FIGI FINANCIAL INCLUSION GLOBAL INITIATIVE

FIGI Symposium
18 May - 24 June, 2021

Security audit of various DFS applications

Methodology

Delphine Peter, Objectif Securite

figi.itu.int
#financialinclusion

Organized by

ITU · THE WORLD BANK IBRD • IDA · Committee on Payments and Market Infrastructures · BIS

Executive summary

- Developed method for testing DFS apps on Android
  - 18 tests organized according to OWASP mobile top ten
- Tested 3 applications
  - Payment applications
- Many best practices are not applied
  - No critical issue detected

1 Introduction

- DFS: Applications used for payment and money transfer without the need of having a bank account

- OWASP: The Open Web Application Security Project
  www.owasp.org
  - 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

2 The tests

- Our tests are organized according to the subjects of the OWASP Mobile Top Ten:
  - M1 Improper Platform Usage
  - M2 Insecure Data Storage
  - M3 Insecure Communication
  - M4 Insecure Authentication
  - M5 Insufficient Cryptography
  - M6 *Insecure Authorization*
  - M7 *Client Code Quality*
  - M8 Code Tampering
  - M9 Reverse Engineering
  - M10 *Extraneous Functionality*
- M6, M7, M10 out of scope because they would need access to the source code or require collaboration with the editor

M1 Improper Platform Usage

The application should make correct use of the features of the platform (phone's operating system)

- T1.1 Android:allowBackup
  - Backup of the application and its data into the cloud should be disabled
- T1.2 Android:debuggable
  - Debugging features of the application should be disabled
- T1.3 Android:installLocation
  - The application should be installed in the internal, more secure, memory
- T1.4 Dangerous permissions
  - The application should not require dangerous permissions, as defined by Android, e.g. allow to make phone calls

M2 Insecure Data Storage

Data should be stored in a way that limits the risks in case of loss or compromise of the phone

- T2.1 Android.permission.WRITE_EXTERNAL_STORAGE
  - No permission to write to a removable memory card
- T2.2 Disabling screenshots
  - If not disabled, screen shots are done automatically to generate thumbnails for task switching

M3 Insecure Communication

Protect against eavesdropping and manipulation of traffic

- T3.1 Application should only use HTTPS connections
  - Test by sniffing traffic
- T3.2 Application should detect Machine-in-the-Middle attacks with untrusted Certificates
  - Would allow anybody to intercept traffic
  - Test by intercepting traffic with proxy
- T3.3 Application should detect Machine-in-the-Middle attacks with trusted certificate
  - Would allow authorities to intercept traffic
  - Test by installing root certificate on phone, intercept with proxy
- T3.4 App manifest should not allow clear text traffic

M4 Insecure Authentication

Prevent unauthorized access to the application

- T4.1 Authentication required before accessing sensitive information
  - Application must require PIN or fingerprint
- T4.2 The application should have an inactivity timeout
- T4.3 If a new fingerprint is added, authentication with fingerprints should be temporarily disabled
  - User should provide PIN to enable fingerprints again
  - Prevents attacks where an attacker adds their fingerprint to access the application
- T4.4 It should not be possible to replay intercepted requests (e.g. a money transfer)
  - An attacker intercepting a request for a money transfer could replay it to steal money from the victim.

M5: Insufficient Cryptography

Cryptography can only protect confidentiality and integrity of data if correctly implemented

- T5.1 The app should not use unsafe crypto primitives
  - E.g. MD5, SHA-1, RC4, DES, 3DES, Blowfish, ECB
  - Search for these in the code
  - Detection of these primitives does not imply that they are used for protecting critical information!
- T5.2 The HTTPS connections should be configured according to best practices
  - Watch where the app connects to, use Qualys SSL labs to evaluate configuration, expect a grade of B or more

M8: Code Tampering

Prevent an attacker from tampering the code on the telephone

- T8.1 The application should refuse to run on a rooted device
  - On a rooted device, users can manipulate the code of the application

M9 Reverse engineering

Prevent attackers from analyzing the logic of the application

- T9.1 The code should be obfuscated
  - When the code is obfuscated, it is much more difficult to understand the logic of the code
  - This makes it more difficult to manipulate the code or to find potential vulnerabilities
  - Decompile the code and assess its readability