Oracle FLEXCUBE Direct Banking

Security Guide
Release 12.0.3.0.0

Part No. E52543-01

April 2014

Security Guide
April 2014

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/

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.
The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1. Preface

1.1 Intended Audience

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

1.2 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.

1.2.1 Read Sections Completely

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

1.2.2 Understand the Purpose of this Guidance

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

1.2.3 Limitations

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

1.2.4 Test in Non-Production Environment

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

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

1.3 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.

1.3.1 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.

1.3.2 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.

1.3.3 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.

1.3.4 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.

1.3.5 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.

COMMENTS

Please provide comments concerning the improvement of this solution though support channel

About Oracle Software Security assurance refer below link:

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

1.4 Structure

This document is organized into following sections

Section 1: *Preface section of document*.

Section 2: *Abbreviation provides information for terms use in this document.*

Section 3: *Introduction provides information about possible security threats for Oracle Flexcube Direct Banking.*

Section 4 : *External Or Infrastructure Security provides information about external security mechanism.*

Section 5 : *Web Server Security provides information about Application server's security while handling tranction request.*

Section 6 : *Configuring FCDB Securely provides information for securing bank admin password, propety files data security etc.*

Section 7 : *Database Security provides information about database level security.*

Section 8 : *Browser Security provides information web browser level security like encryption, ciphor strength support etc.*
Section 9 : *Recommendation for the usage Oracle Flexcube Direct Banking.*

1.5 Related Information Sources

For more information refer to the following documents:

Oracle FLEXCUBE Direct Banking Installation Guide	Refer the same for the secure installation of the Oracle FLEXCUBE Direct Banking application.
Oracle FLEXCUBE Direct Banking Database Setup Guide	Refer the same for the secure installation of the Oracle FLEXCUBE Direct Banking database.

2. Abbreviations

FCDB / FC DB / FC Direct Banking	Oracle FLEXCUBE Direct Banking
Java EE / JEE	Java Enterprise Edition
Java SE / JSE	Java Standard Edition
DBA	Database Administrator
XML	Extensible Markup Language
XSL	XML Stylesheets
TCP	Transmission Control Protocol
HTTP	Hyper Text Transmission Protocol
HTTPS	Secured Hyper Text Transmission Protocol
SSL	Secured Socket Layer
IDS	Intrusion Detection System

3. Introduction

Oracle FLEXCUBE Direct Banking is a multi channel e-banking platform with support for customer touch points like Internet, Mobile Phones and PDAs.

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

The security threats for Oracle FLEXCUBE Direct Banking can be from various internal and external sources within the financial institution. While the internet banking channel has always been exposed to the vulnerabilities originating from the internet via traditional threats and attacks, the internal breakdown of controls and measures is also responsible for critical information being available exposed to un-authorized users.

Internal security threats can be leakage of critical credential information via inappropriate storage or access control mechanism within and around the application.

The Oracle FLEXCUBE Direct Banking platform relies on multiple levels of security, to secure the infrastructure as well as the application, which are broadly classified as follows.

External or Infrastructure Security

The External Security or Infrastructure Security is the implementation of appropriate measures and controls around the environment / infrastructure in which the Oracle FLEXCUBE Direct Banking solution is deployed. This helps to not only mitigate any internal and external attacks but helps detection and monitoring of exposures very early for the financial institution to manage the same effectively.

Application Security

Application security refers to application security features which provide appropriate controls and access privileges based on authenticated credentials. The application security covers the deployment of the application, security features within the application which protect against security attacks and vulnerabilities.

4. External Or Infrastructure Security

External security mechanisms are typically outside the direct scope of Oracle FLEXCUBE Direct Banking but are highlighted here to identify the various options available during an installation. The key components in the Oracle FLEXCUBE Direct Banking infrastructure are the Web Server, Application Server and the Database Server. Optionally, other components like Oracle AQ, WebSphere MQ etc. may be used for interfaces which are only on a case to case basis.

