instance startup/shutdown) are always logged to the directory \$ORACLE_HOME/rdbms/audit (unless AUDIT_FILE_DEST property is overridden). This file contains the operating system user and terminal ID.

If AUDIT_TRAIL is set to OS, review audit records stored in the file name; in AUDIT_FILE_DEST.

If AUDIT_TRAIL is set to DB, retrieve audit records from the SYS.AUD\$ table. The contents can be viewed directly or via the following views:

- DBA_AUDIT_EXISTS
- DBA_AUDIT_OBJECT
- DBA_AUDIT_SESSION
- DBA_AUDIT_STATEMENT
- DBA_AUDIT_TRAIL
- DBA_OBJ_AUDIT_OPTS
- DBA_PRIV_AUDIT_OPTS
- DBA_STMT_AUDIT_OPTS

The audit trail contains a lot of data; begin by focusing on the following:

- Username: Oracle Username.
- Terminal: Machine from which the user originated.
- Timestamp: Time the action occurred.
- Object Owner: The owner of the object that the user touched.
- Object Name: The name of the object that the user touched.
- Action Name: The action that occurred against the object (INSERT, UPDATE, DELETE, SELECT, EXECUTE).

Archive and purge the audit trail on a regular basis, at least every 90 days. The database connection entries take up significant space. Backup the audit file before purging.

Audit data may contain confidential or privacy related data. Restrict audit trail access appropriately.

It must be noted that auditing features can impose a significant performance overhead. Auditing should thus be limited to the set of items outlined above. Auditing application schema objects should be strictly avoided.

3.3 Database Operating Environment Security (For Production Mode Only)

3.3.1 Overview

The environment in which Oracle Applications run contributes to or detracts from overall system security. This section contains security recommendations for tightening Oracle file system security along with more general advice for overall system hardening.

3.3.2 Hardening

- The directory \$ORACLE_HOME/bin contains Oracle executables. Check that the operating system owner of these executables matches the operating system user under which the files have been installed. A typical mistake is to install the executables in user oracle's directory but owned by root.
- Prevent remote login to the Oracle (and root) accounts. Instead, require that legitimate users connect to their own accounts and to the Oracle account. Better yet, use pseudo to restrict access to executables.

Refer to the product installation documentation for the complete instructions on setting file permissions.

On UNIX systems:

- Set the permissions on \$ORACLE_HOME/bin to 0751 or less. Set all other directories in \$ORACLE_HOME to 0750 or less. Note, this limits access to the Oracle user and its groups (probably DBA).
- Set file permissions for listener.ora and sqlnet.ora to 0600.
- Set file permissions for tnsnames.ora to 0644.
- Ensure that the owner, group and modes of the Oracle files created upon installation are set to allow minimum privilege. The following commands make this change. Note, the group and owner are for illustration only, the correct group and owner should be substituted.

```
$chgrp -R    <dba>          $ORACLE_HOME
$chown -R   <oracle>     $ORACLE_HOME
```

- Review owners and groups when cloning a database
- Protect the \$ORACLE_HOME/rdbms/admin directory including catalog.sql, catproc.sql and backup scripts.
- Secure scripts containing usernames and passwords
- Verify that set user id (SUID) and set group id (SGID) are not set on binaries. In general, Oracle recommends that the SUID and SGID bits to be removed from binaries shipped by Oracle.

On windows systems, NTFS must be used. The FAT/FAT32 file system provides no security.

Use secure shell (ssh) to access middle-tier and database hosts. This replaces telnet, rsh, rlogin, rcp and ftp.

The following services may provide operational convenience:

- NTP (Network Time Protocol) – for synchronizing the clock on the UNIX hosts to provide accurate audit records and simplify trouble-shooting.
- CRON – for operating system cleanup and log file rotation

COPYRIGHT (C) 2020 Oracle Financial Services Software Limited.

3.3.3 Authentication

Good security requires secure accounts.

- Make sure that all OS accounts have a non-guessable password. To ensure that the passwords are not guessable, use crack or john-the-ripper (password cracking tools) on a regular basis. Often, people use passwords associated with them: license plate numbers, children's names or a hobby. A password tester may check for these. In addition, change passwords from time to time.
- Automatically disable accounts after several failed login attempts.
- .netrc files weaken security.
- The fewer people with root access, the easier it is to track changes.
- The root password must be a strong, non-guessable password. In addition, change the root password every three (3) months and whenever an administrator leaves company. Always logout of root shells; never leave root shells unattended.
- Limit root to console login, only (specified in /etc/security).
- Root, and only root, should have UID 0.
- Check root '*.*' files for security holes. The root '*.*' files SHOULD have 700 or 600 permissions
- umask for root is 022 (rwxr-xr-x). A umask of 077 (rwx-----) is best, but often not practical
- To avoid trojan horse programs, always use full pathnames including aliases. Root should NEVER have "." in path.
- NEVER allow non-root write access to any directories in root's path.
- If possible, do not create root's temporary files in publicly writable directories.

Do not share user accounts. Remove or disable user accounts upon termination. Disable login for well known accounts that do not need direct login access (bin, daemon, sys, uucp, lp, adm). Require strong passwords and, in some cases, a restricted shell.

It is hard to imagine what kind of guests should have access to a production system. For this reason do not allow guest access.

3.3.4 Authorization

Only run NFS as needed, apply latest patches. When creating the /etc/exports file, use limited access flags when possible (such as readonly or nosuid). By using fully qualified hostnames, only the named host may access the file system.

