ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

# A Review on Android Security Threats

**Abhishek Tiwari[1], Mansi Mishra[2], Pragya Kamal[3], Mayank Deep Khare[4]**

Student, Department of Information Technology, Buddha Institute of Technology[1,2]

Assistant Professor, Department of Information Technology, Buddha Institute of Technology[3]

Assistant Professor, Department of Computer Science, Noida Institute of Engineering & Technology[4]

**Abstract:** Android is presently the world's most famous and generally utilized working framework in cell phones. It has increased colossal piece of the overall industry because of its open design and the prominence of its application programming interface in the engineer group. While the force of Android comes as its openness and simple to learn and execute nature. It clearly uncovered the gathering of an interconnected framework to a specific level of security dangers to the end clients. The expanded fame of the Android gadgets and related money related advantages pulled in the malware engineers, bringing about a colossal ascent of the Android malware applications. These issues could run from the customer side infusion, despicable session taking care of, broken cryptography, and inadequate transport layer insurance to uncertain information stockpiling. This study broadly covers different Android OS particular dangers and vulnerabilities.

**Keywords:** Android, Security, Threats, Vulnerabilities, Malware, Smartphones, Risks, Application, Developer.

## I. INTRODUCTION

With the late fast improvement of portable advances, Android has picked up an enormous number of clients throughout the most recent couple of years as cell phone [1]. Android created in Linux part has turned into the world most famous portable working framework because of its components, for example, open source and simple application improvement. In any case, dangers additionally take off in the then [3]. Android OS permits engineer opportunity gets to and adjusts the source code of applications running on it. So Android is giving a free stage to the engineers with various offices to produce new applications at a fast rate. Different applications are running on this stage, giving redid administrations to both people and ventures. Cell phones additionally cause security vulnerabilities and dangers because of its gigantic stockpiling of individual data and business protection. Security of Android OS has been an extraordinary worry in the overall [5] [2]. Because of the substantial client base, shrewd gadgets are utilized to store delicate individual data more much of the time than portable PCs and desktops. As an outcome, Android malware can't just take private data, for example, the contact list, instant messages, and area from its client, yet can likewise pick up control of gadget, cause monetary loss of the clients by making hidden premium rate telephone calls, instant messages and taking cash from bank accounts [4]. There are extensive assortments of Android vulnerabilities, and they can happen in any layers of Android OS stack, for example, application layer or structure layer. Different security issues, for example, unapproved access from one application to the others[5] (data spillage), consent heightening, repackaging applications to infuse malignant code, conspiring, and Denial of Service (DOS) assaults. In the interim, the speedy advancement of the amount of usages on Android markets makes it hard for application feature focus, for instance, Google App Store, for instance, to completely check if an application is authentic or noxious. Therefore, portable clients are left to choose for themselves whether an application is sheltered to utilize. As indicated by the yearly Mobile Threat Report (2013) by Juniper Networks discharge, Mobile malware developed at a rate of 614 percent amid the previous year (Kerr 2013), of which 92% are focused on particularly at Android. This paper goes for supplementing the previous audits by extending the scope of different Android dangers. In this study, we will cover about Android OS security dangers. Whatever remains of this paper is composed as tails: We first portray the Android OS and application and their models in Section II and after that some essential Android security dangers and issues in Section III lastly, the conclusion in Section IV.

## II. ANDROID OS AND APPLICATIONS ARCHITECTURE

In this area, we portray the design of the Android OS and its applications. Android is being produced and kept up by Google and advanced by the Open Handset Alliance (OHA). Android OS is put on top of the Linux bit, and it incorporates the middleware, libraries and APIs written in c dialect, and application programming running on an application system which incorporates Java-perfect libraries. Android's source code is discharged by Google under open source licenses.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

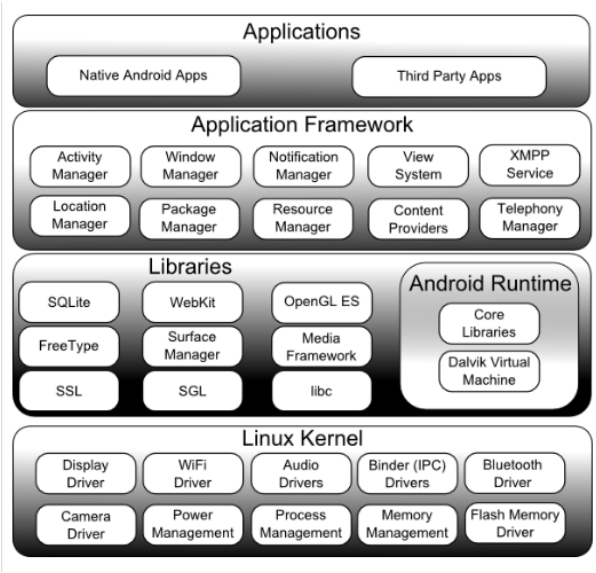
International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017