The External Security deals with the hardening and securing of the infrastructure using industry standard tools and techniques for the end to end secure deployment of Oracle FLEXCUBE Direct Banking.

The implementation and support of the External Infrastructure Security is the complete responsibility of the financial institution implementing the Oracle FLEXCUBE Direct Banking platform. The security policies of the bank, deployment variations, data center locations, network connectivity will determine the extent of security required for the end to end deployment of the infrastructure.

4.1 Firewalls

Appropriate Firewalls are to be implemented to create multiple De-Militarized Zones (DMZs) in the internet banking deployment architecture. Firewalls act as the primary defense mechanism against any unauthorized access from the Internet or any other internal networks.

Multiple DMZ creation is recommended wherein the Web Server, Application Server and Database Server reside in different DMZs.

Security practices suggest the use of firewalls from different vendors for the internet facing firewall and the internal firewalls. This makes the intrusion difficult via multiple levels of different firewalls. The final decision of the firewall deployment should be performed by the bank based on their internal security guidelines.

Additional firewalls can be introduced, to enhance security, within the Oracle FLEXCUBE Direct Banking deployment.

The Oracle FLEXCUBE Direct Banking architecture allows the logical and physical segregation of the Presentation, Channel Management and Business Layers which allows additional DMZs to be configured.

VPN is used to provide appropriate access to external parties or vendors to the infrastructure based on appropriate authentication mechanisms. This is typical for monitoring activities and only restricted used access to be provided to the production infrastructure. Direct VPN access to the production infrastructure should be avoided to prevent unauthorized access.

4.2 Intrusion Detection Systems

Intrusion detection is the methodology by which undesirable or aberrant activity is detected on a host or a network. The two main approaches to intrusion detection systems are host-based (HIDS) and network-based (NIDS). A combination of both provides the most complete coverage for a secured deployment of Oracle FLEXCUBE Direct Banking.

4.2.1 Network Based IDS

Network-based intrusion detection systems run on one or several critically placed hosts and view the network as a whole. NIDS use NICs running in promiscuous mode to capture and analyze raw packet data in real time. Most NIDS use one of two methods to identify an attack, statistic anomaly detection or pattern-matching detection.

4.2.2 Host Based IDS

Host-based intrusion detection systems are installed on each host system (Web Server, Application Server and Database Server of FLEXCUBE Direct Banking) and look for attacks directed directly at the host. Most HIDS employ automated checks of log files, file checksums, file and directory permissions, local network port activity, and other basic host security items and offer the benefit of being able to detect attacks local to the machine or on an encrypted or switched network where a NIDS might have issues.

4.3 Network Level Security

Network level security provides protection against external attacks on the network. The network level security can be enabled using appropriate IPSec Encryptors, WAN Encryptors etc. Implementation of 128-bit SSL between the Client Browser to Web Server and from Web Server to Application Server also provides additional security of the data on the network.

4.4 Infrastructure Hardening

Infrastructure hardening refers to individual hardening of all components within the Internet banking infrastructure like Networks, Servers, Firewalls, Routers, and Load Balancers etc. The hardening process should follow industry standard guidelines and deactivate processes which are not required, provide appropriate access control.

Also server hardening can be augmented by use of appropriate antivirus software for protection against Virus and Trojan horse attacks.

5. Web Server Security

Oracle FLEXCUBE Direct Banking uses the Web Server as the entry point for all transactional requests.

The following configurations are valid for the Apache Web Server as well as any variants like IBM HTTP Server or Oracle HTTP Server based on the same.

The Apache / IBM HTTP / Oracle HTTP Server documentation should be referred for the details of the server directives used within this section for the various configurations. If multiple virtual host setups are done, then the contexts within which the configurations have to be set have to be chosen accordingly.

5.1 Remove Server Information

The following configuration should be done in the http.conf configuration file of Apache Web Server to remove the exposure of the Apache version used by the financial institution.