Device files /dev/null, /dev/tty and /dev/console should be world writable but NEVER executable. Most other device files should be unreadable and non-writable by regular users.

Always get programs from a known source. Use a checksum to verify they have not been altered.

Create minimal writable file systems (esp. system files/directories). Limit user file writes to their own directories and /tmp. Add directories for specific groups. Limit important file access to authorized personnel. Use setuid/setgid only where absolutely necessary.

3.3.5 Maintenance

Good security practice does not end after installation. Continued maintenance tasks include:

- Install the latest software patches.
- Install latest operating system patches.
- Verify user accounts - delete or lock accounts no longer required.
- Run security software and review output.
- Keep up to date on security issues by subscribing to security mailing lists, reading security news groups and following the latest security procedures.
- Implement trusted file systems like NIS, NIS+ or others such as HP-UX trusted system.
- Test the system with tools like NESSUS (network security) and CRACK (password checker).
- Install Tripwire to detect changes to files
- Monitor log files including btmp, wtmp, syslog, sulog, etc. Also check the snort logs.

The environment in which Oracle Applications run contributes to or detracts from overall system security. This section contains security recommendations for tightening Oracle file system security along with more general advice for overall system hardening.

3.4 Database Practices for Risk Mitigation in Production(For Production Mode Only)

3.4.1 Production Database – Access Prevention

- Only people with a “need to know” or a legitimate administrative purpose should be allowed any form of access to production database.
- All access to production databases must be logged with a specific user ID that maps to a specific individual, either staff or a vendor, including administrator login. There should be separate/dedicated DB user created for FLEXCUBE application which should not be shared or used for any other purpose.
- Monitoring should be done to track non application sessions into the database and activities performed in that session. Guidelines on monitoring tool can be referred from Oracle documentation or Oracle DBA team.
- While providing production access to any individual, process of authorization by higher level management with proper justification should be followed. The access should be controlled by timelines that certain user access will be expired after specific period and should go through renewal process to reactivate.
- The support consultants (OFSS / OFSS partner / Third party vendor / Bank IT) should have individual IDs created on database with strictly Read-Only access. Any consultant demanding for updateable/full access should be reported to senior management of respective vendor.
- All third party tools using Production database for reporting purpose (Like BO reports) should access it via a Read-only access ID meant for specific reporting tool.
- As much as possible, reports should be generated on backup schema and not on production schema.

- If third party tool is writing into production database for processing, it should be restricted via APIs wherever applicable. There should be dedicated user created for each distinct interface and that user activity should be tracked to ensure authenticated sessions/activities.
- Database passwords should be changed at regular intervals at scheduled frequency or event basis. Bank can refer to Oracle FLEXCUBE Password change document attached in appendix section to understand the steps. However, bank is requested to approach to Global Support team before taking up password change activity for the first time.
- Inactive DB User Id which are not used for a specific period (say one month) should be disabled and deleted in case of say 3 months of inactivity. Any activation/recreation of such ID should follow standard process/mechanism. Password profile should be created which will automatically take care of disabling the user ids after inactivity for specified time. These are standard recommendations, bank can have their own timelines defined for these activities.
- The awareness must be created within bank's teams, whoever are having any form of access to production database, to ensure that they do not share their password and do not leave their database session unattended.
- Database Sessions using Toad, sql*navigator etc and running heavy queries from those sessions should be avoided during production End of Day process as it creates additional load and may lead to locks if not used properly.

3.4.2 Production Database – Data Protection

- Any production data shared with support consultant (OFSS / OFSS partner / Third Party vendor) should only be shared in masked form. The vital & sensitive information like Customer Name, Customer's personal details, SWIFT address, account title, address, email id should be masked. (Sample data masking scripts are attached in appendix section for reference. Global Support team can provide actual data masking scripts on request that are applicable for your installation). Vendors should be indicated/informed to delete the shared data once the incident is resolved.
- Printing of production data should be avoided as much as possible and should be printed only when necessary. Printed version of production data should be kept only for required period and destroyed using standard mechanism to avoid it falling into wrong hands. Whenever customer statements are printed, the delivery should be concluded within stipulated period and should be securely stored until then.
- Standard corporate policies like Clean Desk Policy help in strengthening the Data protection. Forming of data controller team to ensure sanity/masking of data before it is handed over for any purpose.
- The Backups and storages should ensure labeling and encryption wherever required. The media recycle policy can be adopted to ensure that old unwanted backup tapes/media are not misplaced.
- Ensure NOT TO share data on personal email ids. No part of data should be uploaded through non official web sites. Sharing data with third party vendor, partners, business teams should be done in protected and encrypted form by ensuring key customer data is masked.

