Security Practices Guide

Oracle Financial Services Lending and Leasing

Release 14.8.0.0.0

Part No. F22291-01

December 2019

ORACLE

Security Practices Guide
December 2019
Oracle Financial Services Software Limited

Oracle Park

Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India
Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax:+91 22 6718 3001
https://www.oracle.com/industries/financial-services/index.html

Table of Contents

1. About this Manual

1.1 Introduction

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

1.2 Audience

This guide is primarily intended for IT department or administrators deploying the application and third party or vendor software's. Some information may be relevant to IT decision makers and users of the application are also included. Readers are assumed to possess basic operating system, network, and system administration skills with awareness of vendor/third-party software.

1.3 Scope

1.3.1 Read Sections Completely

Each section should be read and understood completely. Instructions should never be blindly applied. Relevant discussion may occur immediately after instructions for an action, so be sure to read whole sections before beginning implementation.

1.3.2 Understand the Purpose of this Guidance

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

1.3.3 Limitations

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

1.3.4 Test in Non-Production Environment

To the extent possible, guidance should be tested in a non-production environment before deployment. Ensure that any test environment simulates the configuration in which the application will be deployed as closely as possible.

1.4 Related Information Sources

The related information sources are provided under the following sections.

1.4.1 Data Center Practices

For additional information on data centre security practices, refer to the following links:

Reference	Link
PAM Details	http://www.linux-pam.org/Linux-PAM-html/sag-overview.html http://www.ibm.com/developerworks/library/l-pam/
File system Details	https://en.wikipedia.org/wiki/Filesystem_permissions_http://en.wikipedia.org/wiki/Access_control_list
Port below 1024	http://www.staldal.nu/tech/2007/10/31/why-can-only-root-listen-to-ports-below-1024/
Security Auditing	https://www.sans.org/reading-room/whitepapers/auditing/security-auditing-continuous-process-1150
Oracle Weblogic Server Security Guide - Recommended Deployment Topologies	https://docs.oracle.com/middleware/1221/wls/wls-secure.htm

1.4.2 Oracle Database Security

For additional information on Oracle database security, refer to the following links:

Reference	Link
Oracle Database Vault	http://docs.oracle.com/database/121/DVADM/dvintro.htm#DVADM001
Oracle Authentication	https://docs.oracle.com/database/121/DBSEG/toc.htm
Oracle Auditing Details	https://docs.oracle.com/database/121/DBSEG/toc.htm https://docs.oracle.com/database/121/DBSEG/auditing.htm#DBSEG1023
Oracle Database Security	https://www.oracle.com/database/security/index.html
Oracle Secure Backup	http://www.oracle.com/technetwork/database/database-technologies/secure-backup/overview/index.html http://docs.oracle.com/cd/E16926_01/doc.121/e16564/osb_rman_backup.htm#OBADM183
Best Practices for Oracle Databases	http://www.red-database-security.com/wp/sentrigo_webinar.pdf
Database Security Best Practices	http://www.applicure.com/blog/database-security-best-practice

Reference	Link
Cryptographic Algorithms	https://en.wikipedia.org/wiki/Category:Cryptographic_algorithms
TDE best practices	http://www.oracle.com/technetwork/database/security/twp-transparent-data-encryption-bes-130696.pdf

1.4.3 Database Operating Environment Security

For additional information on security recommendations/practices followed for database environment, refer to the following links:

Reference	Link
SMTP Details	https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol
Netrc Details	http://vgstrategies.about.com/od/pccheatsc/a/Call-of-Duty-Black-Ops-CIA-Logins-Passwords.htm
Oracle Database Security Guide - Keeping Your Oracle Database Secure	http://docs.oracle.com/database/121/DBSEG/guidelines.htm#DBSEG009
Importance, Levels, Requirement of Security in Database Environment	http://ecomputernotes.com/database-system/adv-database/security-in-database-environment
Database Hardening Guidelines	https://security.berkeley.edu/resources/best-practices-how-articles/database-hardening-best-practices?destination=node/138

1.4.4 Application Server Security

For additional details on some of the common security considerations to be followed, refer to the following links:

Reference	Link
Weblogic Installation Guide	http://docs.oracle.com/middleware/1221/core/WLSIG/toc.htm
Cipher Suite Details	https://en.wikipedia.org/wiki/Cipher_suite http://docs.oracle.com/middleware/1221/wls/SCOVR/concepts.htm#SCOVR172
Weblogic Connection Filters	http://docs.oracle.com/middleware/1221/wls/WLACH/taskhelp/security/ConfigureConnectionFiltering.html
SNMP Details	http://docs.oracle.com/cd/E13222_01/wls/docs81/snmpman/index.html
Weblogic hostname verification	http://docs.oracle.com/cloud/latest/fmw122100/SECMG/hostname_verifier.htm#SECMG571

