White Paper on Android Security and Network Management- Best Practices

OptiSol™

Solutions | Services

www.optisolbusiness.com

TABLE OF CONTENTS

PREFACE

OptiSol is excited to release this white paper on Android Security and Network Management best practices that focuses on providing measures to protect mobile apps from breach. The report highlights the security solutions that can be incorporated in the mobile application in the entire development cycle to make it safer from the security threats.

Since 2006, OptiSol acts as the reliable outsourcing services partner in providing optimized business solutions with delivery excellence for Startups and Small and Medium Enterprises. We provide comprehensive IT solutions that includes Web and Mobile Application, Professional services and Commercial Package implementations. We have flexible engagement options on proven Onsite – Offshore framework that suits need from all gamut of customers – Startup, Independent Software Vendor (ISV), small or mid-size companies and Enterprises.

We have been building Enterprise Mobility Solutions on native iOS and Android platforms along with Web service development. We cater to three segments of enterprises – with ERP platforms like Oracle EBS, with custom ERP solutions and with web applications for process automation. You can visit us at www.optisolbusiness.com to know more about our offerings.

EXECUTIVE SUMMARY

In today's online environment, the possibility of network vulnerability and data leakage is common and dangerous. The growing requirement of mobile applications in business has raised a huge question about the level of security it provides in the safeguarding corporate data and protection from mobile threats. Thus, one of the key requirements in mobile app development is applying a vigorous security model that protects data privacy and allows secure usage. The android applications are developed with inbuilt security features, which are in leveraged by developers during the building and testing stages of application development. This paper gives the best practices that need to be followed in the entire development cycle to provide industry enabled security in applications.

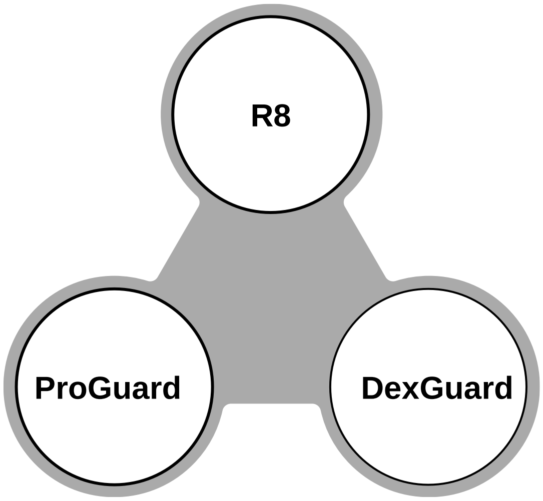
HIGH-LEVEL USER AUTHENTICATION

The sensitive information remains secure through a robust session management and a disconnected system. You should prioritize the importance of setting up the advanced authentication mechanism, using tools like JSON web tokens or OAuth 2.0.

- OAuth 2.0
- JSON web tokens
- Authenticate users and keys with biometrics (Face Recognition, Fingerprint)

CODE OBFUSCATION

Protect the source code by making it unintelligible for both humans and decompiler. All this, while preserving its entire operations during the compilation. The purpose of the obfuscation process is to give an impenetrable code. It promotes the confidentiality of all intellectual properties against reverse engineering.



We recommend R8 or ProGuard as a baseline for any application to make reverse engineering a little harder. For more advanced app protection, you can use commercial tool DexGuard (commercial tool). It optimizes and protects the entire Android app, including bytecode, native code, resources, and assets.

Reference: https://www.guardsquare.com/en/products/dexguard

DATA ENCRYPTION AND SECURE THE SERVER

Security in mobile apps involves keeping all kinds of information stored in the device secure. It encompasses the data that is transited between the back-end server and the application as well as the source code.

- Using HTTPS and SSL provide secure protocols for transferring data between your app and servers.
- The communications between an app and the servers must be over an HTTPS connection.
- By using HTTPS and the server that is configured with a certificate issued by a trusted certificate authority, assures that your network traffic is secure against attacks.
- Encrypting input data before passing to server and server will decrypt to get original data.

NETWORK SECURITY MANAGEMENT

Use SSL traffic

- If your app communicates with a web server that has a certificate issued by a well-known, trusted CA, the HTTPS request is very simple.