3.4.3 Production Database – Release Management

- It is suggested to define release calendar for releasing bug fixes and software releases into production environment. It is advised to have this calendar always setup towards end of week, before EOD. Unless emergency data correction scripts, critical fixes should not be released in between release cycle. Fixes should be clubbed and released in batches/lots as per predefined release calendar. Release calendar should not be set for releases near End of Month.
- The fix releases into production environment should be handled by release team who has authorized access with dedicated user IDs to manage release procedure. The release team can ensure standard release procedure / steps and should take up pre & post checks during deployment of releases. Ensuring valid objects in database post release deployment is one of key checks to be followed by this team. There should also be check done for ensuring expected rows are affected/impacted after applying data patch before firing commit in the database.
- The release team to ensure that necessary backup of production units is taken before applying any release into production environment. The back out scripts should also be handy in case any release step fails and calls for a rollback.
- Standard scripts (like shell programs in Unix) can be created for application of the fixes so that order of the units to be applied can be managed and other human errors/mistakes can be avoided. These scripts should be tested on UAT environment for each release before using the same for production environment.
- It is suggested that the release schedule and details of fixes to be released are announced with all stake holders within bank as well as with support vendors and Global Support team at least 2 days in advance. Any down time requirement for database maintenance or release plan should be announced well in advance, including to customers as applicable. Support incident should be closed when the release is moved to production so that OFSS can update their VSS sources.
- There could be emergency release requirement, either to quick fix the critical issue, or data correction for critical data anomaly or any patch required to address batch abort. It is suggested to have shift in-charge designated within release management team to take care of these emergency releases. It is suggested to get every script validated with the help of authorized Support vendor/consultant at site or with the help of Global Support team.
- Please ensure not to reuse any previous script or data patch that could have been used in the past to address batch abort or any data anomaly. It is always suggested to get new script/patch prepared or to get the script/patch validated with the help of authorized Support vendor/consultant/GSupp before it is applied/used on production database.

3.5 Application Server Security

3.5.1 Overview

This section describes how to secure the Oracle WebLogic Server production environment that hosts the Oracle FLEXCUBE Core Banking environment.

3.5.2 Installation of Oracle Weblogic Server

By default, Oracle WebLogic Server is installed with a JDK and several development utilities. These are not required in a production environment.

The installation footprint of Oracle WebLogic Server can be reduced via the following measures:

- During installation of Oracle WebLogic Server, customize the components to be installed. The following components are not required by Oracle FLEXCUBE Core Banking in a production environment:
- Oracle WebLogic Workshop
- Web 2.0 HTTP Pub-Sub Server
- Third Party JDBC Drivers (for MySQL and Sybase)
- WebLogic Server examples
- Delete the Pointbase database which is not required for production usage.

3.5.3 Securing the Weblogic Server Installation(For Production Mode Only)

Once installed, the measures listed below can be employed to secure the WebLogic Server installation.

3.5.3.1 Network perimeter protection

It is highly recommended to employ the use of a firewall (as hardware or software) to lockdown the network access to the WebLogic cluster.

For additional information on planning the firewall configuration for a WebLogic Cluster, refer to the section “Security Options for Cluster Architectures” in the “Using Clusters” guide of the Oracle WebLogic Server documentation.

3.5.3.2 Operating system Users and Groups

It is highly recommended to run the WebLogic Server as a limited user process. The root user account in Unix/Linux and the Administrator account in Windows should not be used to run WebLogic Server since they are privileged user accounts. Other privileged accounts should also not be used to run the WebLogic server.

Hence, it is preferable to create a limited user account say “WebLogic Owner” for running the application server. Additional user accounts are not recommended; in the eventuality, that an additional account is required (say, if the WebLogic owner account is locked out), one of the system administrator accounts can be used to remedy the situation. Having two system administrator accounts is recommended, as it always ensures backup.

3.5.3.3 File System Access to OS Users

Access rights to the Oracle Home, WebLogic Server product directory, and the WebLogic domain directories should be provided only to the “WebLogic Owner” user. Privileged users will anyway have access to the WebLogic Server installation, by default.

Users in the “Others” category can be restricted from reading the afore-mentioned directories.

Ensure that the following files in the WebLogic installation are available only to the WebLogic owner:

- The security LDAP database which is usually located in the WL_HOME\user_projects\domains\DOMAIN_NAME\servers\SERVER_NAME\data\ldap\ldapfiles directory
- The keystore used in the keystore configuration of the server(s)
- The Root Certificate Authority keystore

Oracle WebLogic Server provides persistent stores for several FLEXCUBE systems, some of which are utilized by Oracle FLEXCUBE Core Banking. Ensure that access to the persistent file stores based on files is restricted to the WebLogic owner OS user. The default persistent file store is located in the *datastore\default* directory under the *server name* subdirectory under the WebLogic domain’s root directory. If custom (user-defined) persistence stores have been created, the same restrictions should be applied on the files and directories used by such stores.

3.5.3.4 Usage of Protected Ports

In the case of Oracle WebLogic Server

- Operate WebLogic Server using an unprivileged account, bind to unprotected ports, and use NAT to map protected ports to the unprotected ports.
- Configure WebLogic Server to start with a privileged account, bind to protected ports, and then change the user account to an unprivileged user account. For this purpose, Oracle WebLogic Server on UNIX needs to be configured to have a post-bind user ID or group ID. For additional details, refer to the section “Create and configure machines to run on UNIX” in the “Administration Console Online Help”.

3.5.3.5 Usage of Weblogic Connection Filters

Although firewalls restrict the ability of machines to communicate with the WebLogic Server, machines in the data center can still access network services provided by the WebLogic Server.

Configure the Oracle WebLogic Server installation to use connection filters to ensure that only certain machines in the data center can access the WebLogic Server services like HTTP, LDAP, RMI-IIOP etc.

3.5.3.6 Secure the Embedded LDAP Port

In a WebLogic Server cluster, restrict access to the embedded LDAP server port only to machines in the WebLogic Server cluster, through the user of connection filters.

3.5.4 Securing the WebLogic Security Service(For Production Mode Only)

You need to ensure the following.

3.5.4.1 Impose size and Time Limits

Consider enforcing constraints on size and on the amount of time taken for a message to arrive at the server. This will ensure protection against denial-of-service attacks against WebLogic Server. Additional details are provided in the Oracle WebLogic Server documentation, in the guide “Securing a Production Environment”, and also in the “Administration Console Online Help”.