Set ServerTokens Prod

Set ServerSignature off

5.2 Remove Redundant Directives

The following configuration should be done in the http.conf configuration file of Apache Web Server to remove the exposure of the Apache version used by the financial institution.

5.3 Remove Server Manuals

The server manuals from the manuals directory on the server should be removed or protected with appropriate access control to be not allowed from the internet.

5.4 Protect Administrative Information

The Web Server contains a number of web pages which provide administrative functionality and information. These pages offer information about various services, the server’s state and its configuration. These pages must be restricted or disabled in a production system.

The httpd.conf should be configured to allow access or limit the web pages and static content accessible from the internet.

Include the following restrictions / allowances as appropriate for the required URLs

```
<Location ~ "/B001/ENULogin.jsp">

Order allow, deny

Allow from all

Deny from localhost <list of UNTRUSTED / INTERNAL IPs>

</Location>
```

This is only an example.

The exact content cannot be provided here since the static content may change for the implementation based on the bank’s branding strategy and guidelines.

5.5 Prevent Search Engine Indexing

The robot exclusions should be enabled for the web servers.

Refer http://www.robotstxt.org/wc/robots.html for more information.

The following would indicate that no robots are allows to access the site.

```
User-Agent:  *

Disallow: /
```

The following META tag can also be added in the static HTML files to indicate to the robots to not index the content of the page or scan it for further links. The META tag should be placed in the head section of the HTML page.

```
<META NAME=”ROBOTS” CONTENT=”NOINDEX, NOFOLLOW” />
```

5.6 Configure SSL Versions

The protocols supported for secure communication can include TLS V1.2. The use of TLS V1.2 is recommended to be setup within httpd.conf file. This may disallow certain older browsers that the customers would be using but allow greater security in the SSL communication.

```
SSLProtocol -all +TLSv1.2
```

5.7 Configure SSL Cipher Specifications for 128-bit and above

The SSL Ciphers to only support 128-bit and 168-bit encryption are provided below. The minimum SSL Cipher should be 128-bit and any cipher less than 128-bit should be not allowed. The following to be added in the httpd.conf file.

TLSV1

2F TLS_RSA_WITH_AES_128_CBC_SHA AES SHA (128 bit)

35b TLS_RSA_WITH_AES_256_CBC_SHA AES SHA (256 bit)

```
SSLCipherRequire 3A

SSLCipherRequire 34

SSLCipherRequire 35

SSLCipherRequire 2F

SSLCipherRequire 35b
```

Refer http://www-01.ibm.com/support/docview.wss?uid=swg21112074 for more details for IBM HTTP Server.

5.8 Block TRACE access

Prevent the TRACE HTTP method for being invoked from the internet. The following configuration should be added to the httpd.conf file.

```
###Added to prevent HTTP TRACE

RewriteEngine on

RewriteCond %{REQUEST_METHOD} ^TRACE

RewriteRule .* -[F]
```

6. Configuring FCDB Securely

6.1 Securing FCDB Bank Admin Password

The FCDB Bank Admin password should be secured after the initial setup. The financial institution should create additional administrative users for their operational needs.

6.2 Set key pair for login password encryption

The login password is encrypted on the client using the RSA algorithm. The public key – private key pair for this is configurable.

To generate a new key-pair the following steps should be followed:

1. Change directory to <FCDB_INSTALL_HOME>/system/home
2. Execute the below command:
   keytool -genkeypair -alias fcdb -keystore fcdb_keystore.jks -storetype JKS -keyalg RSA -keysize 2048
3. Follow the on-screen instructions to create or update the keystore.
4. Execute the below command to extract the public certificate (for use with iOS devices):
   keytool -export -alias fcdb -keystore fcdb_keystore.jks -file fcdb_sign.cer

The keystore is to be placed in <FCDB_INSTALL_HOME>/system/home.
For the internet channel, the public key gets sent by the server in the login page response.
For the mobile applications, the public key gets distributed in the mobile application itself, as a property.