Reference	Link
OWSM Details	http://docs.oracle.com/cloud/latest/fmw122100/WSSOV/owsm-security.htm#WSSOV386
Configuring SSL	http://docs.oracle.com/cloud/latest/fmw122100/SECMG/ssl_overview.htm#SECMG718
Configuring Keystore	http://docs.oracle.com/cloud/latest/fmw122100/SECMG/identity_trust.htm#SECMG365
Managing keystores, wallets and certificates	http://docs.oracle.com/cloud/latest/fmw122100/ASADM/wallets.htm#ASADM2021
WebLogic Server Security Best Practices	https://docs.oracle.com/middleware/1221/wls/LOCKD/intro.htm#LOCKD107
Oracle Fusion Middleware Information Roadmap for Oracle WebLogic Server - Security	https://docs.oracle.com/middleware/1221/wls/wls-secure.htm

1.4.5 Desktop Security

For additional information on common security considerations to be followed, refer to the following links:

Reference	Link
Desktop Security	http://cnc.ucr.edu/security/desktop.html

Note

The above mentioned URLs do not refer to product versions where the application is qualified. These URLs provide additional reading material to understand various concepts discussed in the document.

2. 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.

2.1 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.

2.2 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.

2.3 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.

2.4 Leverage Security Features, Never Disable Them

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.

2.5 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.

About Oracle Software Security assurance refer below link:

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

2.6 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 this 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.

2.7 Conventions Used

Term	Refers to
Application	Oracle Financial Services Lending and Leasing

3. Pre-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 the application.

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 hard to guess. 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

Firewall rules should be established to ensure that only a required set of services is accessible to machines outside the data center. 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 center.

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

3.1.7 Network Service Protection

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

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

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 defenses against viruses (and other malware), consider running antivirus scanners on servers regardless of the OS.

3.1.12 Configure Security Auditing

Most server operating systems allow 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 Backup Controls

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, redologs, 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 softbase should be done atleast 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 tablespace 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.

4. Installation

4.1 Oracle Database Security

4.1.1 Overview

This section contains security recommendations for the Database.

4.1.2 Hardening

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

4.1.3 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 database command to change a password:

SQL> PASSWORD or SQL>PASSWORD USERNAME

Always password command should be used because the password is sent unencrypted over the net (without Advanced Security Option) if the alter user syntax is used.

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.

4.1.4 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 value. The O7_DICTIONARY_ACCESSIBILITY initialization parameter controls restrictions on system privileges when you upgrade from Oracle Database release 7 to Oracle8i and later releases

O7_DICTIONARY_ACCESSIBILITY = FALSE

4.1.5 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 = OS

Set parameter AUDIT_FILE_DEST to the directory where the audit records should be stored. By default, the operating system files are in the $ORACLE_BASE/admin/$ORACLE_SID/adump directory for both UNIX and Windows systems.

AUDIT_FILE_DEST = /opt/app/oracle/admin/ORCL/adump

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 application 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.

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.

4.1.6 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.

4.1.7 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.

4.1.8 Securing Audit Information

Oracle Audit vault is an audit solution that consolidates, detects, monitors, alerts and reports and audit data for security auditing and compliance. Oracle Audit vault provides mechanisms to collect audit data from various oracle database. It helps to consolidate audit data from multiple systems into a single centralized repository. Thus, DBA's of individual systems will not be able to tamper with audit information of their respective databases.

4.1.9 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.

4.1.10 Data Encryption

Oracle Advanced Security TDE provides the ability to encrypt sensitive application data on storage media completely transparent to the application itself. Transparent Data Encryption (TDE) stops would-be attackers from bypassing the database and reading sensitive information from storage by enforcing data-at-rest encryption in the database layer. Applications and users authenticated to the database continue to have access to application data transparently (no application code or configuration changes are required), while attacks from OS users attempting to read sensitive data from tablespace files and attacks from thieves attempting to read information from acquired disks or backups are denied access to the clear text data. TDE addresses encryption requirements associated with public and private privacy and security mandates such as PCI and CaliforniaSB1386.

4.2 Database Operating Environment Security

4.2.1 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.

4.2.2 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 su to the Oracle account. Better yet, use sudo 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.

The database and applications require that the underlying operating system provide certain services.

Electronic Mail

Application may require access to a SMTP Mail Transfer Agent (SMTP MTA) typically send mail. This is required for outbound emails, typically notifications from the application (if this feature is desired by the financial institution). If possible, restrict access to the operating system users who absolutely need the mail facility from the shell.

Remote Access

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

4.2.3 Authentication

Good security requires secure accounts.

- Make sure that all OS accounts are hard to guess. To ensure that the passwords are not guessable, use crack or john-the-ripper (password cracking tools) on a regular basis. Use password cracking tools on a regular basis to ensure password complexity. 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, hard to guess. 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.

4.2.4 Authorization

Run NFS only 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.

4.2.5 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. Consider setting up automatic email or paging to warn system administrators of any suspicious behaviour. 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 some generaladvice guidelines for overall system hardening.

4.2.6 Access Prevention

Authorized Access: Only people with a ‘need to know’ or a legitimate administrative purpose should be allowed any form of access to production database.

Access Logging: 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 application which should not be shared or used for any other purpose.

Access Monitoring: 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.

Providing Access: 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.

