ACL Digital
an ALTEN group company

Innovate, Digitize, Transform

MOBILE APPLICATION SECURITY

By Ramkumar Murugadoss and Arif Nasim

Objective

The objective of this paper is to help the developer community across organizations apply necessary security mechanisms while building their mobile applications.

The exponential growth in the number of mobile devices and solutions has led to vulnerability in mobile applications being more common especially for applications that deal with personal and sensitive data transactions. Attackers target these applications to steal sensitive user data. Although many mobile applications do provide high security mechanisms, there is still room to provide improved security.

Key factors driving mobile application vulnerability include the following.

- Sensitive information storage in mobile
- The hard coded values
- Application binary reversing/decompiling
- Sensitive data transport between mobile application and server

Key Challenges and Threats

Mobile applications face multitude of challenges and threats, extending across areas including operating system, database and networks. Few prominent ones are given below.

- The attacker who can crack the application may interrupt the server as well
- Applications may leak data, when it is connected to vulnerable networks
- Using constant keys and values inside the application makes it vulnerable to the de-compilation tools
- Data stored inside the application using preferences/SQLite can be easily dumped
- The key threats and trends have been cited as OS manageability and updates, security policies and malware attacks targeting the customer applications that use value transactions.

The Risk analysis

According to OWASP (Open Web Application Security Project) the following are the top 10 risks on mobile application security

- Insecure Data Storage
- Weak Server Side Controls
- Insufficient Transport Layer Protection
- Client Side Injection
- Poor Authorization and Authentication
- Improper Session Handling
- Security Decisions Via Untrusted Inputs
- Side Channel Data Leakage
- Broken Cryptography
- Sensitive Information Disclosure

Listed above are the risks that mainly deal with user's data. In addition to these, the applications also face risks like de-compilation of the app's binaries using tools that leads to sensitive data leakage and might give a chance to attack the server if the API details are exposed.

The image below covers the most common tools that can be used to de-compile/reverse engineer the Android & iOS binaries.

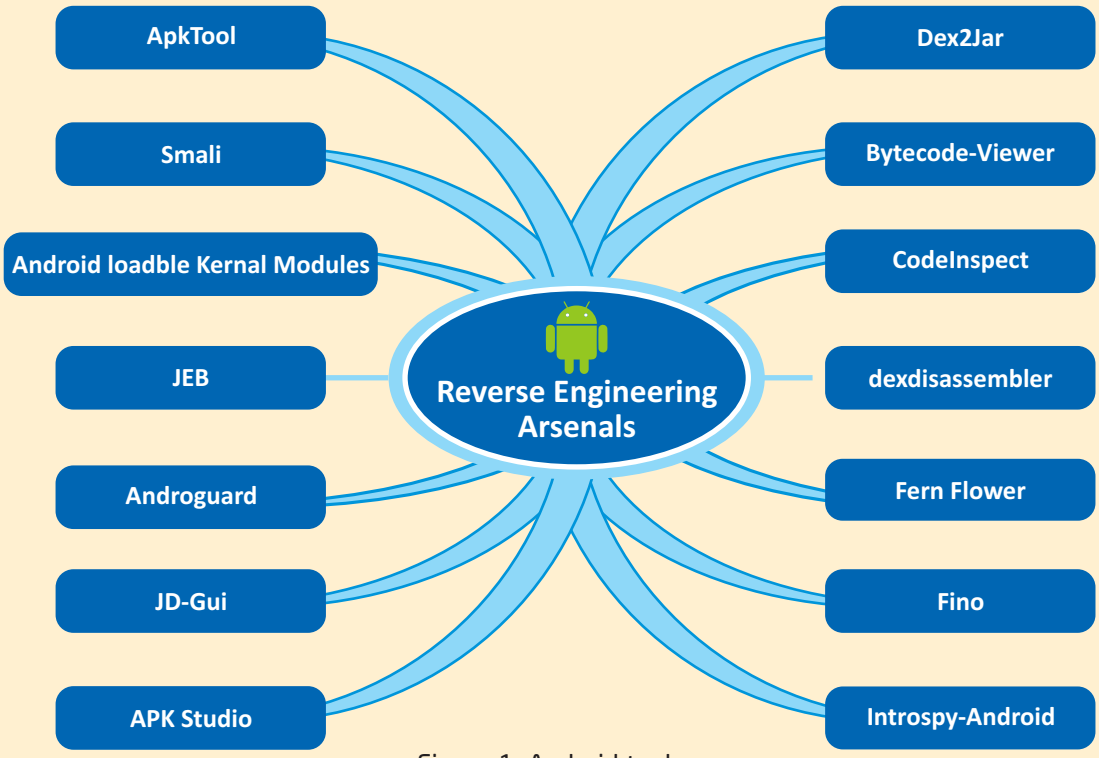


Figure 1: Android tools

Figure 2: iOS tools

Common Security Vulnerabilities

1. Storing sensitive data on Mobile

Most of the mobile applications have the need to store the data (it may be user specific or some sensitive data which the server requires to sync with the mobile app) on the mobile app. The possible ways of storing data in the mobile app are SQLite (Database), App's Preferences or File storage.