Please note that the alias for the key pair is “fcdb”, and therefore do not change that.

The keystore password and the private key password are to be stored in the properties file fcat.properties in the following properties.

FCDB.KEYSTORE.PASSWORD
FCDB.KEY.fcdb.PASSWORD

We recommend that the 2 passwords be kept different.

Also, on a Unix system we recommend that the folder permissions for <FCDB_INSTALL_HOME>/system/home be set to 600 post key generation.

6.3 Configuring password hashing iterations

The user login password is stored in the database after applying a one way hash function on it. The number of hashing iterations to be performed is configurable and is stored in the property HASHING.ROTATION.COUNT in fcat.properties file.

If no value is mentioned, it is assumed to be 1000.

In case it starts to affect performance, then a lower value can be configured.

6.4 Encrypt fcat.properties File

The fcat.properties file should be encrypted using the following batch (.bat / .sh) file

securepropertiesfile <<ClearText Properties File>> <<Output Encrypted Properties File>>

The ClearText properties file should be backed up and securely stored.

Note: Please set file permissions in such a way that only authorized users have access to these files on the operating system.

6.5 Branding Considerations

The WAR file should be cleaned after the Branding exercise for any unwanted static content, JSP files etc. The images, JS, CSS files that are not required should be removed before the final deployment.

6.6 Remove Copyrights from Static Web Content like HTML, CSS, JS Files

The JS, CSS and Static HTML files can contain copyrights and modification history information. This should be removed from the static content from within the WAR files to avoid any social engineering attacks.

No modification history and copyrights should be maintained in the static content.