Restricted Access for support: 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.

Restricted Access for reporting tools: 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.

Reports from Backup Schema: As much as possible, reports should be generated on backup schema and not on production schema.

Restricted Access for interfaces: 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.

Change Password: Database passwords should be changed at regular intervals at scheduled frequency or event basis.

Production ID restrictions: 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.

Awareness: The awareness must be created within bank's teams, whoever is 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.

Restrict DB Sessions during EOD: 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.

4.2.7 Data Protection

Data Masking: Any production data shared with support consultant (OFSS / OFSS partner / Third Party vendor) should only be shared in masked form. The vital & sensitive information such as Customer's personal details should be masked. Vendors should be indicated/ informed to delete the shared data once the incident is resolved.

Printing of Production data: 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.

Adopt Standard Data Protection Policies: 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.

Protected backup: 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.

Data Sharing: 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.

4.3 Application Server Security

4.3.1 Overview

This section describes how to secure the Oracle WebLogic Server production environment that hosts the Oracle Financial Services Lending and Leasing environment.

4.3.2 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 Financial Services Lending and Leasing 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.

4.3.3 Securing the WebLogic Server installation

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

4.3.3.1 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.

4.3.3.2 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.

4.3.3.3 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 subsystems, some of which are utilized by the application. 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 *path $DOMAIN_HOME\<domain>\servers\<servername>\data\store\default* 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.

4.3.3.4 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’.

4.3.3.5 Choice of the SSL cipher suite

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

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

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

```
....
<ssl>
<enabled>true</enabled>
<ciphersuite>TLS_RSA_WITH_AES_256_CBC_SHA</ciphersuite>
<ciphersuite>TLS_RSA_WITH_AES_128_CBC_SHA</ciphersuite>
</ssl>
....
```

Note

- Configuration of WebLogic Server to support the above defined cipher suites might also require an additional command line argument to be passed to WebLogic Server, so that a FIPS 140-2 compliant crypto module is utilized. This is done by adding **-Dweblogic.security.SSL.nojce=true** as a JVM argument.
- The restriction on cipher suites needs to be performed for every managed server.
- The order of cipher suites is important – Oracle WebLogic Server chooses the first available cipher suite in the list, that is also supported by the client.
- Cipher suites with RC4 are enabled despite it being second best to AES. This is primarily for older clients that do not support AES (for instance, Microsoft Internet Explorer 6, 7 and 8 on Windows XP).
- Cipher suites using Triple DES (3DES) are not listed since the maximum effective security provided by the algorithm is 112 bits.

4.3.3.6 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.

4.3.3.7 Usage of Domain-wide Administration Port for Administrative Traffic

When Oracle WebLogic Server is configured to enable administrative access via the administration port, data is exchanged over SSL, preventing any attacker from sniffing sensitive information about the WebLogic Server configuration.

Furthermore, once the Administration port is enabled, WebLogic Server will serve administration requests on a dedicated port with dedicated resources. A denial service attack mounted on the HTTP/HTTPS channels will not prevent administrators from logging into the WebLogic Server administration console to take corrective actions.

Hence, it is recommended to enable the use of the administration port. Additionally, employ firewall rules or WebLogic Connection Filters to restrict access to the Administration Port to trusted machines from where administrators can log in.

Do note that the Administration Port requires that SSL be enabled and also on every Managed Server. Additionally, the administration port will be common across all managed servers in the domain

Further details on configuring the administration port can be found in the “Administration Console Online Help” guide in the Oracle WebLogic Server documentation.

4.3.3.8 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.

4.3.3.9 Precautions when using SNMP

It is recommended to refer the WebLogic SNMP Management Guide to configure SNMP agents in Oracle WebLogic Server. Due care must be observed over the usage of SNMP v1

and v2 since passwords are sent over clear text in these older version of the protocol. Additional steps required for securing SNMP v3 communication are outlined in the guide.

Oracle Financial Services recommends that changes once done in this regard, be tested thoroughly for impact on business continuity.

4.3.4 Securing the WebLogic Security Service

You need to ensure the following.

4.3.4.1 Enable SSL, but avoid using Demonstration Certificates

Enable the use of SSL so that the servers can be accessed via the SSL listen ports for all supported protocols (including HTTPS).

Oracle WebLogic Server includes demonstration private keys, certificates and trusted certificate authorities that are not intended for use in production. Usage of these keys in production is a security risk due to the free availability of private keys; anyone who has a copy of the WebLogic Server has knowledge of the private keys and can compromise SSL/TLS traffic.

Therefore,

- Use a local CA to issue certificates, or
- Use a root or intermediate CA like VeriSign, Thawte etc. to issue certificates

Oracle Financial Services does not recommend the use of self-signed certificates in production.

Consider avoiding the use of certificates with a MD5 signature; usage of certificates with SHA-1 signatures is recommended. Most root and intermediate CAs have begun phasing out the use of MD5 for signing certificates.

