A Review of Android operating system security issues

Mohamed Razeed Mohamed Nowfeek

Department of Information Technology, ATI-Sammanthurai, Sri Lanka Institute of Advanced Technological Education(SLIATE).

***Abstract:* with the growth and development in mobile phone operating systems and hardware technologies now Security issue is an immense challenge. Presently, in the market among all the smart-phone operating systems android has the major share. As the features and powers of such phones increase, their vulnerability also increases and makes them prone towards security threats. Permission -based model used by Android operating system which allows Android applications to access system information, device information, user information and external resources of Smartphone. in Android, application developer has to declare the permissions. For an Android application to be installed successfully, the user must approve certain permissions. These consents are declarations. if the consents are allowed by the user, At the time of installation, the app can access information and resources anytime. Again, Permission need not request. Because of its security vulnerabilities, the Android operating system is prone to a variety of security attacks and vulnerabilities. In this review paper, the author has made a methodical study on why android operating system security is significant, summarizes the security attacks and issues of android operating system, what some of the possible susceptibilities are and what security measures have been implemented currently to ensure security and proposed solution.**

***Keywords:* security issues, Android operating system, Smart phone operating system, vulnerability.**

I. INTRODUCTION

There are many different types of mobile operating systems on the market. Android is a one of the mobile operating system which runs on Linux kernel. The Android working framework is open source, with the source code released under the Apache license. These codes are used to control mobile device via Google–enabled java Android mobile application developed based on Java Programming libraries. It is an Important platform to develop mobile application using software stack provided in the Google.

The security platform is far superior to Blackberry or J2ME platforms. In most cases, programs can't write or read each other's code. Android SDK. Android combines OS features like Unix User, efficient shared memory, preemptive multi-tasking Identifiers (UIDs) and file permission with Java language and its class library.

Android had an 84.1 percent market share in the first quarter of 2016, while iOS, BlackBerry, Windows, and others had 14.8 percent, 0.2 percent, 0.7 percent, and 0.2 percent, respectively. [2]. Android was one of the most popular smartphone operating systems in the third quarter of 2016. [1]There were 2.6 million applications accessible in the Google Play Store. [1] and a total number of Android operating system-based smartphones sold was 2.1 Billion Therefore, When compared to other mobile operating systems, it is evident that Android has the biggest market share. Apple Inc. developed iOS (iPhone OS), which is only available on Apple devices including the iPhone, iPod, and iPad touch.After Android, iOS is the most popular operating system. [2]. It is possible to install software from unidentified sources. Aside from the Google Play Store, there are a few more options for Android users. But, the apps can be only installed from AppStore, in iOS. It's one of Android's most serious security flaws.Due to various security breaches.There areseveral efforts have been put forward for addressing the security related issues in Android operating system and understanding current these shortcomings scenarios.

II. LITERATURE REVIEW

Many researchers have found and studied the security issues on Android operating system and mechanisms to overcome issues up to certain extent.

MohdShahdi Ahmad et al. [2]offered a security comparison of iOS and Android, concluding that Android is more secure than iOS.

To isolate apps data and codes from other apps and developers can define permissions precise to their own apps Security features are surrounded in Android OS to lessen security issues related with apps Sandbox concept is implemented. Explicitly, the file system is encrypted against any theft or loss of devices. [4]

Authors [5] for the Android OS smartphone to accomplish the implicit authentication a multi-sensors-based system proposed by Wei-Han. This method also has the ability to update operator model. The test displays that the efficiency of this model only requires 10 seconds to run the model, 20 seconds to detect abnormal or fake request. In this model, the level of correctness attained can reach up to 90-95%. The system continually learns the user's actions patterns and setting by allowing the user to use a phone without troubling the operator's activities.