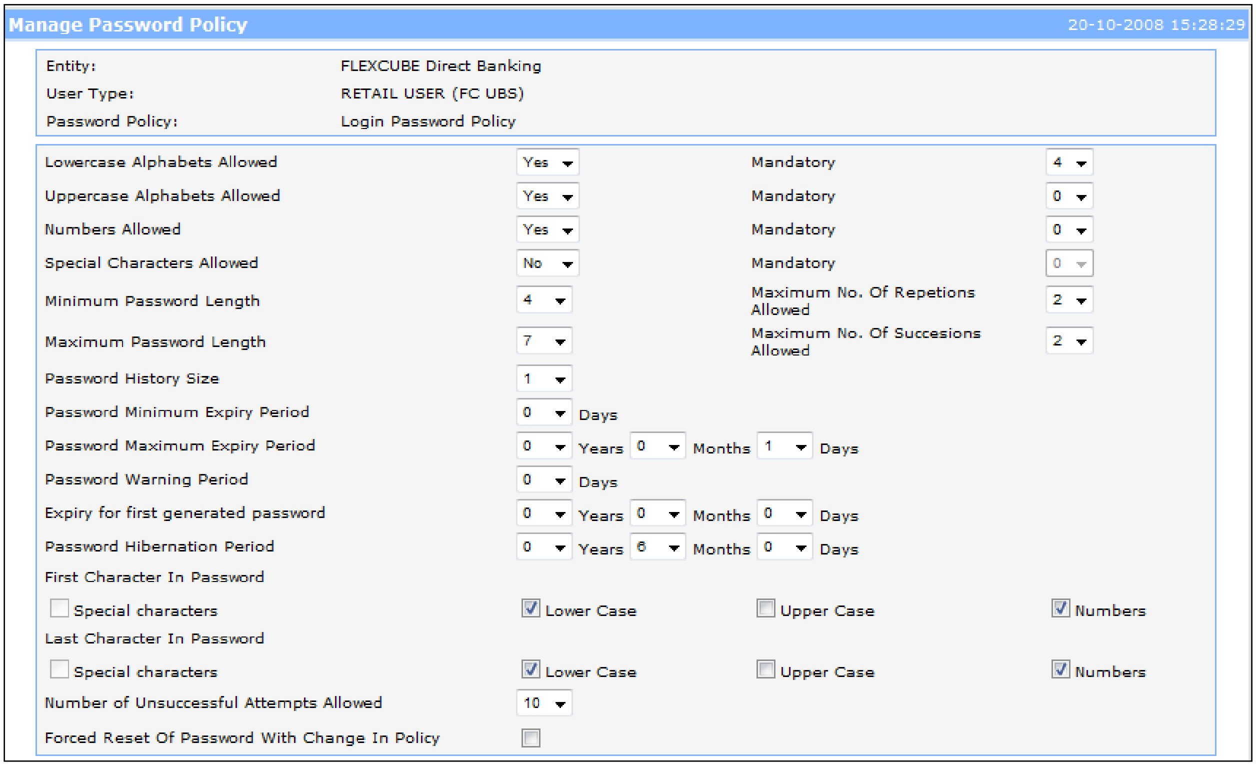
6.7 Disable Directory Browsing

Directory browsing feature should be disabled in the WAR files if newly built.

6.8 Password Management

Password Management refers to management of the credential information for authentication or authorization purposes within the system. This is typically used in reference to management of passwords used within the system.

The Manage Password Policy function should be used by the administrators to setup the appropriate password policy rules. This is available in the administration of the Oracle FLEXCUBE Direct Banking application.



Manage Password Policy 20-10-2008 15:28:29

Entity:	FLEXCUBE Direct Banking
User Type:	RETAIL USER (FC UBS)
Password Policy:	Login Password Policy

Lowercase Alphabets Allowed	Yes	Mandatory	4
Uppercase Alphabets Allowed	Yes	Mandatory	0
Numbers Allowed	Yes	Mandatory	0
Special Characters Allowed	No	Mandatory	0
Minimum Password Length	4	Maximum No. Of Repetions Allowed	2
Maximum Password Length	7	Maximum No. Of Succesions Allowed	2
Password History Size	1		
Password Minimum Expiry Period	0 Days		
Password Maximum Expiry Period	0 Years 0 Months 1 Days		
Password Warning Period	0 Days		
Expiry for first generated password	0 Years 0 Months 0 Days		
Password Hibernation Period	0 Years 6 Months 0 Days		
First Character In Password			
□ Special characters	☑ Lower Case	□ Upper Case	☑ Numbers
Last Character In Password			
□ Special characters	☑ Lower Case	□ Upper Case	☑ Numbers
Number of Unsuccessful Attempts Allowed	10		
Forced Reset Of Password With Change In Policy	□		

The following parameters can be configured.

- Minimum and Maximum Length (We recommend a minimum of 8 characters)
- History Size
- Expiry period
- Warning period
- Min Number of Digits required (We recommend at least 1 digit to be made mandatory in the password)

- Min Number of Lower case alphabets required (We recommend at least 1 lower case alphabet to be made mandatory in the password)
- Min Number of upper case alphabets required (We recommend at least 1 upper case alphabet to be made mandatory in the password)
- Special characters allowed (We recommend at least 1 special character to be made mandatory in the password)
- Maximum unsuccessful login attempts (We recommend that this number be not more than 10)

Additionally, a custom password policy can be developed and plugged for alternate password rules.

6.9 Setup Audit Logs

Non Repudiation ensures that a transaction can be tracked and data cannot be renounced or a transaction denied. This is a critical security feature of any financial application on the internet. The audit log feature should be enabled for all the respective transactions within Oracle FLEXCUBE Direct Banking.

This is enabled using the AUDITLOGREQUIRED flag in the MSTCHANNELATS for the required interactions.

Support has been added to mask sensitive data fields in the logs inserted into “auditlog” table.

The same support also takes care of masking sensitive data in ChannelController logs and FLEXCUBEConnectHostPlugin logs.