4.3.4.2 Enforce Security Constraints on Digital Certificates

Oracle WebLogic Server performs certificate validation whenever it establishes an outbound SSL connection, or when a two-way SSL connection is established. As part of certificate validation, WebLogic Server checks if the certificate contains the Basic Constraints extension. Ensuring the presence of the Basic Constraints extension will prevent attackers from generating new certificates to aid in website spoofing.

Ensure the check for Basic Constraints extension is enabled, by verifying whether the following line is absent in the WebLogic Server startup command.

```
-Dweblogic.security.SSL.enforceConstraints=off
```

Also verify if any messages have been logged at WebLogic server boot, providing information about the presence of certificates that could be rejected by clients.

4.3.4.3 Ensure that Host Name Verification is Enabled

Oracle WebLogic Server implements host name verification when it acts as a SSL client; this prevents man-in-the-middle attacks from being performed against SSL itself.

It should be noted that the application deployed on WebLogic Server will establish outbound SSL connections in certain scenarios, for instance, when requests are made to the Oracle BI Publisher server. In such an event, Oracle WebLogic Server will behave as a SSL client.

Oracle WebLogic Server will behave as a SSL client in several scenarios besides the outbound SSL requests made by applications deployed on Oracle WebLogic Server. For instance, managed servers will establish SSL connections with the Admin server at boot time. Hence, it is recommended to ensure that host name verification is enabled in Oracle WebLogic Server, which happens to be the secure default.

Oracle Financial Services highly recommends the usage of certificates that will pass verification. Oracle Financial Services also recommends against the usage of demonstration certificates in production. It should be noted that usage of demonstration certificates in a testing or development environment containing a multi-server WebLogic cluster, will result in boot failures for managed servers.

4.3.4.4 Impose Size and Time Limits on Messages

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.

4.3.4.5 Restrict the Number of Open Sockets

Consider limiting the number of sockets opened by WebLogic Server, to prevent some forms on denial-of-service attacks. Further details are available 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 – the number of sockets opened is dependent entirely on system load, which is bound to vary across time, and also across installations.

4.3.4.6 Configure WebLogic Server to Manage Overload

Oracle WebLogic Server can be configured to detect, avoid and recover from overload conditions. Configuring WebLogic Server to manage overload conditions allows for WebLogic Server administrators to connect to it, and take remedial actions. Further details on this topic are available 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 – the definition of an overload condition depends on the system capabilities; therefore, overload conditions are bound to be defined differently for machines of differing capabilities.

4.3.4.7 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, application utilizes the WebLogic Security Service for managing user accounts.

Therefore, changes recommended by the WebLogic Server guide should be applied 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 10.3.

4.3.4.8 Enable 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).

4.3.4.9 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.

4.3.4.10 Setting up Secure Flag for Cookies

If the secure flag is set on a cookie, then browsers will not submit the cookie in any requests that use an unencrypted HTTP connection, thereby preventing the cookie from being trivially intercepted by an attacker monitoring network traffic.

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

1. Cookies are set with Http only as true
2. Cookie secure flag set to true
3. 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>
<session-descriptor>
    <cookie-name>JSESSIONID</cookie-name>
    <cookie-path>/<DeployedApplicationPath></cookie-path>
    <cookie-http-only>true</cookie-http-only>
    <cookie-secure>true</cookie-secure>
    <url-rewriting-enabled>false</url-rewriting-enabled>
```

```
</session-descriptor>
```

Always make sure Cookies are set with always Auth Flag enabled by default for WebLogic server and also recommended to apply the weblogic patch 10.3.5 for versions using below weblogic 10.3.5 to reflect the above changes.

4.3.5 Securing the Application

The following guidelines serve to secure the application deployed on Oracle WebLogic Server.

4.3.5.1 Enforce the Usage of SSL

The Installer configures the application such that all HTTP connections to the application are over SSL/TLS. In other words, all HTTP traffic will be prohibited; only HTTPS traffic will be allowed. It is highly recommended to enable this option is in a production environment, especially when WebLogic Server acts as the SSL terminator.

Ensure that the following snippet of code is present in the web.xml file of the web module.

```
<security-constraint>
    <web-resource-collection>
      <web-resource-name>images</web-resource-name>
      <url-pattern>/faces/skins/*</url-pattern>
      <url-pattern>/faces/images/*</url-pattern>
    </web-resource-collection>
  </security-constraint>
 <security-constraint>
    <web-resource-collection>
      <web-resource-name>OFSLL</web-resource-name>
      <url-pattern>/faces/*</url-pattern>
    </web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
```

Recommendation: Disable the compression of the data over SSL to avoid certain known security vulnerability.

4.3.5.2 Two-way SSL Connection

A two-way SSL is used when the server needs to authenticate the client. In a two-way SSL connection the client verifies the identity of the server and then passes its identity certificate to the server. The server then validates the identity certificate of the client before completing the SSL handshake.

In order to establish a two-way SSL connection, need to have two certificates, one for the server and the other for client.

4.3.5.3 Ensure the Servlet Servlet is Disabled

The application does not use the ServletServlet to create default mappings for servlets. All servlets are directly mapped to the required URLs.

Ensure that the following code snippet (or a similar one that uses the weblogic.servlet.ServletServlet) *does not exist* in the web.xml of the web application.