Oracle Financial Services recommends that changes, once done in this regard, be tested thoroughly for impact on business continuity – it is quite possible that WebLogic Server receive valid messages that are large enough to be considered as an attack, when such is not the case.

3.5.4.2 User Lockouts and Login Time Limits

The Oracle WebLogic Server guide on “Securing a Production Environment” has a section on configuring user lockouts and login time limits to prevent attacks on user accounts. In general, Oracle FLEXCUBE Core Banking does not utilize the WebLogic Security Service for managing FLEXCUBE Core Banking user accounts.

Therefore, changes recommended by the WebLogic Server guide should be applied only after assessing the impact on production. The changes applied would be suitable for accounts managed by Oracle WebLogic Server. Note that the WebLogic Server Online Console guide will reference “Compatibility Security” which is deprecated in Oracle WebLogic Server 12c (12.2.1.3.0).

Generally, Oracle FLEXCUBE Core Banking employs its own protection mechanisms with respect to user lockouts.

3.5.4.3 Enabling Configuration Auditing

Configuration auditing can be enabled to ensure that changes to any WebLogic resource configuration in the WebLogic domain are audited. Enabling this option also allows for auditing of management operations performed by a user on any WebLogic resource.

For additional details, refer to the “Administration Console Online Help”, and the “Configuring WebLogic Security Providers” section in the “Securing WebLogic Server” guide of the Oracle WebLogic Server documentation.

Note that enabling configuration auditing will affect the performance of the system, even though auditing may be enabled for auditing a few events (including configuration changes).

3.5.4.4 System Administrator Accounts

Create at least two system administrator accounts (WebLogic user accounts) for administration of the WebLogic server. The first administrator account will be created when the WebLogic domain is created. Create the second account with the Admin security role.

Provide unique names to the administrator accounts that cannot be easily guessed. Oracle Financial Services discourages naming the WebLogic administrator account as 'weblogic' with a password of 'weblogic'.

Again, having two system administrators ensures that at least one system administrator has access to the WebLogic server in the event of the other being locked out.

3.5.4.5 JAVA2 Security Manager Policy changes in Weblogic Servers

When we run WebLogic Server under Java 2 (SDK 1.2 or later), WebLogic Server can use the Java Security Manager in Java 2, which prevents untrusted code from performing actions that are restricted by the Java security policy file.

The JVM has security mechanisms built into it that allow you to define restrictions to code through a Java security policy file. The Java Security Manager uses the Java security policy file to enforce a set of permissions granted to classes. The permissions allow specified classes running in that instance of the JVM to permit or not permit certain runtime operations. In many cases, where the threat model does not include malicious code being run in the JVM, the Java Security Manager is unnecessary. However, when untrusted third-parties use WebLogic Server and untrusted classes are being run, the Java Security Manager may be useful.

To use the Java Security Manager with WebLogic Server, specify the `-Djava.security.policy` and `-Djava.security.manager` arguments when starting WebLogic Server. The `-Djava.security.policy` argument specifies a filename (using a relative or fully-qualified pathname) that contains Java 2 security policies.

WebLogic Server provides a sample Java security policy file, which you can edit and use. The file is located at `WL_HOME\server\lib\weblogic.policy`.

Currently In FC Core application, we are using default weblogic.policy file and no specific application related changes has been done.

4. Post Installation

4.1 Desktop Security

4.1.1 Hardening the Browser

Oracle FLEXCUBE Core Banking is **certified** for usage in Google Chrome, Mozilla Firefox. The browser provides recommendations from a security perspective and customers are encouraged to employ the recommendations provided by them.

In all browsers, it is recommended to enable the popup blocker with a specific rule to disable popup-blocking for the FLEXCUBE web application.

4.1.1.1 Hardening Google Chrome

For Google Chrome, we provided guidance for enhancing Chrome security in the following documents for Chrome.

- Google Chrome Desktop Security Guide



Google Chrome
Security Guide.docx

4.1.1.2 Hardening Mozilla Firefox

For Mozilla Firefox, we provided guidance for enhancing Mozilla Firefox security in the following documents for Mozilla Firefox.

- Mozilla Firefox Desktop Security Guide



Mozilla Firefox
Security Guide.docx

Among the guidelines provided in these documents, Oracle specifically recommends the following settings to all customers of FLEXCUBE :

- Certificate Security - Ensure the usage of SSL 3.0 and TLS 1.0. Disable SSL 2.0 as it is an insecure protocol.
- Privacy Settings - Set Form auto complete options to Disabled. This will prevent inadvertent caching of data keyed by users.
- Privacy Settings - we recommend to clear cache every time browser window is closed– Oracle FLEXCUBE relies heavily on client-side caching performed by browser using this folder. The application will behave slowly after this setting is enabled, since the browser will download resources from the server after every browser restart. Hence, it is not recommended to enable this setting. It should be noted that the details of transactions performed by the FLEXCUBE users are not cached in the Temporary Internet Files folder (irrespective of this setting).
- Other Security Recommendations - Do not save encrypted pages to disk – By default, browsers stores both encrypted and unencrypted content in the Temporary Internet Files folder. Enabling this setting is bound to cause performance issues (especially when FLEXCUBE is accessed over HTTPS), since the browser will no longer cache resources. As stated before, details of transactions performed by users will not be cached in the Temporary Internet Files folder (irrespective of this setting).

4.1.1.3 Disabling of AutoComplete option for Sensitive Pages

Disabling of AutoComplete options is also mandatory at anywhere, where sensitive/credential data is involved. To apply the change on browser level below mentioned methods will be helpful. Changes vary according to browsers and it was provided on the browsers specific guide.