The configuration for the same has to be done in the database table “appldata”.

Against the DATANAME and DATAVALUE values of “TRUNCATEFIELD”, the VALUESTRING column is used to mention the idrequest for which fields are to be masked and the actual fields to be masked.

6.10 XSS validation framework

This framework is in place so as to negate the threat of Cross Site Scripting attacks by using the white listing method of input validation.
In the database (Admin schema) there is a table named `WHITELISTLENGTHMAP` in which one can configure the regular expression validation for every parameter for every request.

The table is defined as follows:

```
WHITELISTLENGTHMAP
(
  PARAM_NAME       VARCHAR2(50),
  PARAM_LENGTH     NUMBER(10),
  PARAM_REGEX      VARCHAR2(50),
  PARAM_REQUESTID  VARCHAR2(20)
)
```

For example: If you want to create a validation that the field *fldbranchlocation* being sent in the request RRLOB02 needs to be a maximum of 50 characters long and can be alphanumeric only, then the following script needs to be fired:

```
insert into whitelistlengthmap values('fldbranchlocation','50','[,0-9A-Za-z -
]*','RRLOB02');
```

White List validation already exists for the following fields across all requests. This validation is built into the base code itself. One need not configure these using the above framework.

```
fldLangId
fldSessionId
fldDataId
fldDeviceId
fldSectionId
fldServiceType
```

In case one or more configurations get missed, there is a fall back on the black listing method of input validation. This is again a configuration but a generic one. This configuration applies to any field for which white listing has not been configured.
The Black Listing approach simply identifies a set of characters that are to be removed from any incoming data field and replaces each such character with another configured character.

The configuration is placed in <entity>.xml file and the properties are named as
`FCAT.FILTER.EXP.TXN` and
`FCAT.REPLACE.EXP.TXN`

The set of malicious characters are placed in `FCAT.FILTER.EXP.TXN`
The corresponding set of characters that replace the above characters is placed in `FCAT.REPLACE.EXP.TXN`

Note: For the transaction "Submit Application" in Originations, the 2nd screen being painted contains the same HTML response as the previous screen. In IE version 8+ there is an in-built XSS filter that disallows the rendering if response is the same as the request. Therefore to overcome this, the value of the header **X-XSS-Protection** has been set to 0. It's applicable only to that screen.

6.11 FCDB – Host (FCUBS) authentication

If the backend host (FCUBS only) switches on authentication at its end, then FCDB also has a configuration in place to send the password along with the user name in the request XMLs that are sent to the host.
In the database table mstproperties (Admin Schema), 2 properties need to be set

1) <ID_ENTITY>.HAS.HOST.PASSWORD → Y
2) <ID_ENTITY>.HOST.PASSWORD → <ACTUAL PASSWORD>

Sample Database Scripts:

insert into mstproperties (IDSERVER, PROPNAME, PROPVALUE, ENABLED, ISMODIFIED, DATEFFECTIVE, DATMODIFIED, ISGUIENABLED) values ('ZZ', '<ID_ENTITY>.HAS.HOST.PASSWORD', 'Y', 'Y', 'Y', to_date('30-01-2012 18:26:50', 'dd-mm-yyyy hh24:mi:ss'), to_date('30-01-2012 18:26:50', 'dd-mm-yyyy hh24:mi:ss'), 'Y');

insert into mstproperties (IDSERVER, PROPNAME, PROPVALUE, ENABLED, ISMODIFIED, DATEFFECTIVE, DATMODIFIED, ISGUIENABLED) values ('ZZ', '<ID_ENTITY>.HOST.PASSWORD', '<FCUBSPASSWORD>', 'Y', 'Y', to_date('30-01-2012 18:26:50', 'dd-mm-yyyy hh24:mi:ss'), to_date('30-01-2012 18:26:50', 'dd-mm-yyyy hh24:mi:ss'), 'Y');

6.12 Authorization Engine

FCDB is extensible in a way that new transactions can be built using the given framework in a non invasive development mode (NID mode), by the implementation team. Security from a functional perspective (data spoofing) can be provided by using the Authorization Engine framework.