```
<servlet>
  <servlet-name>ServletServlet</servlet-name>
  <servlet-class>weblogic.servlet.ServletServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>ServletServlet</servlet-name>
  <url-pattern>/myservlet/*</url-pattern>
</servlet-mapping>
```

4.3.5.4 Ensure the Proper Session time out

Session timeout represents the event occurring when a user do not perform any action on a web site during an interval (defined in application). The event, on server side, changes the status of the user session to 'invalid' (i.e. 'not used anymore') and instruct the Application/web server to destroy it (deleting all data contained into it). Application allows defining the session time out. The default value for session time out is 30 minutes.

4.4 Securing the Application Web-Interface

4.4.1 Overview

Different applications deployed on disparate platforms and using different infrastructure need to be able to communicate and integrate seamlessly with the OFSLL in order to exchange data. The OFSLL webservices/interfaces will cater to these integration needs.

The integration needs can be broadly categorized as follows:

- Inbound application integration - used when any external system needs to add, modify or query information within OFSLL
- Outbound application integration - used when any external system needs to be notified of the various events that occur within OFSLL or submit requests to external systems

4.4.2 Inbound Application Integration

OFSLL provides XML based inbound interfaces thus enhancing the need to communicate and integrate with the external systems. The data exchanged between OFSLL and the external systems will be in the form of XML messages. These XML messages are defined in respective interface user guide.

OFSLL Inbound Application Integration Gateway uses the Synchronous Deployment Pattern for addressing the integration needs.

The Synchronous Deployment Pattern is classified into the following:

- OFSLL Web Services Based Synchronous Inbound Application Integration Deployment Pattern
- OFSLL HTTP Servlet Based Synchronous Inbound Application Integration Deployment Pattern

4.4.3 Web Services Based Synchronous Deployment Pattern

The web services deployment pattern will be used in integration scenarios where the external system connecting to OFSLL wants to connect using standards-based, inter-operable web services. In this deployment pattern, the external system will use the SOAP (Simple Object Access Protocol) messages to communicate to the OFSLL web services.

The OFSLL services are of a 'message based' services, i.e., the actual request will be in the form of an XML message, but the request will be a 'payload' within the SOAP message. After the necessary processing is done in OFSLL based on the request, the response is returned to the external system as an XML message which will be a 'payload' within the response SOAP message. The transaction control for the processing will stay with the OFSLL.

After deploying the Web Services, one must configure Weblogic Policy and SSL communication on Web Services.

Follow the steps below to configure:

1. Login to WebLogic application server console (http://hostname:port/console)
2. Select one of the deployed webservice. The WSDL of the webservice will be accessible on http before applying WS-Policy.
3. Browse to Configuration ' WS-Policy and select the policy (say Wssp1.2-Https-UsernameToken-Plain.xml) and save the Deployment Plan.
4. For SSL communication, the vendor servers seek public certificates. Hence, one needs to download the certificates from vendor website and import into your java keystore. You then need to configure Weblogic to present the certificates to vendor servers for successful handshake. Import the downloaded certificates into keystore.
5. Select the weblogic Keystores as ‘Custom Identity and Custom trust’ from the available list and configure the following:
   - Custom Identity Keystore: Java keystore holding the certificates
   - Custom Trust Keystore: Java keystore holding the certificates
   - Custom Identity Keystore Type: jks
   - Custom Trust Keystore Type: jks
   - %Keystore Passpharses: keystore password

4.4.4 HTTP Servlet Based Synchronous Deployment Pattern

The HTTP servlet deployment pattern will be used in integration scenarios where the external system wants to connect to OFSLL using simple HTTP messages.

This is especially applicable to systems such as the following:

- Systems that prefer to use a simple http message based approach without wanting to use SOAP as the standard

For this deployment pattern, OFSLL will expose a servlet. The actual request will be in the form of an XML message. This XML message is embedded into the body of the HTTP request sent to the OFSLL interface. After the necessary processing is done in OFSLL based on the request, the response is returned to the external system as an XML message which is once again embedded within the body of the response HTTP message. The transaction control for the processing will stay with the OFSLL.

4.4.5 Outbound Application Integration

The OFSLL Outbound Application Integration is used to send outbound requests and notify external systems.

The requests and notification messages generated by OFSLL can be text or XML messages. These messages are defined as per the format expected by external systems.

5. Post-Installation

5.1 Desktop Security

5.1.1 Application of Security Patches

Oracle Financial Services highly recommends the following:

- Browsers should be upgraded whenever newer versions are released, for they often include new security features. Additionally, in-built security features of browsers should not be turned off.
- Security patches issued by the Operating System vendor should be applied regularly.
- Updates to anti-virus software and anti-spyware programs should be applied regularly.

Additionally, it is recommended that major upgrades such as browser upgrades and Operating System service packs be tested for impact on business continuity.

