ITU DFS Security Recommendations Part 2: Securing Mobile Payment Apps

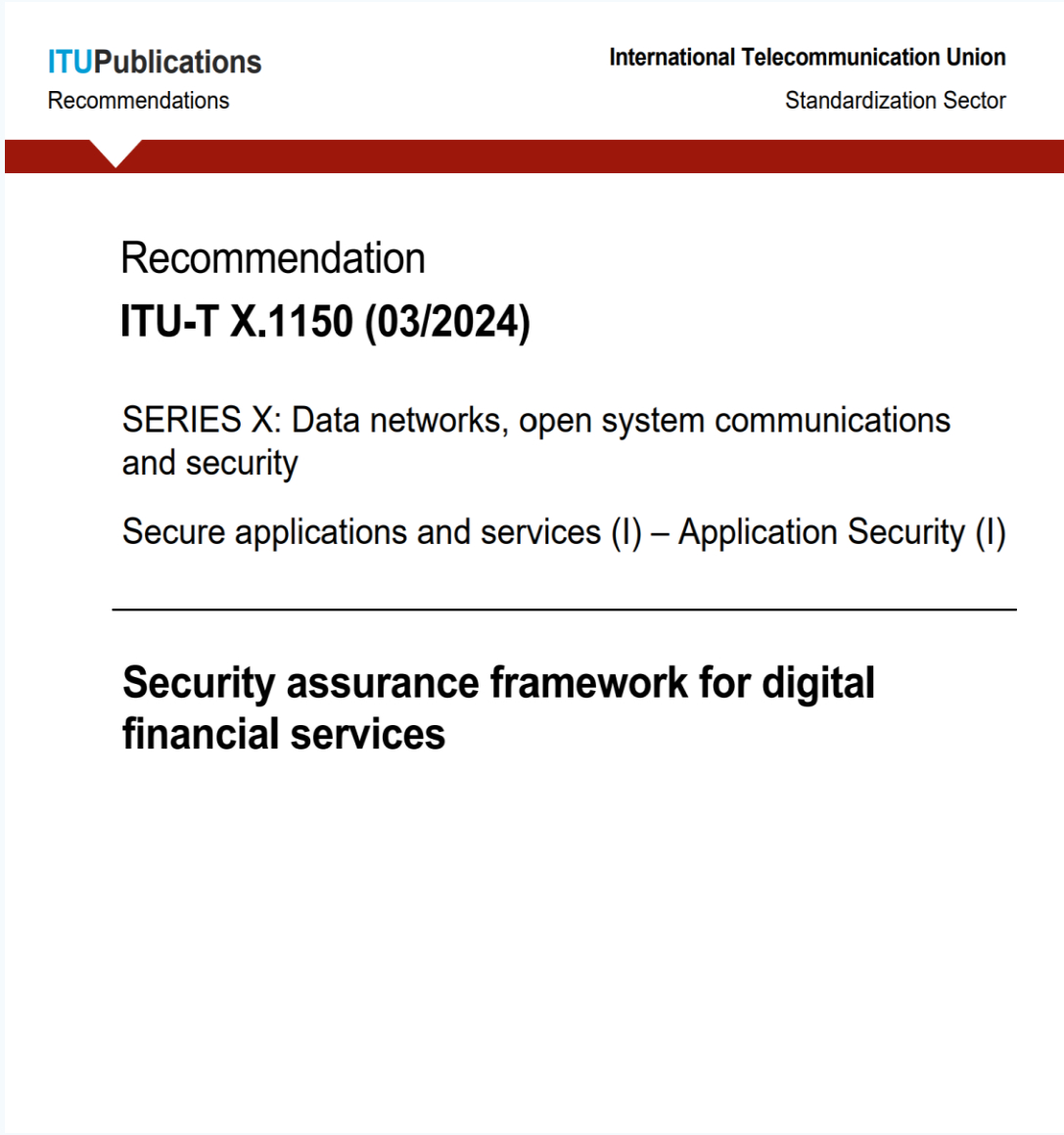
Negash Namrud, Project Officer, ITU

January 2025

http://www.itu.int/go/dfssl

Mobile Payment App Security Best Practices

- Contained in ITU-T Recommendation X.1150
- Draws upon:
  - GSMA study on mobile money best practices,
  - ENISA smartphone security development guidelines,
  - State Bank of Pakistan mobile payment applications security framework


ITUPublications
Recommendations

International Telecommunication Union
Standardization Sector

Recommendation
ITU-T X.1150 (03/2024)

SERIES X: Data networks, open system communications and security

Secure applications and services (I) – Application Security (I)

Security assurance framework for digital financial services

Mobile Payment App Security Best Practices

- Can be used as input to an app security policy by DFS providers to provide minimum security baselines for app developers and DFS providers as well as setting criteria for verifying compliance of apps

- Template considerations:
  i. device and application integrity.
  ii. communication security and certificate handling.
  iii. user authentication.
  iv. secure data handling.
  v. secure application development.