**1. Linux kernel**

The essential layer is the Linux Kernel. The entire Android OS is based on top of the Linux Kernel with some further building changes[6]. Kindly don't get confounded by the terms Linux and Linux Kernel. The term Kernel implies the center of any Operating System. By saying Android depends on Linux Kernel, it doesn't imply that it is Linux dispersion. It dislikes that. It basically implies Android at its center is Linux. Be that as it may, you can't run any Linux bundles on Android. It is an entirely unexpected OS. It is this Linux piece that cooperates with the equipment and it contains all the basic equipment drivers. Drivers are projects that control and speak with the equipment. For instance, consider the Bluetooth work. All gadgets have Bluetooth equipment in it. In this way the portion must incorporate a Bluetooth driver to speak with the Bluetooth equipment. The Linux portion likewise goes about as a reflection layer between the equipment and other programming layers. Android is based on a most prevalent and demonstrated establishment, the porting of Android to an assortment of equipment turned into a moderately effortless errand.

**2. Libraries**

The following layer is the Android's local libraries. It is this layer empowers the gadget to handle diverse sorts of information. These libraries are composed in c or c++ dialect and are particular for specific equipment.

**Some of the important native libraries include the following:**

**Surface Manager:**It is utilized for compositing window supervisor with off-screen buffering. Off-screen buffering implies the applications can't specifically draw on the screen; rather the drawings go to the off-screen cushion. There it is joined with different drawings and frame the last screen the client will see.

This off-screen cradle is the explanation for the straightforwardness of windows.

**Media framework:** Media framework provides different media codecs allowing the recording and playback of different media formats

**SQLite:** SQLite is the database engine used in Android for data storage purposes

**WebKit:** It is the browser engine used to display HTML content

**OpenGL:** Used to render 2D or 3D graphics content to the screen.

**3. Android Runtime**

Android Runtime consists of Dalvik Virtual Machine and Core Java libraries.

**Dalvik Virtual Machine:**

It is a sort of JVM utilized as a part of Android gadgets to run applications and is advanced for low handling force and low memory situations. Dissimilar to the JVM, the Dalvik Virtual Machine doesn't run .class documents, rather it runs .dex records. .dex documents are worked from the .class record at the season of arrangement and give higher proficiency in low asset situations. The Dalvik VM permits different occasions of a Virtual machine to be made at the same time giving security, detachment, memory administration and threading support.

**ART:**

Google has presented another virtual machine known as ART (Android Runtime) in their more up to date arrivals of Android. In Lollipop, the Dalvik Virtual Machine is totally supplanted by ART. Craftsmanship has many points of interest over Dalvik VM, for example, AOT (Ahead of Time) assemblage and enhanced rubbish gathering which help the execution of applications altogether.

**Core Java Libraries:**

These are different from Java SE and Java ME libraries. However, these libraries provide most of the functionalities defined in the Java SE libraries.

**4. Application Framework**

These are the obstructs that our applications specifically cooperate with. These projects deal with the fundamental elements of a telephone like asset administration, voice call administration, and so forth. As a designer, you simply consider these are some fundamental apparatuses with which we are building our applications.

**Important blocks of Application framework are:**

**Activity Manager:** Manages the activity life cycle of applications

**Content Providers:** Manage the data sharing between applications

**Telephony Manager:** Manages all voice calls. We use telephony manager if we want to access voice calls in our application.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

**Location Manager:** Location management, using GPS or cell tower

**Resource Manager:** Manage the various types of resources we use in our Application

**Notifications Manager:**It permits applications to show alarms and notifications to the client. With this administration, applications can advise the client of occasions that occur out of sight.

**5. Applications**

Applications are the top layer in the Android engineering, and this is the place our applications will fit into. A few standard applications come pre-introduced with each gadget, such as:

• SMS client app
• Dialer
• Web browser
• Contact manager

As a designer we can compose an application which replaces any current framework application. That is, you are not constrained in getting to a specific component. You are for all intents and purposes boundless and can whatever you need to do with the Android (the length of the clients of your application allows it). In this manner Android is opening unlimited chances to the engineer.