4.2 Branch Environment Security

4.2.1 OHS Configuration

For the OHS configuration kindly refer the installation manual sections 3.3 and 3.5 in 'Oracle FLEXCUBE BRANCH Installation Guide'.

4.2.2 Security by Default Issues

There is a by default option to create some users by the FLEXCUBE 11.9.0.0.0 during the installation. Using these users the client can create various tellers and supervisors. Once this activity is completed, as per best practices it is mandatory to lock these users created during installation process. Also every user should be enforced to change password at first login itself.

4.2.3 OHS Hardening

These sections contain step-by-step guides for security enhancements.

Disabling of Directory Listing

Directory browsing or directory listing should be disabled on the server level to ensure the security of the product.

Directory listing is a web server function that displays a list of all the files when there is not an index file, such as index.php and default.asp in a specific website directory.

One of the easier methods to achieve the same in OHS server is to go to
To disable Directory Listing Edit httpd.conf as :

Comment :- Options Indexes FollowSymLinks

Add :- Options FollowSymLinks MultiViews

Detailed steps provided in “OHS hardening.pdf”.

4.2.3.1 Setting up X-frame Option

Clickjacking is a way to trick visitors into interacting with a victim website without the user knowing he's doing it by e.g. overlaying other things such as images over the elements. Framebusting is a common technique to prevent clickjacking. In addition to that X-Frame-Options was introduced as an alternative. It can be used to prevent framing of the pages that are delivered to browsers in the browser: the browser simply refuses to render the page in a frame if the header is present depending on the set value. Values are DENY: Stops all framing and SAMEORIGIN: Stops framing except for the same website that delivered the page itself. In FLEXCUBE 11.9.0.0.0, we are using Javascript Framebreakers to avoid clickjacking. Steps to set X-Frame option are in in “OHS hardening.pdf”.

4.2.3.2 Security Headers in HTTP Response

The X-Content-Type-Options response HTTP header is a marker used by the server to indicate that the MIME types advertised in the Content-Type headers should not be changed and be followed. This allows to opt-out of MIME type sniffing, or, in other words, it is a way to say that the webmasters knew what they were doing.

Site security testers usually expect this header to be set. To Set X-Content-Type add following

module in httpd.conf :

```
<IfModule mod_headers.c>
```

```
Header set X-Content-Type-Options nosniff
```

```
</IfModule>
```

Detailed steps provided in “OHS hardening.pdf”.

4.2.3.3 Disabling Support for weak SSL 3.0 and TLS 1.0 protocol

After the recent POODLE unpleasantness, both Google and Chrome secured their latest browser versions (Firefox 35, Chrome 40) by barring the use of the SSL 3.0 encryption protocol entirely, since POODLE utilizes this protocol as an attack vector. (Microsoft has released various patches and quick-fixes for Internet Explorer 11 and states they'll completely disable SSL 3.0 in April 2015.)

Disabling SSL 3.0 is definitely a Good Thing. However, the subsequent revelation that TLS 1.0 is **also** vulnerable seems to have caught them on the off foot – TLS 1.0 is still enabled by default in all three major browsers as of this writing.

Enable TLS 1.1 , TLS 1.2 mentioned in “OHS Hardening.docx”

4.2.3.4 Hide Etag

It allows remote attackers to obtain sensitive information like inode number, multipart MIME boundary, and child process through Etag header.
To prevent this vulnerability, Add the following directive and save the httpd.conf.

FileETag None

4.2.3.5 Disabling Support for weak SSL Cipher suites

A weak encryption scheme can be subjected to brute force attacks that have a reasonable chance of succeeding using current methods and resources. An attacker may be able to execute a man-in-the-middle attack which would allow them to intercept, monitor and tamper with sensitive data.

To avoid such attacks disable support for weak cipher suites through Branch machines 'Group Policy'.

Steps to disable weak SSL Cipher Suites are in in "OHS Hardening.docx"



OHS Hardening.doc

4.3 Oracle FLEXCUBE Core Banking Controls

4.3.1 Overview

This chapter describes the various programs available within Oracle FLEXCUBE 11.9.0.0.0, 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.

4.3.2 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 property 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.

4.3.2.1 Sign-on Messages

Login Type	Message	Explanation
Standalone Login	User Already Logged In	The user has already logged into the system and is attempting a login through a different terminal.
	User ID/Password is wrong	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).
SSO Login	Authentication failed	An incorrect user ID or password was entered.
	SSO Session Invalid	When User gets logged out or system is inactive for longer time than interval time.

4.3.3 Display/Print User Profile

This function provides an on-line display of user profiles and their access rights. The information includes:

- The type (customer / staff)
- The status of the profile - enabled or disabled or on-hold

- The time of the last login
- The date of the last password /status change
- The number of invalid login attempts
- The language code / home branch of the user

4.3.4 Clear User Profile

A user ID can get locked into the system due to various reasons like invalid attempts, improper logout or a system failure. Invalid attempts get tracked in the application tables for audit in case of any security breach or analysis.

The System administrator user can reset the status of the user who got locked in. To accomplish this task, System Administrator uses the “Modify Login Status” screen (codTask=755). Here the login status of the locked user ID can be changed to unlocked.

4.3.5 Change User Password

Users can use this function to change their passwords. A user password should contain a minimum of eight characters and a maximum of twelve characters (both parameterizable). It should be different from the current and one previous password. The program will prompt the user to confirm the new password when the user will have to sign-on again with the new password.

