Volume 7, Issue 4, April 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Review on Android App Security

Sajid Nabi Khan, Ikhlaq Ul Firdous
MIET, Department of CSE, Kot Bhalwal, Jammu,
J&K, India

Abstract: ***Android is the most popular operating system and has the biggest market share among all the mobile OS's. Its ubiquitous influence makes it an easy target for the malware developers and other computer criminals. Android is the most secure operating system than its counterparts. It has very few restrictions for the developer and has no security scan for the apps being uploaded in the Play Store, which thus increases the risk for end users. In this paper we have reviewed android security model, security provided by the android OS, and the security provided by the android apps.***

Keywords: Android OS security, Android Platform Architecture, Security

I. INTRODUCTION

Android is currently the most popular mobile OS. Used from phones and watches to cars and TV's, it has customized our digital lives. There are more than 2 million android apps on the Play Store. The reason behind its popularity is that its source code is released by the Google under open source licenses. Android is popular with the mobile companies that require a ready-made, low cost and customizable OS for their devices. Android's platform fragmentation caused issues with security, in which the major of Android devices did not receive security patches, but have been improved in the recent developments.

The open nature of Android and its large user base has made it an appealing platform to attack. Premium service abuse is the most common type of Android malware, where text messages are sent from infected phones to premium-rate telephone numbers without the consent or even knowledge of the user. Other malware displays unwanted and intrusive advertisements on the device, or sends personal information to unauthorized third parties. It is obvious why Android is a target, but what makes it vulnerable? Google took certain measures to provide a better security to the user; the android apps run in a sandbox, the Google Bouncer malware scanner is used to watch over and scan apps on Play Store, before installing an app the Play Store displays a list of the requirements that the app needs to function, but are they really enough?

