EBI Application Security Standard

Prepared by,		Reviewed by,	Approved by,

Status : Published

Version : 1.0

Last update :

Prepared by :

Reviewed by :

Approved by :

Overview **1**
Application Security Standard **2**
Input Validation 2
Authentication 2
Session Management 3
Authorization 3
Error handling and logging 3
Communication Security 3
Data Protection 3
Data Encryption 4
WebView 4
SFTP 4
Penetration Testing 4

Overview

This document covers the application security standard to ensure the applications that banks use are safe and secured. The majority of Bank applications are developed by our engineer, therefore Bank needs to ensure that applications going to live in production are secure from known vulnerabilities.

Application Security Standard

Encompass a set of general software security coding principles and guidelines, in a comprehensive checklist format, that can be integrated into the development life-cycle. The goal is to prevent and minimize vulnerabilities that could be exploited by attackers and to ensure the confidentiality, integrity, and availability of applications.

Input Validation

Apply input validation on both the **Syntactic** and **Semantic** level, all validation must be done for trusted and untrusted systems which are then processed by the backend.

- Validate for expected data types
- Validate data range and length
- Validate all client provided data including all parameters, HTTP headers content, and URLs.
- Encode data to a common character set before validating
- If any potentially hazardous characters must be allowed as input, be sure that you implement additional controls like output encoding, secure task specific APIs and accounting for the utilization of that data throughout the application . Examples of common hazardous characters include: < > " ' % ( ) & + \ \' \

- If the standard validation routine cannot address the following inputs, then they should be checked discreetly
    - Check for null bytes (%00)
    - Check for new line characters (%0d, %0a, \r, \n)
    - Check for “dot-dot-slash" (../ or ..\) path alterations characters. In cases where UTF-8 extended
    - character set encoding is supported, address alternate representation like: %c0%ae%c0%ae/

Authentication

Authentication aims to verifying that an individual, entity or website is whom it claims for verification

- Require authentication for all pages and resources, except those specifically intended to be public.
- Used unique credential (e.g username & password, phone number), and token that used to identify every request with the application (include internal and external services).
- Used strong password with minimum password length is 8 characters which contains and combines with alphanumeric and symbols.
- Used a strong PIN with minimum PIN length of 6 digits with no repeating and consecutive number.
- Used strong cryptography one-way salted hashes for password
  The standard minimum is using SHA2.

Session Management

Process of securely handling user sessions

- Set session timeout, re-authenticate every 12 hours and terminate sessions after 15 minutes of inactivity
- Generate new session for any re-authentication
- Used random uuid for session generation
- Do not exposed session/token in URL, error messages, or logs

Authorization

Verifying the request to granting or denying specific requests

- Enforce least privilege by granting only the minimum privilege necessary
- Validate permission on every request

Error handling and logging

Verifying the request to granting or denying specific requests

- Do not disclose sensitive information, debugging or stack trace information in error response

- Implement generic error messages and use custom error pages
- Logging controls should support both success and failure of specified security events

Communication Security

- Used encrypted / secured communication (HTTPS) with minimum Transport Layer Security (TLS) 1.3 or TLS 1.2 with minimum encryption module.
    - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_AND_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS certificates should be valid and have the correct domain name, not expired, and installed with digital certificates when required
- Filter parameters containing sensitive information from the HTTP referer, when linking to external sites
- Internal applications or Dashboards have to use private connection or whitelisted IP Address such as VPN

Data Protection

- Implement least privilege, restrict users to only the functionality, data and system information that is required to perform their tasks
- Encrypt highly sensitive stored information, like authentication verification data, even on the server side.
- Do not include sensitive information in URLs request parameters.

Data Encryption

- Use strong encryption e.g. AES/GCM/256
- PII data should be encrypted, refer to this document

WebView

- Make sure using latest WebView versions available, as they often include security enhancements and bug fixes
- Enable safe browsing features provided by the WebView
- Implement secure communications, refer to Communication Security
- Used latest and stable library
- Enable security headers (CSP, X-frame-options, xss-protection, referrer-policy, and etc)
- No web caching for sensitive data on WebView
- Do not include sensitive information in URLs request parameters

SFTP

- SFTP server should follow EBI System Security Standard (hardening, security agents, vulnerability & patch management)
- Ensure connection to SFTP server is on a private IP network not over public or Internet
- Enforce whitelisting of Partner's IP address to ensure we only allow connections "to and from" recognized IP addresses or hosts
- Enforce authentication in SFTP server either with strong password or using PKI / X509v3 Digital certificates
- Ensure the SFTP server is isolated from the rest of the critical network
- Enable logging and monitoring in SFTP server.
- Ensure integrity of data files being pulled from related third parties.
- Ensure regularly backup data in SFTP server for at least 90 days for any risk or security investigations.

Penetration Testing

- For new applications, penetration testing needs to be performed to identify and remediate security vulnerabilities prior to application released to production.

For more information about secure coding practice, please refer to this document.