A. Kaur et al. [3]presented that it is possible to revoke granted permissions from android application.

Authors [6]suggest detailed assessment criteria determining the status of security of common OS such as Android, BlackBerry, Apple iOS, Symbian and Windows phone in the term of development of malware, and based on the proof of the study and give comparative analysis. However, the ease with which malware attacks can be developed in this example does not apply to other smartphone operating systems. In conclusion,they proposed solution against that malware, (a) operators to be aware, (b) using saves or giving applications. to inspect prevailing development of malware on smartphone Android platform and average programmer those have access library of smartphone and functionary tools

Hamandi*et al.* [7] proposed. They look at some of the message design flaws that lead to a series of vulnerabilities in the Android operating system, and they show how applications can be constructed for malware detection to prevent being harmed by this flaw. The application has been shown in limited control and successfully passed the standard inspection procedure intended to catch malware. To decline the risk of such attacks A set of possible solutions is also presented. SMS messages and use them fundamental truths to send/receive short messages, these applications look as a normal application. Because different operators throughout the world provide a service that allows customers to send credit/units by SMS, this facility is being abused to send credit unlawfully. Subsystem “permission”, subsystem "broadcasting receiver," and gathering mechanism to contribute to the establishment of a haven for SMS malware, giving them total control over the receiving, sending, and hiding SMS messages. Operators can utilize such subsystems to stream and balance malware attacks that have the potential to harm a large number of operators and telecommunications companies. Therefore, the application hides the malicious approval from telecom operators that can arise after the operation for credit transfer.

Zuquete and Decker [8], showed serious weaknesses in some private provider’s Android operating system. They defined the proof-of-concept for them, which may be used to investigate the effects of vulnerabilities like root access. supplier of application proposed advanced features to configure and control device, developed on purpose and with the aim stated. the installation of such "features" must be at least possible released to the user, in their observation, so the risks of an unprotected USB connection is recognized by them.

Escalation Attack /Counter Attacks: In [9], authors proposed a system These systems monitor the proposed scheme reallyused to call for the application method. for prevention and detection that protects Android operating system with features like escalation attack or counter-attacks that try to increase full access to all data. If the call system is called by special components of the Android system in normal process, from performing it, the rule prevented. The scheme can detect and block unknown and new malware.

III. SECURITY ATTACKS AND SECURITY ISSUES IN ANDROID

The Android operating system's security is built around a permission-based framework that controls and regulates third-party Android apps' access to crucial resources.By developers' end-users and marketers, this permission-based technique is commonly criticized for controlling application permission and ineffective permission administration.Users can either accept or reject all permission requests when installing an app. The information of Android OS users is prone to leakage, putting their privacy at risk. Android operating system major security attacks and issues will be discussed, here.

a. Spyware

One of the main causes of serious security issues in the Android operating system is spyware.Spyware is a type of malicious software. When a user installs programs from unknown sources and visits a malicious website, an apk file is immediately downloaded. It is possible to install programs from anonymous sources in Android, in addition to the Google Play Store.

b. Permission Escalation Attack

Obviously, access crucial resources without seeking the necessary authorization.It enables a malicious program to collaborate with other programs. [10]

c. Information leakage

Information leakage occurs when without any limit from OS the users grant resources. However, the Android operating system's permission control mechanism ineffectively protects users' privacy and resources from malware. The sensitive information leaked therefore the device to be is a critical state. The exploitation of this vulnerability is veryeasy as an attacker can gain an access to the part of the device where the sensitive data is being stored. The leaking Android application mayThe leaking Android application may store sensitive user information in an unsecured location on the device or communicate device identification information, such as application metadata such as network details, to a third party. place an information that is sensitive for the user in the insecure location in the device or may send the device identification information e.g. application metadata such as network details. Other malicious programs on the same device may have access to this device's insecure location. The impacts of the data leakage of an Android device are severe. As per a security researcher group news website 58% of Android devices have privacy leaks and around 3% have PII (personally identifiable information) leakage. [11]

d. Colluding

From the side of the users colluding threat is happening. Users install a series of apps with the same certificate and grant various permissions, which could be critical or non-sensitive.After being installed, these apps can use a shared