5.1.2 Hardening the browser

OFSLL is certified for usage in different browsers. Please refer the respective release documents on the versions of browsers on which OFSLL has been certified. Each of these browsers provide recommendations from a security perspective and customers are encouraged to employ the recommendations provided by them.

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

- 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 autocomplete options to Disabled. This will prevent inadvertent caching of data keyed by users.

Customers are encouraged to employ the security recommendations provided by each browser vendor.

5.1.3 Terminal Lockout Policy

Oracle Financial Services recommends that a terminal lockout policy be put in place to automatically lockout unattended PC sessions after a certain duration. This is primarily because the application 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 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 screen savers.

5.1.4 Access Control

A non-administrative user should not have access to browser memory dump path (Windows -- C:\Users\<windows user>\AppData\Local\Temp) on the server. This will prevent any malicious user or software from scanning the passwords from the browser memory dumps.

5.2 Oracle Financial Services Lending and Leasing Controls

5.2.1 Overview

This chapter describes the various programs available within the application, 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.

5.2.2 Disable Logging

It is recommended that the debug logging facility of the application be turned off, once the system is in production.

1. The application/ADF logging can be disabled by configuring the ADF Logging from Oracle Fusion Middleware Enterprise Management Console.
2. The webservice logging can be disabled by not passing the java VM (Djava.util.logging.config.file) in the start-up script.
3. The database logging can disabled by disabling CMN_DEBUG_METHOD CMN_DEBUG_LEVEL in system_parameters

5.2.3 Audit Trail Report

A detailed Audit Trail is maintained by the system on all the activities performed by the user from the moment of login. This audit trail lists all the functions invoked by the user, along with the date and time. The program reports the activities, beginning with the last one. It can be displayed or printed. The records can be optionally purged once a printout is taken. This program should be allotted only to the Security Officer.

5.2.4 Security Violation Report

This program can be used to display or print the Violation Report. The report gives details of exceptional activities performed by a user during the day. The difference between the Violation Report and the Audit Trail is that the former gives details of all the activities performed by the users during the day, and the latter gives details of exceptional activities, for e.g. forced password change, unsuccessful logins (enabled through Weblogic server), User already logged in, etc. The details given include:

- Time
- The name of the operator
- The name of the function
- The ID of the terminal
- A message giving the reason for the login

The system gives the Security reports a numerical sequence. The Security Report includes the following messages:

5.2.4.1 Sign-on Messages

Message	Explanation
You have previous open logins.	The user has already logged into the system and is attempting a login through a different terminal.
Incorrect User name or Password	An incorrect user ID or password was entered.
Incorrect User name or Password	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).

5.3 Display/Print User Profile

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

- The user responsibility
- The log-in user company and branch
- The time of the last login

5.4 Clear User Profile

A user ID can get locked into the system due to various reasons like an improper logout or a system failure. The user can be reset by enabling the user through Weblogic Console.

5.5 Change User Password

Users can use this function to change their passwords.

- A user password should contain a minimum of eight characters.
- It should be different from the current and two previous passwords.
- The program should prompt the user to confirm the new password when the user will have to sign-on again with the new password.
- Users need to change the default password during their first login.

5.6 List of Logged-in Users

The user can run this program to see which users are in use within the application at the time the program is being run. The information includes the following:

- The ID of the user
- The login time

6. Generic Information

6.1 User Management

The application provides for the creation of new users through Weblogic Console. Users should be created by the administrator and details to be conveyed to the user manually. On the first login, the user is forced to change the password. The password is hashed iteratively after being appended with a randomly generated salt value. The hashing algorithm used is SHA-512.

User privileges are maintained by user group. The user group details captured in the weblogic are mapped to the respective users in the User and Groups weblogic screen. The user is provided access to various modules in the application based on the user-user group mapping.

6.1.1 Access Enforcement

In the application, one can manage the user access rights in the following ways.

6.1.2 Managing Access Rights based on User Group

Based on the user ID-user group mapping, the user access to various modules in can be restricted. For example, a user with the collector role may be provided access only to collection module.

6.1.3 Managing Access at Menu Level

One can restrict the access rights at the Menu level. Simply create a new group and assign required Menu level authorization keys to the group.

6.1.4 Managing Access Rights at Tab Level