## III. ANDROID SECURITY ISSUES AND THREATS

The Android ecosystem has two main security risks, according to mobile security experts:

- The Google Play Store
- The fragmentation of devices and OS versions

**The Google Play Store's risks:**Android is a really open OS, and that makes it unsafe and prompts to potential security vulnerabilities when not oversaw intelligently. Google Play (in the past called the Android Market), the advanced circulation stage for applications for Android gadgets, is itself a wellspring of potential security dangers. "With Google Play, there are a higher rate of applications that contain malware or social designing to associate with malware, than some other application store by request of greatness, "It's not an all-around policed environment, and these elements keep on creating contact or resistance toward more prominent reception of Android in the undertaking." When clients download applications from Google Play, they frequently don't focus on the degree of authorizations an application can have on their gadget, "They typically simply acknowledge the consent amid establishment, and as a general rule, applications request more authorizations that they truly require." Android security is based upon a consent based component which manages the entrance of outsider Android applications to basic assets on an Android gadget. Such authorization based instrument is generally scrutinized for its coarse-grained control of utilization consents and the inefficient authorization administration, by designers, advertisers, and end-clients. For instance, clients can either acknowledge all authorization demands from an application to introduce it, or not to introduce the application. This kind of authorization administration is ended up being undesirable for the gadgets security. In this segment, we talk about the primary security issues of the Android, which prompts to client data spillage and puts the client's protection in risk [11].

**The fragmentation of devices and OS versions**

The Android stage additionally endures the issue of fracture; there are different forms of Android in the market, even on current gadgets. Producers frequently roll out their own improvements to Android so they could be behind Google's present reference discharge. Likewise, bearers and producers may not upgrade their gadgets Android rendition when Google does, or they take months or even years to do as such. Accordingly, many individuals inside a similar association may utilize obsolete adaptations that could be filled with security vulnerabilities. Investigate demonstrates that a larger part of Android gadget clients worldwide have gadgets with noncurrent forms of the OS, if clients have more seasoned variants of Android, that could mean vulnerabilities are left unpatched, and new elements of the OS won't contact them. The fracture issue increases the assault surface; in this way, there's no single security arrangement that will fit the majority of Android's varieties.

**There are some other threats stated**

**1. Information leakage:**

In current Android engineering plan, applications are confined from getting to assets or different applications unless it is approved by the clients. Clients need to give all asset get to demands before introducing and utilizing an application. Data spillage happens when clients give assets with no confinement from OS [24] [25]. This is finished by benefit heightening assault. With more than 1.4 million accessible applications in Google Play, a significant number of malevolent applications have been presented to Android clients for establishment. In any case, when introducing another application, just a little segment of clients focus on the asset being asked for, since they tend to race through provoked authorization ask for screens to get the chance to utilize the app. Just a little bit (3%) of clients are wary and makes revise answers to authorization allowing questions[7]. the purposes behind the ineffectualness of the present authorization control framework include:

(1) Unpracticed clients don't understand asset solicitations are immaterial and will trade off their protection,

(2) Clients have the desire to utilize the application and might be obliged to trade their security for utilizing the application [5].

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

**2. Email Links and Downloads**

In this danger, you have been communicated something specific that recommends heading off to a suspicious-sounding site, or there is a spontaneous connection, it is best to erase the email. The most effortless approach to avoid this sort of danger is to tell anybody that may conceivably email you to put a subject title that plainly recognizes the topic, so you know the substance is protected.

**3.RepackagingApps**

Repackaging is a standout amongst the most critical and regular security issues of the Android OS. Repackaging is the way toward dismantling/decompiling of .apk files utilizing figuring out procedures and including (infusing) vindictive code into the fundamental source code. Repackaging strategies that can be utilized on the Android stage permit noxious code to be camouflaged as an ordinary application. It is difficult to recognize a repackaged malignant code and a typical application in light of the fact that the repackaged application normally seems to work in an indistinguishable route from the honest to goodness one.

The repackaging steps are as per the following[8][9]:

**Unpacking:** unpacking APK files using available tools such as apktool, which is a tool based on reverse-engineering.

**Decompiling:** decompiling the Java source code using JAD and extracting the source code of Java classes.

**Codeinjection:**

injectingCodeandaddingresourcesintothemainsourcecodeusingJava developing environments.

**Repacking:** rebuilding the files using apktool and signing the generated files using jarsigner. Geimini and KungFu are examples of Trojans which are based on APK repackaging. These Trojans can be bundled into many valid Android apps.

**4.DenialofService(DOS)attack**

The expanding number of advanced cell clients and predominance of cell phones (telephones, tablets) which are associated with the Internet can be a stage for development of DOS assaults. Since the larger part of cell phones are not furnished with similar assurances (i.e. against infection programs) asPCs; noxious applications find it as an appropriate stage for DOS assaults. Abusing constrained CPU, memory, organisedata transfer capacity and battery power are the primary objectives of DOS assaults[10].

**5. Colluding**

The Colluding danger is a customer side assault. In this assault, clients introduce an arrangement of applications created by a similar engineer and same certificate and allow distinctive sorts of consents including delicate and non-touchy. Subsequent to introducing applications, these applications can exploit a mutual UID and access every one of their consents and assets[12].