There are different modules available; the complete list can be obtained from the table *mstmodules* in the Admin schema. The correct modules (based on your transaction) need to be configured in the table *mstinitauthmodules* in the Admin schema.

For example: The module ACM (Account Check Module) checks whether the source account sent in the request for the given transaction belongs to the logged in user or not.
The mapper class that will map the fields from the service request to the Authorization Module request is configured against your service in the table *mstservices*

6.13 Configuring secure SMTP

The product uses a framework to generate Email alerts to its users. The protocol to be used is SMTPS out of the box. We recommend that you use a secure protocol like SMTPS. The default port for secure SMTP is 465.

The port and the protocol, both are configurable in the Database in the table called mstproperties.

The properties have been named as follows:
ALERTNOTIFY_EMAIL.PROTOCOL
ALERTNOTIFY_EMAIL.PORT

7. Database Security

7.1 Configure Listener on a Non Default Port Number

By default, the TNS Listener receives service requests on TCP port 1521. Configure it to listen on another port number. Although not foolproof, this makes attacks more difficult.

7.2 Complex Password Setup

The password to the default accounts like SYS, SYSTEM etc. should be complex and securely stored by the bank. These are supposed to be maintained and managed by the financial institution implementing the Oracle FLEXCUBE Direct Banking solution.

7.3 Remove OS Authentication

This setting prevents the database from using an insecure logon protocol. The init.ora should contain REMOTE_OS_AUTHENT=FALSE

7.4 Restrict access to SQL Trace Files

The init.ora parameter _TRACE_FILES_PUBLIC grants file system read access to anyone who has activated SQL tracing. This should be set to False.

_TRACE_FILES_PUBLIC=FALSE

7.5 Remove Operating System Trusted Remote Roles

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

REMOTE_OS_ROLES=FALSE

7.6 Maintain and Secure Database Backups

The database backups should be securely 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.

7.7 Setup Database Security and Audit

The DBA should configure audit of the required activities like Schema Changes, Connection Access etc. for the database. This would allow analysis for any security violations within the database.
It is also recommended that Transparent Data Encryption (TDE) be done on the Database so as to secure the data files. Sensitive data such as Credit Card Numbers thus gets secured. TDE can be done in the following way for Oracle FCDB:

```
--create a folder structure $home/admin/($sid)/wallet
--restart the db

--set encryption key
alter system set encryption key authenticated by "<PASSWORD>";

--to open wallet
alter system set encryption wallet open authenticated by "<PASSWORD>";

--to close wallet
alter system set encryption wallet close;

--tde for column
ALTER TABLE  <FCDB ADMIN SCHEMA>.usercustextsystems
MODIFY (idcust ENCRYPT using 'AES192' NO SALT);

--tde for tablespace
CREATE
    TABLESPACE "SECURE"
    DATAFILE '<DBF File Location>' SIZE 1024M
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 64K
    SEGMENT SPACE MANAGEMENT  AUTO
    ENCRYPTION USING 'AES192'
    DEFAULT STORAGE(ENCRYPT);

--table movement to encrypted tablespace
alter table <FCDB ADMIN SCHEMA>.usercustextsystems  move tablespace SECURE;
alter table <FCDB ADMIN SCHEMA>.usercustrel  move tablespace SECURE;
alter table <FCDB ADMIN SCHEMA>.mstuser  move tablespace SECURE;
```

7.8 Database Setup and Configuration

The Oracle FLEXCUBE Direct Banking Database Setup requires the use of multiple database users for setup and maintenance of the application.