Storing the sensitive data using anyone of these options makes it hackable. For example, if the information is stored using SQLite, the SQLite files can be easily copied and it can be viewed through any SQLite browser available. The data stored in preferences can be easily dumped using few commands and as we all know very well, the files can easily be copied from a device.

2. Data on Transport layer

Any information from server to the mobile app will have to go through the network. There are packet sniffer tools available which the attackers can use to view the data packets and trace the requests and the responses, thereby exposing user's sensitive information. Apart from this the attacker can also override the responses to the mobile apps by manipulating the firewall policies.

3. App de-compilation reverse engineering

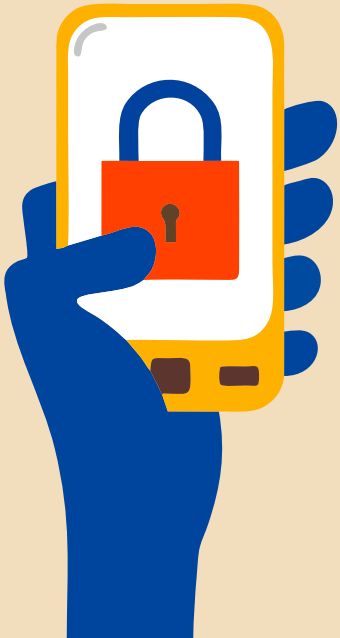
There are tools available to de-compile the app's binary, shown in figures 1 & 2. In this case the app's business logic can be traced and it will enable easy access for the attackers to play with the mobile app. By de-compilation, the app's code/data becomes visible and vulnerable for manipulation.

Preventing the Security Vulnerabilities

With the above stated vulnerabilities, following are the approaches that can be taken to secure the mobile app and the data.

Encryption & Decryption of data

Storing any sensitive data on the mobile is vulnerable. On the other hand, storing the encrypted data on the mobile reduces the chance of vulnerability. There various encryption & decryption algorithms like AES, DES, RSA, etc. that help in securing the code and storing the encrypted data in the preference or SQLite.



SQLChiper

SQLChiper is an encrypted database, which is an open source library and provides AES-256 bit encryption for the SQLite database files. In general the SQLite files are exposable and can be viewed through SQLite browsers, so storing sensitive data on SQLite can be dangerous. The SQLChiper provides a mechanism where the database files can be encrypted hiding the complete table structure of the SQLite.

Using Native Objects (.so files)

Another method to hide the sensitive information and the confidential business logic can be done by linking the native objects (.so) files with the application. The de-compilation of shared objects is complicated and to get the information from those files may be possible but is definitely not so easy. The native objects linked with the mobile application can be called only by that application and the complete application logic can be hidden inside the native objects.

Obfuscation

Obfuscation is a method to make the source code difficult for understanding. If the app binary is de-compiled, the complete source code will be visible to the attackers. In order to confuse attackers, the source code can be obfuscated. In general, obfuscation will complicate the source to understand by restructuring the classes and methods. Apart from this few obfuscators also provide features like byte code which makes it an extremely hard task to de-compile and use.

Focus areas for mobile application security testing

From a security perspective, following are 4 major areas to concentrate while doing mobile app testing.

File System	Application layer
Testing the data that the app is writing to the device's file system and also how this data is being stored.	Testing the app's communication with the web-services, if it is HTTP or SSL, and verifying the communication layer security.
Transport layer	**Application reversing**
Testing how the app is communicating over the network and the data security of the app over network.	Testing how the app is exposing the source code when decompiling the app's binary.

Conclusion

While mobile applications certainly bring enormous opportunities for businesses on the go, the potential risks are increasing day by day. Need of the hour is to implement adequate security measures for the mobile applications to secure their users and data. The insights outlined in this whitepaper shall provide much needed insight into the mobile security, threat analysis and a checklist for evaluation of various security threats before they can cause harm.

References

- http://www.gartner.com/newsroom/id/3127418
- http://www.decompilingandroid.com/mobile-app-security/top-10-mobile-security-risks/
- https://www.owasp.org/images/9/95/ASDC12-Smart_Bombs_Mobile_Vulnerability_and_Exploitation.pdf
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project
- https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=M-Tools
- http://sqlcipher.net

To know more about how ACL can partner with you to help create Digital Transformation, connect with: **business@acldigital.com**

www.acldigital.com

USA | UK | France | India