**6. Bots, Trojans and malwares**

These are noxious programming which when downloaded by a client may hurt their framework by different ways. Bots are at times hard to perceive and may come in different structures. They additionally may bring about various issues to your Android security. Bots are some of the time ready to go about as a key lumberjack, who implies they can give your login data to digital crooks that will utilize it for spam or to take your personality, access your gadget. This risk is best put to rest by avoiding outsider applications that seem suspicious and to abstain from downloading untrusted programming.

## IV. CONCLUSION

Alongside the expanding pervasiveness of Android cell phones, the quantity of Android applications including malware is expanding day by day. In spite of conveyed Android security systems, malware exploits the Android security escape clauses to abuse the allowed assets. There by, numerous endeavors have been proposed to limit the effort of vulnerabilities in Android gadgets. In this study we researched about numerous conspicuous sorts of dangers concerning the security of Android and expressed about Android design architecture.

## REFERENCES

[1] Gartner, “Gartner: 1.1 billion android smartphones, Tablets expected to ship in 2014” Online; accessed at June 5, 2015, http://tinyurl.com/n8t3h9y.

[2] ParvezFaruki, AmmarBharmal, VijayLaxmi**:** "Android Security: A Survey of Issues, Malware Penetration, and Defenses" Authors, Computer Eng. Dept., Malaviya Nat. Inst. of Technol. (MNIT), Jaipur, India.

[3] Maya Krishnan: "Survey on Security Risks in Android OS and an Introduction to Samsung KNOX" Author, Department of Computer Science, CMRIT,India.

[4] “A clear-eyedA clear-eyed guide to Android's actual security Risks”>BY>BobViolino http://www.infoworld.com/article/2609338/android/a-clear-eyed-guide-to-android-s-actual-security-risks.html

[5] “5 SECURITY THREATS TO YOUR ANDROID PHONE” at koolspan.com

[6] “Android Operating System Architecture” http://www.eazytutz.com/android/android-architecture/

[7] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, “Android permissions: User attention, comprehension, and behavior,” in Proc. of the 8th Symposium on Usable Privacy and Security (SOUPS’12), Pittsburgh, PA, USA. ACM, July 2012, pp. 3:13:14.[Online].Available:http://doi.acm.org/10.1145/2335356.2335360

[8] H. Huang, S. Zhu, P. Liu, and D. Wu, “A framework for evaluating mobile app repackaging detection algorithms,” in Proc. of the 6th International Conference on Trust and Trustworthy Computing (TRUST’13), London, UK, LNCS, M. Huth, N. Asokan, S. ˇCapkun, I. Flechais, and L. ColesKemp, Eds., vol. 7904. Springer

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

ICACTRP 2017

International Conference on Advances in Computational Techniques and Research Practices

Noida Institute of Engineering & Technology, Greater Noida

Vol. 6, Special Issue 2, February 2017

Berlin Heidelberg, June 2013, pp. 169–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-38908-5 13

[9] W. Zhou, Y. Zhou, X. Jiang, and P. Ning, "Detecting repackaged smartphone applications in third-party android marketplaces," in Proc. of the 2nd ACM Conference on Data and Application Security and Privacy (CODASPY'12), San Antonio, Texas, USA. ACM,March2012,pp.317–326.[Online].Available: http://doi.acm.org/10.1145/2133601.2133640

[10] E. Kovacs, "Wi-fi direct flaw exposes android devices to dos attacks," Online; accessed at July 8, 2015, http://www.securityweek.com/wi-fi-direct-flaw-exposes-android-devices-dos-attacks.

[11] P. Faruki, A. Bharmal, V. Laxmi, V. Ganmoor, M. Gaur, M. Conti, and R. Muttukrishnan, "Android security: A survey of issues, malware penetration and defenses," 2015. [Online]. Available: http://dx.doi.org/10.1109/COMST.2014.2386139

[12] C. Marforio,A. Francillon, and S. Capkun, "ApplicationCollusion Attack on the Permission-Based Security Model and Its Implications for Modern Smartphone Systems". Department of Computer Science, ETH Zurich, 2010[Online]. Available: https:/books. google.com/books?id=nvszMwEACAAJ

## BIOGRAPHIES

**Abhishek Tiwari**, pursuing B.Tech in Information Technology from Buddha Institute of Technology, Gorakhpur and at present working in Mobiloitte, Delhi as a trainee.

**Mansi Mishra**, pursuing B.Tech in Information Technology from Buddha Institute of Technology, Gorakhpur. At present, Student of B. Tech IV year.

**Ms. Pragya Kamal** , completed M.Tech(IT) from Madan Mohan Malaviya University of Technology, Gorakhpur in 2015.Working at BIT,Gorakhpur for 1.5 years as Assistant Professor in CS Department.

**Mr. Mayank Deep Khare**, completed M.Tech(Information Technology) from Madan Mohan Malaviya University of Technology, Gorakhpur in 2015. Working at NEIT,Greater Noida for last one year as Assistant Professor in CS Department.