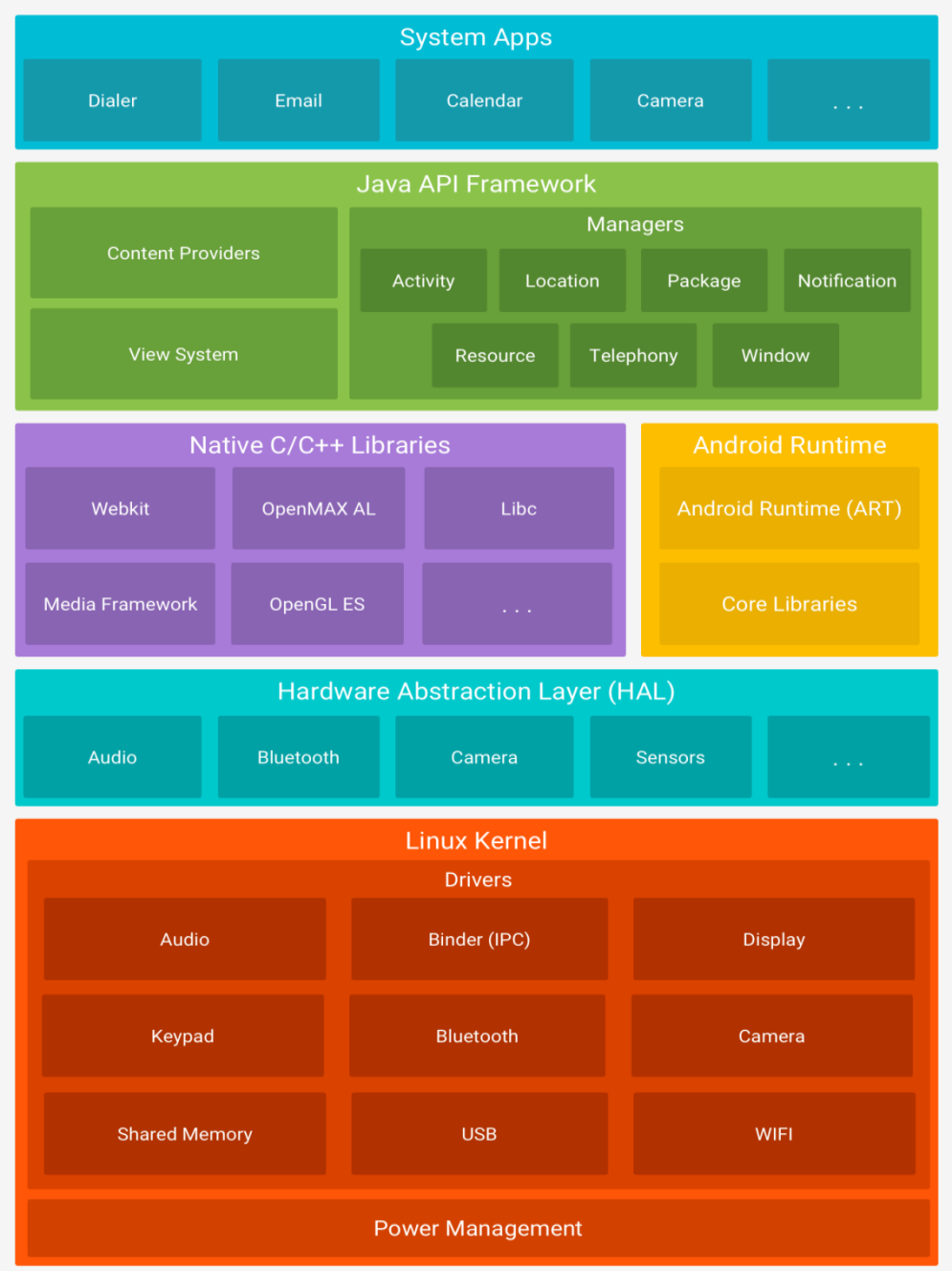
II. ANDROID PLATFORM ARCHITECTURE

Android is an open source, Linux-based software stack created for a wide array of devices and form factors. The diagram shows the major components of the Android platform and they are explained:

- ***The Linux Kernel:*** It is the foundation of the Android platform. Using a Linux kernel allows android to take advantage of key security features and allows the device manufactures to make hardware drivers for a well known kernel.
- ***Hardware Abstraction Layer (HAL):*** The HAL provides standard interfaces that expose device hardware capabilities to the higher level Java API framework. It consists of multiple library modules. When a framework API makes a call to access device hardware, the Android system loads the library module for that hardware component.
- ***Android Runtime:*** Android Runtime is an application runtime environment used by the Android operating system. Some of the major features of ART include the following:
  - Ahead-of-time (AOT) and just-in-time (JIT) compilation
  - Optimized garbage collection (GC)
  - Better debugging support, including a dedicated sampling profiler, detailed diagnostic exceptions and crash reporting, and the ability to set watchpoints to monitor specific fields.

- ***Native C/C++ Libraries:*** A lot of core Android system components and services are built from native code that requires native libraries written in C and C++.
- ***Java API Framework:*** The entire feature-set of the Android OS is available to you through APIs written in the Java language. These APIs form the building blocks you need to create Android apps by simplifying the reuse of core, modular system components and services, which include the following:
  - A rich and extensible View System you can use to build an app's UI, including lists, grids, text boxes, buttons, and even an embeddable web browser

- A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
- A Notification Manager that enables all apps to display custom alerts in the status bar
- An Activity Manager that manages the lifecycle of apps and provides a common navigation back stack
- Content Providers that enable apps to access data from other apps, such as the Contacts app, or to share their own data.



- ***System Apps:*** Android OS comes with a set of core apps for email, text messaging, calendars, internet browsing, contacts, and more. Apps included with the platform have no special status among the apps the user chooses to install. So a third-party app can become the user's default web browser, text messenger, or even the default keyboard. The system apps function both as apps for users and to provide key capabilities that developers can access from their own app.

III. ISSUES IN THE SECURITY OF ANDROID

Android has robust security measures, but even then it is not 100% secure. There are a lot of security issues faced by Android, few of them are:

i. QuadRooter Vulnerability: QuadRooter is a set of four vulnerabilities affecting Android devices built using Qualcomm chipsets. An attacker can exploit these vulnerabilities using a malicious app. Such an app would require no special permissions to take advantage of these vulnerabilities, alleviating any suspicion users may have when installing.

ii. The 'Certifi-gate' mRST flaw: This is a flaw in two mobile Remote Support Tool plug-ins used by many handset makers, including Samsung, LG, HTC, Huawei and ZTE running Android versions up to 5.1. Attackers could exploit it by sneaking a bogus app onto a phone which exploits the flaw in a way that elevates the attacker's permissions. From that point on, the attacker would have complete remote control over the smartphone.

iii. 'Stagefright' MMS flaw: It is arguably the most serious security flaw ever to hit Android, this one affecting a media playback component of the OS nobody usually thinks much about called Stagefright. The attackers could exploit the issue by sending a malicious video message to almost any Android handset on the plant, which would execute automatically. Incredibly, no user interaction is needed and the message could even render itself invisible by deleting itself.

iv. Android Installer hijacking: Affecting older smartphones only – that was still around half of all Android smartphones at the time of its discovery – this offered a novel way of attackers to replace one installer (or APK file) with another one when using third-party app stores, in effect letting a malicious app replace a legitimate one without the user realising it.

v. Android FakeID flaw: This flaw offers a way for a malicious app to hijack the trusted status of a legitimate app through (by forging its digital certificate), effectively escaping any sandboxing security on the device.

vi. TowelRoot: It was an unusual kernel-level flaw affecting something called the futex subsystem. However, not long after it was incorporated into a tool designed to root Android 4.4 called TowelRoot, which effectively functioned as a benign proof-of-concept exploit.

IV. LITERATURE SURVEY

W. Enck, D. Octeau, P. McDaniel and S. Chaudhri presented 'A study of Android application security' and introduced the ded decompiler, which generated the android application source code directly from the installation image. They designed and executed a horizontal study of smartphone applications based on static analysis of 21 million lines of recovered code and concluded that low or no restriction of entry for application developers increased the security risk for end users[5].