UID to gain access to all of their resources and permissions. [12]

e. Denial of Service Attack

Overusing limited CPU, memory, battery power and network bandwidth are the primarily objectives of DoS attacks. The increasing number of mobile devices which they are connected to the Internet as a large network which could be a step for growth of DoS attacks. Because the smartphones are having less protections or not fortified compared with PCs, the creators of malicious applications identified it as a suitable platform for DoS attacks. [13]

f. Repackaging Apps

Repackaging is one of the most serious and widespread security concerns with the Android operating system. On the Android platform, repackaging techniques allow dangerous code to be disguised as a normal appIt's tough to tell the difference between a repackaged harmful code and a valid program because the repackaged app functions similarly to the original one.Repackaging is the process of disassembling/decompiling of .apk files using reverse-engineering techniques and adding (injecting) malicious code into the main source code. [14]

Repacking: apktool was used to rebuild the files, and jarsigner was used to sign the rebuilt files. The trojans Geimini and KungFu are instances of APK repackaging trojans. Many legitimate Android apps can include these trojans.

IV. PROPOSED SOLUTION/ANALYSIS

Some security solutions for the Android operating system have been presented, and this section divides them into two categories. those are Dynamic, Static which both can use for vulnerability analysis, assessment, and detection. Dynamic methods, time-consuming, are exceptionally suitable when applications are really obscured. Static methods are fast, yet it needs to manage false-positives sensibly. Hybrid methods also there that merge together with the limitation of both static and dynamic methods.

a. RiskMon

Authors [15] proposed RiskMon. It creates a risk assessment baseline that incorporates appropriate application behaviors by combining runtime actions and users' expectations of trustworthy applications. Figure 1 illustrates the basic architecture of the RiskMon.

RiskMon is a machine-learning solution to dealing with this problem, and it provides a framework for continuous and automated risk assessment.

applications are the key part of the framework on users' perceptions. First, it collects the user's expectations on the installed apps on the device and the ranking of permission groups in terms of their relevancy to the matching application. Then, based on the collected information from the user, It establishes a risk assessment foundation for it applications. Finally, RiskMon ranks installed apps depending on the danger of their interactions, which is evaluated by how far they differ from the risk assessment baseline. [15]

RiskMon does not address interactions between interactions that do not use Binder and third-party programs when it comes to implementation. This, indeed displayspossible attack vectors that can bypass RiskMon.