One can restrict the access rights at the Tab level. Simply create a new group and assign required tab level authorization keys to the group.

6.1.5 Managing Access Rights at Button Level

One can restrict the access rights at the Button level. Simply create a new group and assign required Button level authorization keys to the group.

6.1.6 Managing Access Rights at Queue Level

One can restrict the access rights at the Queue level. Simply create a new group and assign required Queue level authorization keys to the group.

6.1.7 Information Flow Enforcement

The application validates the request XML files. If malicious data entry is found in the body of the XML file, such files are filtered out from further processing. The Java classes in the front end calls the back end PLSQL packages for further processing.

PLSQL level validations are in place in the database server. Exclusive use of bind variables and calls to Oracle's DBMS_ASSERT package sanitize the data. The passwords and other important details are encrypted using AES256 logic.

6.1.8 Separation of Duties

The application login is controlled based on the user groups assigned to the user profiles. Based on the groups assigned, a user is allowed access the functionalities in the application. For example, if the group assigned to a user profile is 'Service Agent', that user will be able to access only the Servicing module of the application.

6.1.9 Least Privilege

The application by default assigns no groups to a user. This is zero privilege from a functional perspective. With no groups assigned, a user cannot view the menu list as everything will appear to be blank. The system administrator has to explicitly map the groups to a user he creates. If a user is created by copying an existing user, then the new user will have the groups assigned to it from the user it has been copied. Nevertheless, it's recommended to map users to specific groups as per his job grade, designation etc.

6.1.10 Continuous Monitoring

user_logins table in the application record/archive the user login details every time a user logs in and logs out or when the session expires.

On the security front, the application works in conjunction with Oracle Platform Security Services (OPSS). One may use HP Web Inspect, a dynamic application security testing software for assessing security vulnerabilities. This tool is an automated and configurable web application security testing tool that mimics real-world hacking techniques and attacks.

CA Site Minder provides the enterprise-class secure Single Sign-On (SSO) and Web access management that authenticates users and controls access to Web applications and portals across Internet and Intranet Applications. It enables the secure delivery of essential information and applications to users, partners, suppliers and customers via secure SSO.

6.1.11 Information System Backup

As part of information system backup, the following periodic activities are recommended.

- Backup database related files such as data files, control files, redo-logs, archived files, init.ora, config.ora etc. at the end of day.
- Take online backup of archived redo-log files periodically.
- Do complete export of database and the application once in a week and store it off-site.
- Take complete backup of the Oracle directory periodically, excluding the database related files.
- When the database is huge, do incremental exports (delta or differential exports) and take online table-space backups.
- Use RMAN secure backup to ensure that the backups stolen from the production/ deployed system cannot be restored in another remote system. Additionally, you may use data masking, a feature offered by Oracle Enterprise Manager, to move the data from the production environment to a test environment. Both the activities mentioned in this step are crucial steps towards securing confidential customer data.
- Store the database backups for the required period as per the regulations and bank's history retention policies. Store these backups securely with access given only to authorized users.

6.1.12 Privacy Controls

The application avoids clickjacking and frame spoofing attacks using weblogic configuration. Privacy control and content type has also been placed.

6.1.13 Transmission Integrity and Confidentiality

Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are cryptographic protocols that provide communication security over the Internet. These transport protocols use asymmetric cryptography for authentication of key exchange, symmetric encryption for confidentiality and message authentication codes for message integrity. Application users are recommended to use SSL.

Http Only flag is included in a Set-Cookie HTTP response header. With that, from a browser, a particular cookie should only be accessed by the server. Any attempt to access the cookie from a client script is strictly forbidden.

6.1.14 Password Management

Certain user password related parameters should be defined at the organization level. These parameters will apply to all the users of the system. Examples of such parameters are the number of invalid login attempts after which a user-id should be disabled, the maximum and minimum length for a password, the interval at which the password should be changed by every user, etc.

6.1.14.1 Invalid Logins

The maximum number of allowable invalid login attempt made by a user can be configured through weblogic. Each user accesses the system through a unique User ID and password. While logging on to the system, if either the User Id or the Password is wrong, it amounts to an invalid login attempt.

6.1.14.2 Specifying Parameters for User Passwords

Password Length (characters)

The range of length (in terms of number of characters) of a user password can be set. 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.

Force Password Change after

