HCTL Open International Journal of Technology Innovations and Research (IJTIR)
http://ijtir.hctl.org
Volume 14, April 2015
e-ISSN: 2321-1814, ISBN (Print): 978-1-62951-946-3

Code and Data Security Risk in Android

Brij K. Misra[1], Smita Tripathi[2]

brijmisra02@rediffmail.com

Abstract

For a developer Security of Code and Data is very important and this paper presents risk involved in Android platform's mobile and non-mobile applications. On Android platform for storing data commonly used techniques are by popular open source SQLite database, Preferences, External storage, for graphics common techniques are using Drawables like Xml resources and Images, and for programming java is used. C and C++ by Android Native Development Kit (NDK), Python by Scripting Language for Android (SL4A) and other languages are also used to program Android applications. Though All these commonly used techniques but code and data are easily visible to anyone by using right tools or reverse engineering.

Most of the processes discussed are executable only on rooted Androids. Rooting Android is same as getting Super-user SU permission on Linux and because of Android itself is based on Linux and being totally open source getting super-user permission is not a very difficult task.

Keywords

Android, SQLite database, Preferences, External-storage, Assets, XML

Introduction

Android is an open source software assemble of an operating system, middleware and key applications for mobile devices introduced by Google capable of running multiple application programs. It is a complete operating environment based upon the Linux® V2.6 kernel. Initially, the deployment target for Android was the mobile-phone arena such as smart phones and low-cost flip-phone devices [1, 2].

Importance of Security of valuable code & data on Android platform to prevent hard work of developers and organizations from being copied or modified by tools or reverse engineering is Because of open source nature of android platform it is growing at a tremendous rate, and so the no. of developers & quality of apps also. Being based on linux its exceptional in supporting a vast range of devices, from single core 500Mhz devices to octa-core and >2Ghz processors.. processors from different brands and architectures from mediatech , arm , intel and many others. Roughly 250Mb ram to >2Gb of ram and 250 Mb storage to 64 Gb storage, from hone automation systems to wearable

gadgets, medical devices. All these are just basic features of Android platform. Development is happening very quickly by large Android community, Android Operating System is maturing and also supporting advance features like dual boot android devices, low cost computers by governments , health monitoring systems, and via android debug bridge (ADB) Arduino microcontrollers are extending its capabilities way beyond it ever been imagined. Developing for such a powerful system it is must to secure data & code from being copied or reverse engineered.

Why to Secure Code and Data

Two phenomenons in trend on different app stores.

1) A developer makes a game & it goes viral. Income of developer spikes up suddenly but very soon many different copies of app clutters the app store and ruin the business, earnings fall down.
2) A developer creates a data based app but data is cracked out and pirated causing extreme harm.

The best solution to this problem is storing data in cloud as Database, XML, etc. and allowing only in app access over http or https. Storing data in cloud is much more safe than storing data locally on devices. For Android Platform Open SSL Encryption and Cryptography can be used wisely, and for securing code the techniques used are Obfuscation and Security through obscurity.

Obfuscation or beclouding is the hiding of intended meaning in communication, making communication confusing, willfully ambiguous, and harder to interpret. [5]

An attacker's first step is usually information gathering, so another step can be delaying process of information gathering by security through obscurity technique [6].

Skills and Tools of Attackers

Skills required for these attacks are very basic, any script kiddie with knowledge of Android Project Structure can perform all of them just by studying step by step tutorials. Tools required are Rooted Android device, Apk of the app to be attacked but most of the time they are not even required to be installed. Android root file explorer, Sqlite browser/editor, text editor, zip extractor, Apktool software, and Lucky patcher Android app.

Apktool is a tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications; it makes possible to debug smali code step by step. Also it makes working with app easier because of project-like files structure and automation of some repetitive tasks like building apk, etc. It is NOT intended for piracy and other non-legal uses. [7]

Data Attack Methods

Drawables - Drawable images from any app can be extracted by renaming apk files to file_name.zip and extracting it gives all images in ./file_name/res/drawable and ldpi , mdpi , hdpi, xhdpi, and xxhdpi images will be in respective drawable folder. Can be done also with Apktool.

Assets - Assets from any app can be extracted by renaming apk files to file_name.zip and extracting it gives all images, music , video, html or any data that is not intended to be compiled and stored in assets, in ./file_name/res/assets folder.

Strings - String values from any app can be extracted by renaming apk files to file_name.zip and extracting it gives all strings in ./file_name/resources.arsc file by opening it in any text editor.

Data saved in Shared Preferences- Key value pairs saved in app preferences can be obtained by using any root explorer in android device and going in root in/data/data/__app_package_name__/shared_prefs/__preferences_filr_name__.xml and can be viewed in any text editor. In rooted devices having Read & write permission, Apps can be tricked to have fake data like thousands of coins to buy objects in game by modifying value of key value pair storing no. of coins bought in the game. For modifying shared preferences in rooted device just opening the xml file in a text editor, modifying value of respective key value pair is needed.Can be done also with Cheatdroid app for rooted Androids.

Data stored in Sqlite browser - Sqlite database can be obtained by using any root explorer in android device and going in root in /data/data/__app_package_name__/databases/__databse_file__.db . All Tables and values can be viewed by using any Sqlite browser.Can be done also with dSQLiteManager app for rooted Android.

In rooted Android Devices with read write permission Apps can be tricked to have fake data in databases also. All Tables and values can be modified by using any Sqlite editor and saving bact to its location.Can be done also with aShell app for rooted Android.

Data stored on SDcard-Expansion files or Data Stored on Sdcard like zip, mp4, mp3, pdf, can be accessed in Sdcard/Android/data/__app_package_name__/.. . For accessing data stored on SDcard in obb format go to Sdcard/Android/obb/__app_package_name__/__file_name_.obband change the extension from obb to zip and extract it using any zip extractor.

Code Reverse Engineering

For reverse engineering android applications Apktool is used to Decompile app's apk then code is edited and Package is Recompiled back to app's apk and ready to install. By following these steps any attacker can change the Layouts , graphics, Views, & Strings in Resources and can also modify java code in app.

Decompiling an apk changes binary form of XML and Java into readable code. Recompiling an app changes readable and modified Xml and Java files back to installable apk.

References

[1] http://en.wikipedia.org/wiki/Android_%28operating_system%29
[2] http://en.wikipedia.org/wiki/Android_(operating_system).
[3] http://developer.android.com
[4] http://code.google.com/p/android-scripting/
[5] http://en.m.wikipedia.org/wiki/Obfuscation
[6] http://en.m.wikipedia.org/wiki/Security_through_obscurity
[7] http://code.google.com/p/android-apktool/

Authors

BRIJ K. MISRA

Over 14 year rich & varied experience as a Computer Consultant and Faculty in the best colleges of Lucknow. Presently associate with various companies and universities as Freelancer Trainer to deliver online education. Professionally & academically qualified with good knowledge of all major Operating systems, DBMS/RDBMS, Languages,& Methodologies. Having qualification as M.com,M.C.A.,MTech-IT and MPHIL-C.S.

SMITA TRIPATHI

Over 9 year rich & varied experience as a Computer Faculty in the Lucknow. Presently working in Surya College of Business management as Assistant Professor . Having qualification as B.Sc, M.C.A., M.Tech-IT and MPHIL-CS, Pursuing Ph.D.