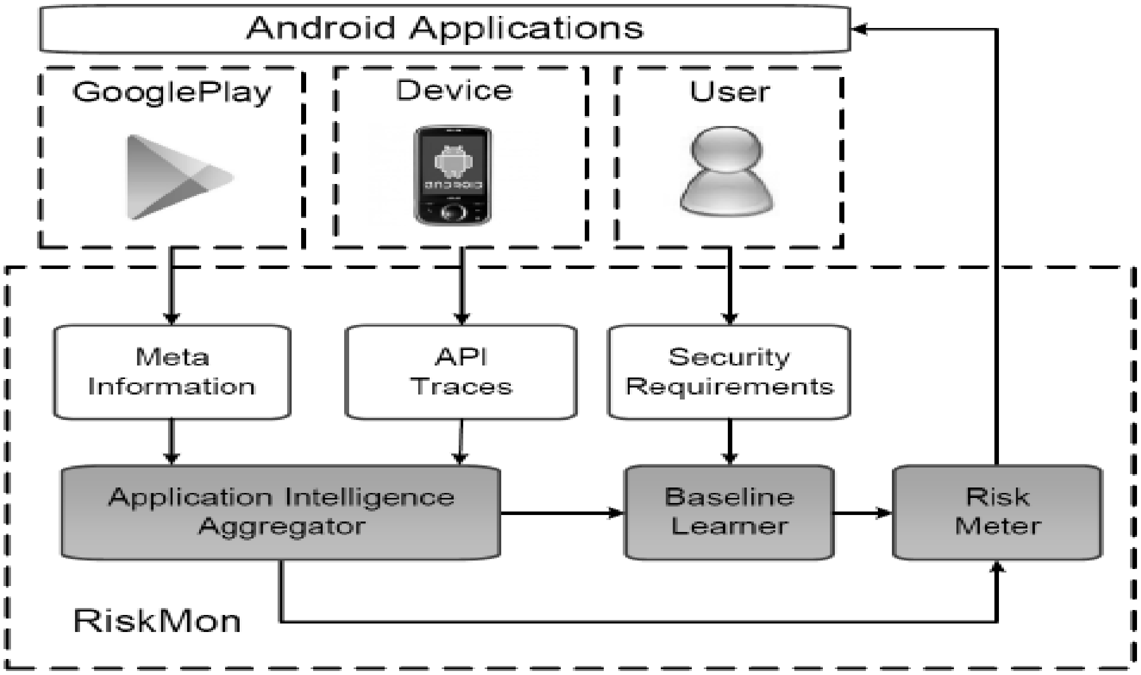


Figure 1:RiskMon architecture

b. Kirin

Authors [16] proposed Kirin practices a set of predefined security rules on apps' requested consents to find corresponding malicious permission requests and characteristics. Kirin's major purpose is to prevent harmful apps from being installed by employing a certification mechanism for apps. [46]. Here, the rules are defined based on those permissions that are complex and primes to mistreating of authorizations and unsafe activities. [16]

Using this system at install time can help operators to make real-time decisions whether installing the apps or not. from top ranked applications from an official Android app Market They tested the Kirin using 311 downloaded apps. Kirin detects 5 malicious apps with a high level of security risk, After experiments. Figure 2 illustrates the Kirin based components and its software installer flow. They use a static analysis tool called Pscout in order to extract all permission specifications for Android apps without altering the apps.

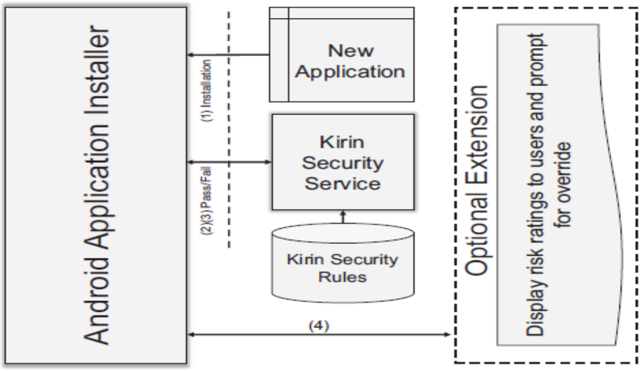


Figure 2:Kirin based software installer flow and its components

c. Crowdro

Authors[17]proposed A framework for analyzing the behavior of Android apps that is useful for distinguishing between apps with similar names and versions that behave differently. A crowdsourcing framework detects abnormally behaving applications. Crowdroid is a malware detection technology that is based on behavior. Figure 3 illustrate the Crowdroid architecture.

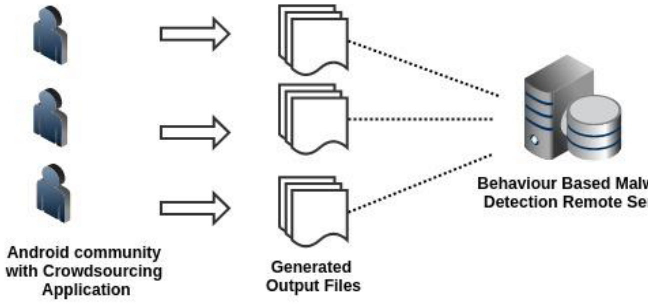

Android community with Crowdsourcing Application
Generated Output Files
Behaviour Based Malv Detection Remote Se

Figure 3: Crowdroid Architecture

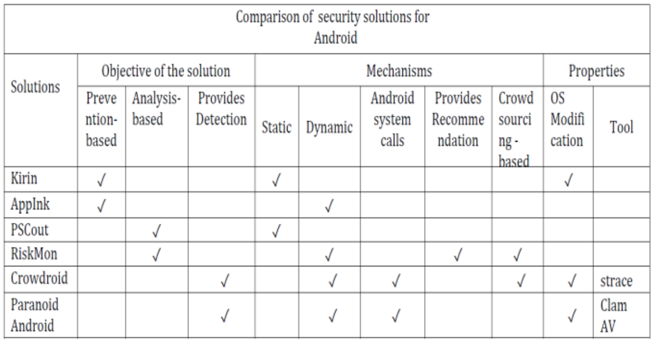
d. Paranoid Android

Authors [18]proposed A security check system in Paranoid Android. The main feature of the Paranoid Android is that the checking process from the operator device is moved to a remote server. The key reason behind the security checks on a remote server is the lack of enough computational resources and battery consumption. This is used in conjunction with a remote security server (cloud-based detection framework) that houses exact duplicates of phones in virtual environments.