Device and Application Integrity

- Applications should thus use the mobile platform services to determine that they and the underlying platform have not been modified
- Remove any extraneous code that might have been added to the application during development
- On the server-side, determine whether the app is running in a high integrity state

Communication Security and Certificate Handling

- Apps should be making use of standardized cryptographic libraries
- TLS certificates should not be expired and should present strong cipher suites.
- Limit the lifetime of issued certificates to 825 days in accordance with the CA
- Assure the trustworthiness of the certificate authority and consider a contingency plan for if the CA is no longer trusted
- Ensure the configuration of TLS is performed in a secure fashion and avoid misconfiguration issues
- Certificate pinning is recommended to prevent replacement of certificates
- Client devices must ensure that they correctly validate server certificate

User Authentication

- PINs and passwords should not be easily guessable and weak credentials should be disallowed (Mobile Apps Password Policy).
- Multi-factor authentication before performing financial or other sensitive functions is strongly encouraged.
- Smartphone authenticator apps should be used for sending one-time passwords rather than SMS
- Assure the trustworthiness of the certificate authority and consider a contingency plan for if the CA is no longer trusted
- Biometric information is used for authentication, it must be stored with appropriate security measures

Secure Data Handling

- Mobile devices should securely store confidential information
- Trusted hardware should be used for the storage of sensitive information - If available on client device.
- Avoid storing information in external storage
- Delete confidential data from caches and memory after it is used
- Restrict data shared with other applications through fine-grained permissions
- Do not hard-code sensitive information such as passwords or keys
- Validate any input coming from the client that is to be stored in databases

Secure Application Development

- Develop according industry-accepted secure coding practices and standards
- Assure a means of securely updating applications.
- Have code independently assessed and tested by internal or external code review teams.

Android & iOS DFS Application Security Tests

Introduction

The Open Web Application Security Project

A collaborative, non-for-profit foundation that works to improve the security of web applications

Also works on security of mobile applications.

OWASP Mobile Top Ten

OWASP project that aims to identify and document the top ten vulnerabilities of mobile applications

Lab methodology

28 and 32 tests on Android and iOS respectively organized according to OWASP mobile top 10.

DFS Application Security Testing

28 and 32 tests on Android and iOS respectively were selected as necessary to ensure a high level of application security to meet the minimum-security best practices (**ITU-T Recommendation X.1150**)

Specifically, the following risk categories are considered:

- M2 Inadequate Supply Chain Security
- M3 Insecure Authentication/Authorization
- M4 Insufficient Input/Output Validation
- M5 Insecure Communication
- M7 Insufficient Binary Protections
- M8 Security Misconfiguration
- M9 Insecure Data Storage
- M10 Insufficient Cryptography

Summary od the tests

- 28 iOS and 32 Android tests organized according to OWASP mobile top ten risks 2024
- Tests with jailbroken/rooted and non jailbroken/non rooted phones
- Static analysis of apps on a workstation
- Dynamic analysis with a man-in-the-middle proxy

Device and application integrity.

- Making Sure that the App Is Properly Signed
- Testing Anti-Debugging Detection
- Testing Emulator Detection
- Testing File Integrity Checks
- Testing Jailbreak Detection
- Testing Obfuscation
- Testing Reverse Engineering Tools Detection
- Testing Root Detection
- Testing Runtime Integrity Checks

Secure data handling.

- Checking Logs for Sensitive Data
- Finding Sensitive Data in the Keyboard Cache
- Finding Sensitive Information in Auto-Generated Screenshots
- Testing App Permissions
- Testing Backups for Sensitive Data
- Testing Custom URL Schemes
- Testing for Injection Flaws
- Testing Implicit Intents
- Testing Key Management
- Testing Local Storage for Sensitive Data
- Testing Universal Links

Communication security and certificate handling.

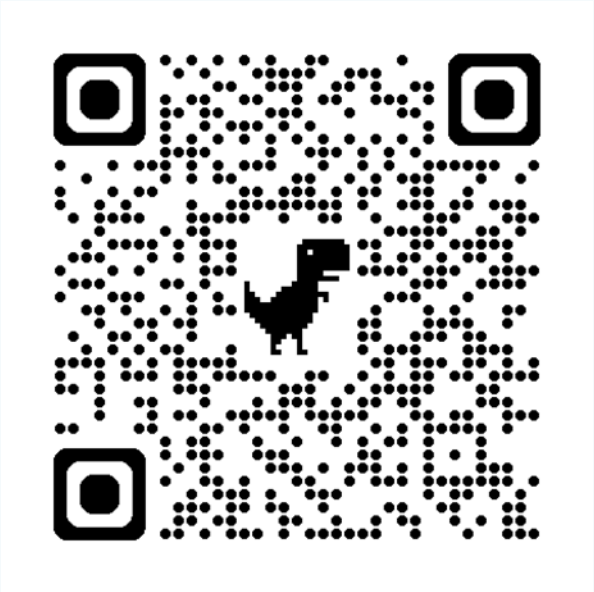
- Testing Custom Certificate Stores and Certificate Pinning
- Testing Data Encryption on the Network
- Testing Endpoint Identify Verification
- Testing the TLS Settings

User authentication.

- Testing Biometric Authentication
- Testing Confirm Credentials
- Testing Local Authentication

Secure application development.

- Checking for Weaknesses in Third Party Libraries
- Make Sure That Free Security Features Are Activated
- Testing Auto-Generated Screenshots for Sensitive Information
- Testing Enforced Updating
- Testing for Sensitive Functionality Exposure Through IPC
- Testing Random Number Generation
- Testing Symmetric Cryptography
- Testing the Configuration of Cryptographic Standard Algorithms
- Testing whether the App is Debuggable
- Verifying the Configuration of Cryptographic Standard Algorithms



http://www.itu.int/go/dfssl

Contact: dfssecuritylab@itu.int

Thank you!