For SSO login :- User can change password through Oracle Identity manager. User details will be copied using OIM connector.

4.3.6 Default password Policy

The Password policy for the Default password is maintained as below:

- The number of characters in a user password is not allowed to exceed the maximum length, or fall below the minimum length that has been specified.
- The minimum length is 1, and the maximum length to 12.
- The Password should be alphanumeric, including special characters.
- The Minimum No of Special Characters allowed is 1.
- The Minimum No of Numeric Characters allowed is 1.
- The Minimum No of Lower Case Characters allowed is 1.
- Minimum No of Upper Case Characters allowed is 1.

It is strongly recommended to change the user password during first login itself.

4.3.7 List of Logged-in Users

The System administrator user can see which users are in use within Oracle FLEXCUBE 11.9.0.0.0 at that point of time. The information includes the following:

- The ID of the terminal
- The ID of the user
- The login time

4.3.8 Default Login Type Configurations

Standalone login is not configurable, it will be always enabled.

SSO login is configurable. User can enable/disable this type of login by modifying 'fcubs.properties' .and "CommonConfig.js" .

By default SSO Login is enabled .

4.3.9 Reporting of Security Violations

While logging into the system it checks :

- If the user logging into the system is a valid user or not via a password
- Is the given user is already logged in then it will give the popup "User Already Logged In"
- For the server being down it will give the popup "Login Failed .Try again later".
- If a user tries to login with a user-id which is logged in from another browser, even if the session has been timed out the system will not allow the user to login until the user gets logged out.

4.3.10 Audit Reports

Audit Reports for any transaction that is done is present and can be viewed via the Audit trail Inquiry screen. The information logged includes Branch, Task Id, Posting Date, Teller Id, Authorizer id, Action, Transaction date, Account No, Customer Id and Audit Comments.

Various type of transactions like Cash, transfer and clearing are posted to accounts across the modules. An adhoc report is generated which provides MIS information listing the transactions performed by all the tellers logged in for the day. This is the teller transaction report for all the tellers. Each column of this report provides details on User ID, Currency, Type, Description, Literal, Number of Transactions, Total Amount , Commission and Charges.

4.3.11 Session Timeouts

Sessions are given a timeout period. This timeout period is set in milliseconds, this timeout period is set by a "interval" property on the branch side day 0. Important to note here is that the Timeout functionality of FLEXCUBE side will work only when the window of UBS is closed.

4.3.12 Terminal Lockouts

Oracle recommends that a terminal lockout policy be put in place to automatically lockout unattended PC sessions after a certain duration. This is primarily because Oracle FLEXCUBE 11.9.0.0.0 will not lock out the browser session, although it does expire the browser session after certain period of inactivity. Users may however be able to access unattended sessions while the FLEXCUBE 11.9.0.0.0 user is still logged in. Hence, organizations are expected to set a corporate policy for handling unattended PC sessions; it is recommended to enable the feature to lock workstations, or to enable password-protected screensavers.

4.3.13 Enable Request Encryption from Client to OHS Server

The request encryption from client to OHS can be enabled by toggling the following two flags after the Host installation:-

- On Host side
/scratch/Domain_name/OJET/runarea/FCBRNMWEnv/config/properties/Encrypt.properties toggle the value of the flag named "ENCRYPTION_ENABLED" to "true".
- On OHS Server

/scratch/domain_name/Oracle/Middleware/Oracle_Home/user_projects/domains/OJET/config/fmwconfig/components/OHS/instances/ohs1/htdocs/js/FB/config/CommonConfig.js toggle the value of the flag named "IsEncryptionRequired" to "true".

5. General

5.1 Overview

FLEXCUBE Core Banking (FCB) incorporates following Security Measures:

- 1) 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.
- 2) FLEXCUBE Core Banking (FCB) Branch Login pops following Sign-on messages whenever the user tries to login in the system:

Message	Explanation
User Already Logged In	The user has already logged into the system and is attempting a login through a different terminal.
Invalid User ID/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) The Display /Print user profile provides the on-line display of user profiles and their access rights which includes following information:
 1. The type (customer / staff)
 2. The status of the profile - enabled or disabled or on-hold
 3. The time of the last login
 4. The date of the last password /status change
 5. The number of invalid login attempts
 6. The language code / home branch of the user
- 4) The User Id gets locked due to various reasons like an improper logout or a system failure, whose status can only be reset by the System Administrator.
- 5) Strong password Creation features allowing creating password having combination of small, large characters, special characters and numbers.
- 6) System Administrator can keep tab on the users logged in that point of time and can view information such as Terminal Id, user Id, Login time of the User's Logged in.
- 7) Following Security Violations are reported while trying to log in to the system:
 1. If the user logging into the system is a valid user or not via a password
 2. Is the given user is already logged in then it will give the popup "User Already Logged In"
 3. For the server being down it will give the popup "Login Failed .Try again later".

4. If a user tries to login with a user-id which is logged in from another browser, even if the session has been timed out the system will not allow the user to login until the user gets logged out.

- 8) Strong Hashing Algorithms are used to store the User login passwords.
- 9) Strong Encryption Algorithms are used to store the Database passwords.
- 10) Multi-level authorization for critical Transactions and Maintenances.
- 11) Audit Reports for any transaction that is done is present and can be viewed via the Audit trail Inquiry screen. The information logged includes Branch, Task Id, Posting Date, Teller Id, Authorizer id, Action, Transaction date, Account No, Customer Id and Audit Comments. Various types of transactions like Cash, transfer and clearing are posted to accounts across the modules. An adhoc report is generated which provides MIS information listing the transactions performed by all the tellers logged in for the day. This is the teller transaction report for all the tellers. Each column of this report provides details on User ID, Currency, Type, Description, Literal, Number of Transactions, Total Amount, Commission and Charges.