For further information on securing Oracle Database, please visit

http://www.oracle.com/pls/db112/portal.portal_db?selected=25&frame=

The privileges required for the two database users are referred as

DBA USER

```
grant DBA to FCDBADMIN_ROLE;
grant RESOURCE to FCDBADMIN_ROLE;
grant CONNECT to FCDBADMIN_ROLE;

/**to be granted in case DBA role needs to be revoked
grant create session to FCDBADMIN_ROLE;
grant create table to FCDBADMIN_ROLE;
grant create sequence  to FCDBADMIN_ROLE;
grant Create Database Link  to FCDBADMIN_ROLE;
grant Create Trigger  to FCDBADMIN_ROLE;
grant Create View  to FCDBADMIN_ROLE;
grant Create Procedure  to FCDBADMIN_ROLE;
grant Create Synonym to FCDBADMIN_ROLE;
grant Create Sequence  to FCDBADMIN_ROLE;
grant Drop Any Sequence to FCDBADMIN_ROLE;
**/
```

APPLICATION USER

```
grant connect to FCDBUSR_ROLE;
```

Individual privileges are granted to the APPLICATION USER on the database tables, views, synonyms etc during the database setup for a restricted access to specific objects only.

The multiple users allow a restricted user to be used from within the application enhancing security of the application. The complete database setup should be done using the instructions during the installation to allow a secured setup. The DBA user is only used for setup, configuration and maintenance but not used from within the application.

8. Browser Security

The browser used to should meet the minimum security requirements like 128-bit encryption and cipher strength support.

8.1 Browser Upgrades

It is important for users to use the latest versions of the browsers at all times to leverage on security fixes available in the browser.

8.2 Turn OFF Auto Complete

For kiosk machines, change Internet Explorer’s AUTOCOMPLETE settings. Browsers can automatically show previous values entered in the same form field. Although desirable for frequently accessed pages, for privacy and security reasons this feature should be disabled.

To turn OFF the Auto Complete feature in Internet Explorer:

1. Navigate through Tools -> Internet Options -> Content

2. From the Content tab, click the AutoComplete button.

3. Uncheck "forms" and "User names and passwords on forms".

Also, do not use the "Remember Password" function; This is a known security vulnerability.

The Oracle FLEXCUBE Direct Banking anyways enforces switching off of the Auto Complete Flag from within the pages but the above should be done as a good practice.

8.3 Clear Browser Cache and Exit Browser

Whenever a public or shared computer is used, it is essential that the user signs out when the internet session is finished. Once the user has signed out, they should delete the browser's cache and history this deleting any cached copies of the pages, if any, in the browser cache.

The browser should be closed after every internet banking session.

Note: FCDB uses a cookie JSESSIONID which contains a session identifier representing an active session of an FCDB user. If Cookies are disabled in the browser, the FCDB login will not work.

9. Recommendations for the usage of Oracle FLEXCUBE Direct Banking

1> During login, it is advisable to use the virtual keyboard if the computer in use is a public computer. In case of a private computer, the virtual keyboard need not be used.
The virtual keyboard is used to negate key logger software that record what keys are typed in. If a public computer has the key logger software installed, the login password is compromised.

2> During login, the “enter by hover” mouse method is again another security measure. It is to be used to enter passwords on public computers. Some key loggers are designed to take screen snapshots on every mouse click. To avoid falling prey to this, the virtual keyboard is designed to enter a character when the mouse hovers over a particular key.

3> A bank administrator must closely examine any repeated failed logins in the FCDB audit logs so as to identify potential brute force attacks.

4> Whenever a bank administrator locks/deactivates a user, it is recommended that he terminates any existing active session of the same user.

5> The role management mechanisms are used to implement separation of privileges, for example ensuring that the initiator of a transaction cannot also be the authorizer. Hence care must be taken when creating entitlements (rules) as well. Administrators should carefully follow their organization's separation of privilege polices when performing role management.

10. Cookies Used

FCDB uses a cookie JSESSIONID that contains a session identifier representing an active session of FCDB user. FCDB login will not work, if we disable Cookies in Browser.