Add a network security configuration

- If your app uses new or custom CAs, you can declare your network's security settings in a configuration file. This process allows you to create the configuration without modifying any app code. We can do in two ways one is manifest and another one is xml file configuration.
  - Certificate pinning: Restrict secure connections to particular certificates
  - Cleartext traffic opt-out: Disable cleartext traffic
  - Enable the Network Security Configuration by manifest or xml file

Create our own trust manager

- Your SSL checker shouldn't accept every certificate. You may need to set up a trust manager and handle all SSL warnings that occur if one of the following conditions applies to your use case:
  - You're communicating with a web server that has a certificate signed by a new or custom CA.
  - That CA isn't trusted by the device you're using.
  - You cannot use a network security configuration.

CLIENT-SIDE SECURE DATA STORAGE

If you are saving data on the external storage and don't want people to access it, encrypt it.

Passwords, session tokens, recognition hash values should be stored as encrypted data and get values by decryption.

We can use SQLCipher for handling encryption and decryption in database handling. We will store encrypted data in database and get values by decryption with same algorithm we used to encrypt.

AES and RSA are still extremely safe for usage in the Android applications. With this encryption level you can store your public keys even in SharedPreferences.

For private secret keys Android provides Android Keystore System that available since API 18. It is a special secure storage and the key once stored in it can't be somehow exported back but it still can be used for the encryption/decryption process.

- File Encryption
- Encrypted key value storage
- Data Encryption (Advanced Encryption Standard)

Reference: https://developer.android.com/guide/topics/security/cryptography

Reference: https://www.zetetic.net/sqlcipher/

Reference: https://github.com/adorsys/secure-storage-android

ANALYTICS

Analytics data can often already be enough to identify users or to be able to access their data. E.g. an analytics framework recording your screen for crash reports can read your users’ login credentials.

- Google Analytics
- App center Analytics
- We can use different tools available in market.

OTHER CONSIDERATIONS

- Use implicit intents and non-exported content providers
- Share data securely across apps
- The files inside the directory are secure as they use the MODE_PRIVATE file creation mode by default.
- Do not pass sensitive information through Broadcast
- Never ever store Password and private keys in shared preference
- Restrict google API key access in cloud console by add package and SHA key details.
- Protect your Service and content provider with Permission by exported as false in manifest.
- Disable log in production mode

USE IMPLICIT INTENTS AND NON-EXPORTED CONTENT PROVIDERS

Show an app chooser

- When you safeguard the data that you exchange between your app and other apps, or between your app and a website, you improve your app's stability and protect the data that you send and receive.

Apply signature-based permissions in manifest

- When sharing data between two apps that you control or own, use signature-based permissions.
- These permissions don't require user confirmation and instead check that the apps accessing the data are signed using the same signing key.
- Therefore, these permissions offer a more streamlined, secure user experience.

Disallow access to your app's content providers

- Unless you intend to send data from your app to a different app that you don't own, you should explicitly disallow other developers' apps from accessing the ContentProvider objects that your app contains.

Ask for credentials before showing sensitive information

- When requesting credentials from users so that they can access sensitive information or premium content in your app, ask for either a PIN/password/pattern or a biometric credential,

SHARE DATA SECURELY ACROSS APPS

Follow these steps to share your app's content with other apps in a more secure manner.

- Enforce read-only or write-only permissions as needed.
- Provide clients one-time access to data by using the FLAG_GRANT_READ_URI_PERMISSION and FLAG_GRANT_WRITE_URI_PERMISSION flags.
- When sharing data, use "content://" URIs, not "file://" URIs. Instances of FileProvider do this for you.

CONCLUSION

Mobile solutions help the organization/enterprises to reach their customers and prospects effectively and are a vital part of the digital transformation strategy. Even though eliminating security threat is not possible, protecting our devices from external threats is a wise decision. This whitepaper has given an overview about the android security and network management practices followed by developers during the application development.

Hope you enjoyed this white paper and we look forward to your suggestions and inputs.

OptiSol™
Solutions | Services

OptiSol – a reliable outsourcing services partner providing optimized business solutions with delivery excellence for Startups and Small & Medium Enterprises. We have been serving its clients in North America, United Kingdom, Europe, Australia and Asia Pacific markets since its inception.

NASSCOM® | amazon web services Partner Network | Microsoft CERTIFIED Partner

TOP CERT peopleperhour TOP CERT SELLER 300TH POSITION OUT OF 20K+ SELLERS | CIO Review 20 MOST PROMISING COMPANIES - 2016 INDIA

OptiSol Business Solutions Private Limited

Baid Hi Tech Park, 5th Floor, Block # 38, East Coast Road,
Thiruvanmiyur, Chennai – 41

+91 44 42108070 (India) | +1415-233-4737, +1908-838-0191 (USA)

www.optisolbusiness.com | info@optisolbusiness.com