S. Powar, Dr. B. B. Meshram on their research 'Android security framework', described android security framework and concluded that the increased exposure of open source smartphone is increasing the security risk. The permission module to secure the phone is very basic. The user has only two options at the time of app installation: first allow all requested permissions and second deny the requested permissions which leads to stop installation[6].

S. Kaur and M. Kaur in their paper 'implementing security on Android application' described how the security in android based systems can be increased[7].

S. Smalley and R. Craig in their research 'Security Enhanced (SE) Android: Bringing Flexible MAC to Android', showed how the android software stack defines and enforces its own security model for apps through its application layer permissions model. They also described how MAC- mandatory access control can be brought to Android by enabling the effective use of Security Enhanced Linux (SELinux) for kernel-level MAC and by developing a set of middleware MAC extensions to the Android permissions model[8].

M.Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, in their study 'Semantically Rich Application-Centric Security in Android' proposed a secure application interaction (Saint), which is an improved infrastructure that governs install-time permission assignment and their run-time use as dictated by application provider policy[9].

T. Luo, H. Hao, W. Du, Y. Wang and H. Yin in the paper 'Attacks on WebView in Android System' discussed a number of attacks on WebView, either by malicious apps or against non-malicious apps. They identified two fundamental causes of the attacks: weakening of the TCB and Sandbox[10].

A.D. Schmidt and S. Albayrak presented a paper on 'Malicious Software for Smartphones' and presented a list of most common behavior patterns and investigated the possibilities to exploit the standard Symbain OS API and additional malware functionalities[11].

G. Dini, F. Martinelli, A. Saracino and D. Sgandurra presented a paper 'MADAM: a multi-level anomaly detector for android malware' in which they showed that MADAM is able to notice several real malware found in the world. The device is not affected by MADAM due to the low range of false positives generated after the training phase[12].

V. RESEARCH FINDING

The security in Android is enforced is enforced by two basic methods. Firstly, each application runs as Linux process with their own user IDs. Thus vulnerability in one application does not affect other applications. IPC mechanisms need to be secured in Android by second enforcement mechanism. Android implements a reference monitor that gives an access to application components based on permission[13].

A study of Android Application Security revealed that the Phone identifiers especially the IMEI, are used to track the users. The IMEI is tied to personally identifiable information (PII). Phone identifiers are used as device fingerprints and sent to advertisement servers[5].

Research has shown that there are two fundamental causes of the attacks on WebView: weakening of the TCB and Sandbox. It has already spread through the masses and has affected millions of devices. Currently, they are developing a method to secure the WebView[10].

It is found out that a lot of developers fail to take necessary security precautions. The sensitive information is frequently written to Androids centralized logs, and even broadcasted to unprotected IPC's[5].

VI. CONCLUSION

The smartphones are creating a boom in the computing world, and Android is topping that list. Android has a lot fewer restrictions for the developer than its counterparts, thus increasing the risk of security for end users. There are lot of factors which are responsible for the threat in the Android OS. In this paper we have reviewed the security in android apps, and studied the functionality and threats to our privacy. We also saw that the Android OS provides better security than other operating systems.

REFERENCES

[1] Android (operating system) https://en.wikipedia.org/wiki/Android_(operating_system)
[2] Android Open Source Project: Android Security Overview https://source.android.com/security/
[3] Android Open Source Project: Platform Architecture https://developer.android.com/guide/platform/index.html#system-apps
[4] Android's 6 biggest security flaws 2016 http://www.techworld.com/security/androids-6-biggest-security-flaws-2016-3622116
[5] W. Enck, D. Octeau, P. McDaniel and S. Chaudhri 'A study of Android Application Security', *The 20th USENIX conference on Security*, 21-21.
[6] S. Powar, Dr. B. B. Meshram 'Android security framework', *International Journal of Engineering Research and Applications.*
[7] S. Kaur and M. Kaur 'Implementing Security on Android Application', *Journal of Environmental Sciences, Computer Science and Engineering & Technology*.
[8] S. Smalley and R. Craig 'Security Enhanced (SE) Android: Bringing Flexible MAC to Android', www.internetsociety.org/sites/default/files/02_4.pdf
[9] M.Ongtang, S. McLaughlin, W. Enck, and P. McDaniel, 'Semantically Rich Application-Centric Security in Android', *Computer Security Applications Conference, 340-349*
[10] T. Luo, H. Hao, W. Du, Y. Wang and H. Yin 'Attacks on WebView in Android System', *27th Annual Computer Security Applications Conference, 343-352*
[11] A.D. Schmidt and S. Albayrak 'Malicious Software for Smartphones', https://www.dailabor.de/fileadmin/files/publications/smartphone_malware.pdf
[12] G. Dini, F. Martinelli, A. Saracino and D. Sgandurra 'MADAM: a multi-level anomaly detector for android malware' http://www.iet.unipi.it/g.dini/research/papers/2012-MMM-ANCS.pdf
[13] W. Enck, Ongtang M., P. McDaniel, Understanding Android Security, IEEE Security Privacy, 7