Sessions are given a timeout period. This timeout period is set in milliseconds; this timeout period is set by a "Interval" Property in Rec_glset table There is no limit on the maximum/minimum limit for session interval timeout. But we set it to 600sec. The Session interval validation is performed at the HTTP Server only.

- 12) Frame-Busting Logic is implemented to prevent **Click jacking** and prevent framing of the pages that are delivered to browsers, in the browser: the browser simply refuses to render the page in a frame if the header is present depending on the set value.

Authentication Features in Core Banking

The User's are authenticated during Application Login itself. Users are also authenticated during Local Authorization of transactions.

All the users of the bank can now be broadly classified depending upon their role and seniority. Such classification is represented in system using Roles/Access Profiles. All screens are linked to one or more such roles to define the access to the screens for the respective type of users. Each user needs to have a user profile defined in the system. This user profile is linked to one of the roles or multiple roles. User roles differentiate users based on the level, nature of tasks to be done by that group of users, access codes for controlling access across branches.

Every user is linked to an access level. The Transactions performed by Teller User's can only be authorized by Supervisor User belonging to higher access level. Any transaction performed by Supervisor user can be authorized by another Supervisor user. User's belonging to lower access level cannot authorize the transactions performed by higher access level users.

Authorization operations can be performed as below:

Authorization can be performed with Consolidated Authorization Reasons i.e. Local and Host Auth Reason.

Parallel Authorization.

Deferred Authorization-Support for Maintenances only

Auto-Submit by Supervisor after Remote Auth.

Seniority of users for authorization of transactions can be defined using the Access Profile level.

A higher Access profile level is indicative of higher level of the user attached to it within the bank.

Transactions posted by a teller can be authorized only by those supervisors whose Access profile have a higher level than that of the teller posting the transaction.

Role Based Access Control in Core Banking

- Access Profile Code: Access Profile code can be attached to a user. Access Profile to enable retail users to have functionalities such as Access Levels, Access Profile Security settings, Limits and Work Times.
- Class Code: Distinguishing Users based on the area of work such as CASA, LOANS etc, would be possible through attaching a Class Code. Class Code validations would be performed during authorization to ensure that a User of the same Class Code only can authorize the transaction.
- Employee Id: Field added to enforce the Staff restriction check functionality in FC Core.
- LOB Code: Line of Business (LOB) code would be attached to identify which cost centre in Bank the User falls under.

User Access profile is a grouping of users with common requirements for access rights. Users with a common set of access rights can be linked to the access profile with those access rights. User access profile differentiates users based on the level, access codes for controlling access across branches. Access profile level transaction limits help business to define limits on financial transactions for all users of the system. These include online and offline limits for same branch and inter-branch transactions. Role Profile 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. Role is basic unit to group access rights for end user. For each user, at least one role is mandatory to access any of the systems. Users are linked to access profiles. The changes made to the access profile attributes are reflected to the users who are associated with the access profile. Class Profile is the maintenance of categories of Users within a bank based on their department, eg. Retail Loans, CASA etc. Currently, the functionality is applicable only to retail accounts.

Maintain Access Profile Level Transactions Limits

Using this option limits on financial transactions can be maintained in the system for all users of the system. The limits can be maintained for a group of users under a particular access profile and currency combination. The online and offline limits for same branch and for inter branch are maintained in this option. Business can also populate the limits that are assigned to the transaction group to the individual transaction mnemonics. Transaction group will list core banking transactions only, hence these limits will be applicable to core banking transactions only

Maintain Transaction group Code

Using this option user can maintain a transaction group code and linkage of branches to this code. This group code is linked to the access profiles in the **Access Profile Definition** (PUC: SMDTMPRO) option. If user performs any transactions on an account that belongs to a branch which is in the group code linked to the user access profile, the system will allow transactions on that account; else the transaction will be rejected. This validation is applicable only to financial transactions.

5.2 Validation

The Validations in the product can be categorized in two parts:

5.2.1 Client Level Validations:

The client side validations comprise of the screen level input validations to validate the user inputs for host processing.

5.2.2 Host Level Validations:

The host side validations comprise of the Input Data validations along the with Business validations.

5.3 Session Management

Session ID is created during login by encoding a random number using Base64Encoder and concatenating it with the encoded User Id . The maximum length of the session ID is 32. For every login a fresh session ID is created. No session tokens are associated with the sessions created. We don't have 302 redirect page at logout.

During Login process, this Session Id is then maintained against the user no of the logged in user. This entry is used to validate the user session, based on the logged session-Id and user no.

5.3.1 Session Storage

The user session information is stored in the branch database tables.

5.3.2 Session logging

The user session is logged in tables in the branch database tables. Session Logging for invalid sessions are not logged and validated.

5.4 Password Management

5.4.1 Password Protection

1. The host database password is stored in the properties file/ registry. These passwords are encrypted using the AES-256 Algorithm and the encrypted passwords are then stored in the property file/registry.
2. The user passwords are encrypted using PBKDF2 hashing algorithm and are stored in the database.
3. Passwords are not transmitted directly from the browser to database. The credentials are maintained in encrypted format within host property files.
4. For all kind of transactions the password sent across the network is in encrypted format, except for the case where a transaction requires the resetting of the password (eg:768 screen)
5. For all transactions over the network we rely on SSL(Secured Socket Layer).
6. In the current scenario we don't have any credential storage framework.
7. We do not support the functionality to email the password of the user.