The password of a user can be made valid for a fixed period after which a password change should be forced. After the specified number of days has elapsed for the user's password, it is no longer valid and a password change is forced. The number of calendar days defined will be applicable for a password change of any nature - either through the 'Change Password' function initiated by the user or a forced change initiated by the system.

Minimum Days between Password Changes

The minimum number of calendar days that must elapse between two password changes can be configured. After a user has changed the user password, it cannot be changed again until the minimum numbers of days you specify here have elapsed.

Intimate Users (before password expiry)

The number of working days before password expiry can be configured, which is used to display a warning message to the user. When the user logs into the system (the stipulated number of days before the expiry date of the password), a warning message will continue to be displayed till the password expires or till the user changes it. By default, the value for this parameter is two (i.e., two days before password expiry).

6.1.14.3 Placing Restrictions on User Passwords

Application allows placing restrictions on the number of alpha and numeric characters that can be specified for a user password.

Maximum Consecutive Repetitive Characters

The maximum number of allowable repetitive characters occurring consecutively, in a user password can be specified. This specification is validated whenever a user changes the user password, and is applicable for a password change of any nature - either through the 'Change Password' function initiated by the user or a forced change initiated by the system.

Minimum Number of Special Characters in Password

Application allows defining minimum number of special characters allowed in a user password. The system validates these specifications only when a user chooses to change the password.

Minimum Number of Numeric Characters in Password

Likewise, application allows defining the minimum number of numeric characters allowed in a user password. The system validates the password only when a user chooses to change his password.

Minimum Number of Lower Case Characters in Password

The minimum number of lowercase characters allowed in a user password also can be configured. The allowed lower case characters are from the US-ASCII character set only. The system validates these specifications only when a user chooses to change the password.

Minimum Number of Upper Case Characters in Password

The minimum number of upper case characters allowed in a user password can be configured. The allowed upper case characters are from the US-ASCII character set only. The system validates these specifications only when a user chooses to change the password.

6.1.14.4 Password Restrictions

Application allows defining a list of passwords that cannot be used by any user of the system in the bank. This list, called the Restrictive Passwords list.

The user name policies available from weblogic for this are:

- Reject if Password Contains the User Name
- Reject if Password Contains the User Name Reversed

7. Security Features

7.1 Introduction

This chapter contains the security features available in Oracle financial services lending and leasing.

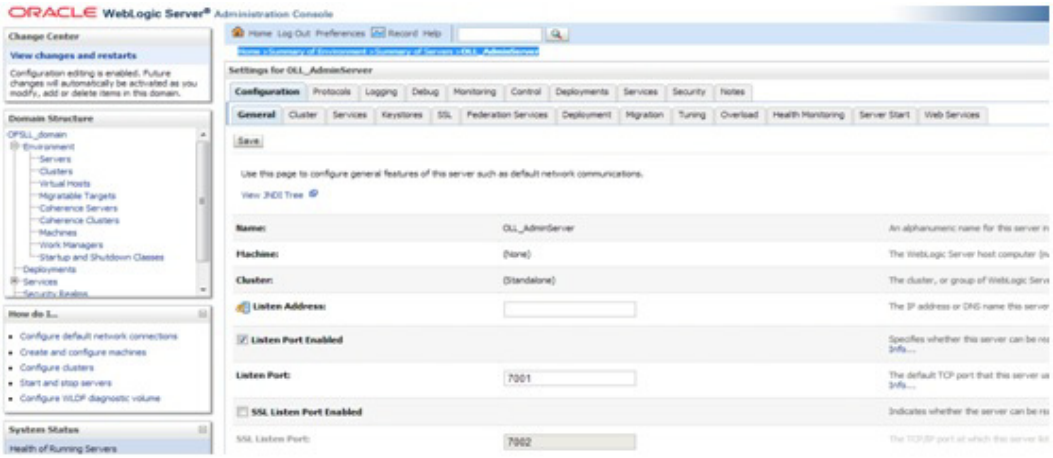
Level	Security Feature	Available by default	Optional	Configurable
Application Level	Transport level security by using SSL	Y	N/A	Y
	Authentication/Authorization	Y	N/A	Y
	Set-up HTTP server in front of weblogic	N	Y	Y
	Addition of WS policies	N	Y	Y
	Restricted access to log files	Y	N/A	Y
	Hashing of authentication passwords	Y	N/A	N
	Storage of credentials in CSF	N	Y	Y
	Storage of authentication certificates in password protected Keystore	Y	N/A	Y
Database Level	Database Auditing	N	Y	Y
	Secure Database back-up	N	Y	Y
	Sensitive data stored within database in encrypted format	Y	N/A	Y
	Database Access Control List	Y	N/A	Y
	File I/O permissions in DB runtime	Y	N/A	Y
	Socket I/O permissions in DB runtime	Y	N/A	Y
	Storage of certificates and credentials in password protected oracle wallet	Y	N/A	Y

Server Level	File system access to appropriate user	Y	N/A	Y
	Access/Usage of protected ports	Y	N/A	Y
	IP filtering	N	Y	Y

7.2 Additional Recommendations

To enable/add transport level security by using SSL:

1. Select Environments > Servers > <Server in which application is Deployed>
2. Select the 'SSL Listen Port Enabled' check-box and input the port number in the SSL Listen port.



3. Authentication/Authorization: As mentioned in application installation document.
4. Set-up HTTP server in front of weblogic:
   - http://docs.oracle.com/cloud/latest/fmw122100/CNFGD/web_server.htm#CNFGD192
   - http://onlineappsdba.com/index.php/2009/09/23/configure-oracle-http-server-infront-of-oracle-weblogic-server-mod_wl_ohs/
5. Addition of WS policies: As mentioned in the Webservices installation document.
6. Restricted access to log files: This can be achieved by granting right permissions to files and folders.
7. Hashing of authentication passwords: The passwords stored in the weblogic are hashed by weblogic. Developers don't have control.
8. Storage of authentication certificates in password protected Keystore: http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html
9. Database Auditing: http://docs.oracle.com/database/121/DBSEG/auditing.htm#DBSEG1023
10. Secure Database back-up: http://docs.oracle.com/database/121/BRADV/toc.htm
11. Database Access Control List: http://docs.oracle.com/database/121/ADXDB/xdb21sec.htm#ADXDB2400
12. File/Socket IO runtime Permissions: http://docs.oracle.com/database/121/JJDEV/chten.htm#JJDEV10000
13. Oracle Wallet: http://docs.oracle.com/database/121/DBIMI/walet.htm#DBIMI160

14. File system access to appropriate user: This can be achieved by granting right permissions to files and folders.
15. Access of protected ports: http://docs.oracle.com/cd/E24628_01/install.121/e24089/firewalls.htm
16. IP Filtering: http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html