5.5 Exception/Error Handling

All the Exceptions are handled using requisite catch Blocks in java.
All the SQL statements, procedure/functions calls are enclosed with PL/SQL Block statements.
Most of the Exception blocks handles requisite exceptions and raise proper errors.In FLEXCUBE CORE BANKING, in some situations explicit object locks are obtained to maintain data concurrency. In such cases the handling is done in the exception block, either by raising proper errors or setting the retries to acquire the objects till max retry counter is reached (i.e. 10).

5.6 Logging

The Debugs are written in the Debug Logs which are present on the host servers, in the application deployment area. The branch dump logs are written on the HTTP Server and are created in temp folder of the HTTP Server. These logs are not publicly accessible and have restricted access. Audits logs are maintained separately for maintenance screens and oltp transactions. On screen level we use BA777 to maintain the audit handling and for OLTP transactions same can be viewed through screen 6006. Regarding plain password stored in XML's we do not have any handling for the same to protect sensitive transaction. It is not recommended to execute long processes like application servers and database servers under the root account, since a compromise of such processes will result in an attacker gaining elevated privileges. Therefore, limit the use of protected ports (port numbers less than 1024 on UNIX like operating systems), since they require the use of a privileged user account (in most cases, this is only the root account). Consider the use of NAT to map protected ports to unprotected ones.

5.7 Additional Oracle Products Offering Security (For Production Mode Only)

There are some other Oracle products offering Security that are qualified with the FLEXCUBE product. These can be used to enhance the security of the Product and environment.

FLEXCUBE is qualified with following ORACLE Products that offers enhanced Security:

a. Database vault

Database Vault 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. Some of the features of database vault:

- Restriction to ANY-type privileges
- Support to restricted base access using IP, TIME and other
- Out-of-the-box reports to address Security matrix
- Configuration of Policy, Rules based on requirement

b. Audit vault

Audit Vault 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. Some of the features of database vault:

- Prevents modifying audit data including privileged users like DBA and Auditors
- Provides proactive threat detection through Alerting
- Event alert help mitigating risk and protect from insider threats
- Continuous monitoring and evolutions of audit data against alert condition

- c. Transparent Data Encryption

Transparent data encryption enables simple and easy encryption for sensitive data in columns without requiring users or applications to manage the encryption key. This freedom can be extremely important when addressing, for example, regulatory compliance issues. No need to use views to decrypt data, because the data is transparently decrypted once a user has passed necessary access control checks. Security administrators have the assurance that the data on disk is encrypted, yet handling encrypted data becomes transparent to applications.

- d. Oracle Database Security
 - Database Firewall
 1. Accurately detects and blocks unauthorized database activity including SQL injection attacks by monitoring traffic to Oracle and non-Oracle databases
 2. Provides enterprise security intelligence and efficient compliance reporting by combining monitoring and audit data
 3. Utilizes a unique SQL grammar analysis engine and easy-to-define whitelists and blacklists to ensure high accuracy and performance
 4. Delivers horizontal and vertical scalability through easy-to-deploy "software appliances"

 - Label Security
 1. Ensure access to sensitive data is restricted to users with the appropriate clearance level
 2. Enforce regulatory compliance with a policy-based administration model
 3. Establish custom data classification schemes for implementing "need to know" access for applications
 4. Labels can be used as factors within Oracle Database Vault command rules for multifactor authorization policies
 5. Integrates with Oracle Identity Management, enabling centralized management of policy definitions

- e. Data Protection / Advanced Security
 - Data Masking
 1. Sensitive information, such as credit card or social security numbers, can be replaced with realistic values
 2. Production data can be safely used for development, testing, or sharing with out-source or off-shore partners
 3. Uses a template library and format rules, consistently transforming data in order to maintain referential integrity for applications
 4. Extensive search capabilities scan enterprise databases for sensitive data and rank results based on probability of match
 5. Helps comply with data privacy mandates such as Sarbanes-Oxley, Payment Card Industry (PCI) Data Security Standard (DSS) and Health Insurance Portability and Accountability Act (HIPAA)

5.8 References

5.8.1 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

<http://www.sas70.us.com/industries/data-center-colocations.php>

<http://www.anixter.com/content/dam/Anixter/White%20Papers/12F0010X00-Four-Layers-Data-Center-Security-WP-EN-US.pdf>

5.8.2 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/products/secure-backup/overview/index.html>

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

http://www.red-database-security.com/wp/sentriago_webinar.pdf

http://www.databasesecurity.com/oracle/twp_security_checklist_db_database.pdf

<http://www.checklist20.com/pdfs/Databases/Oracle%20Database.pdf>

<http://www.applicure.com/blog/database-security-best-practice>

5.8.3 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

<http://ecomputernotes.com/database-system/adv-database/security-in-database-environment>

<https://security.berkeley.edu/node/138?destination=node/138>

5.8.4 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.11111/e14529/security.htm

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

<http://isu.ifmo.ru/docs/IAS904/core.904/b10377/arch.htm#1005544>

http://www.ibm.com/developerworks/websphere/library/techarticles/0209_oberlin/oberlin.html

<http://cnc.ucr.edu/security/desktop.html>

<http://makeitsafe.missouri.edu/best-practices/windows.html>

<https://security.tennessee.edu/pdfs/sdlbp.pdf>



Security Practices Guide
[May] [2020]
Version 11.9.0.0.0 should match the Document first page

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/

Copyright © [2014], [2020], 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 recompilation 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.