There are two stage process is surveyed as a part of security check mechanism. In the stage first, the app monitoring is performed and the same is followed by the device. In this stage, the app's activities are monitored and logs are collected and shifted to the server. The log files are directed only if the device is awake to avoid and decrease the log file transfer overhead. In the second stage compromises of analysis of the collected logs from devices. Paranoid Android uses a ClamAV based antivirus for file scanning.

e. DroidScope

Authors[19] proposed DroidScope by this, the privilege escalation attacks can be detected, even in the kernel. make a set of APIs available for human analysts to adjust their analysis needs It also complicates the attackers' aim of chaotic analysis. upon QEMU emulator the DroidScope is built, and likewise, DroidScope is a Virtual Machine Introspection (VMI) that is for Android applications is dynamic analysis framework. Unlike other dynamic analysis frameworks, it does not reside inside the emulator, instead generating OS- and Dalvik-level semantics from outside the emulator.



Comparison of security solutions for Android

Solutions	Objective of the solution			Mechanisms					Properties	
	Preve ntion-based	Analysis-based	Provides Detection	Static	Dynamic	Android system calls	Provides Recomme ndation	Crowd sourci ng - based	OS Modifi cation	Tool
Kirin	√			√					√	
AppInk	√				√					
PSCout		√		√						
RiskMon		√			√		√	√		
Crowdroid			√		√	√		√	√	strace
Paranoid Android			√		√	√			√	Clam AV

Security solution comparison- Android operating system

Figure 4: Security solution comparison- Android operating system

V. CONCLUSION AND FUTUTR RESEARCH

In these days, Most widely used mobile operating system is Android. There are some advanced features in android. But there are threats and attacks include in this platform, like malware applications. because malware on android platform create various risks. To protect personal information and user privacy, the security of an Android operating system is critical. Android operating system security attacks and issues have been reviewed in this paper. in order to prevent and control android operating system security attacks and issues there are various solutions also have been reviewed in this paper.

Some possible future works of Android operating system is to concentrate on How to develop a more secure Android OS in near future.

REFERENCES

[1] "Smartphone users worldwide 2014-2020 | statistic," Statista," 2020. [Online]. Available: https://www.statista.com/statistics/330695/number-ofsmartphone-users-worldwide.

[2] N. E. M. R. N. R. H. a. N. O. M. S. Ahmad, "Comparison between android and iOS operatingsystem in terms of security," in 8th International Conference on Information Technology in Asia (CITA), 2016.

[3] A. K. a. D. Upadhyay, "Modifying application’s permissions and preventing information stealing on smartphones," in 5th International Conference -Confluence The Next Generation Information Technology, 2016.

[4] "Android Security," [Online]. Available: http://developer.android.com/training /articles/security-tips.html.

[5] W. L. a. R. Lee, "Multi-sensor authentication to improve smartphone security," in Conference on Information Systems Security and Privacy, 2016.

[6] A. Morris, "Multimodal person authentication on a smartphone under realistic conditions," in in Defense and Security Symposium, 2016.

[7] A. C. I. H. E. a. A. K. K. Hamandi, "Android SMS Malware: Vulnerability and Mitigation," in 27th International Conference on Advanced Information Networking and Applications, 2017.

[8] B. D. D. a. A. Zúquete, "Communications and Multimedia Security," Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 8735, 2015.

[9] D. K. M. P. a. S. C. H. Lee, "Protecting data on android platform against privilege escalation attack," International Journal of Computer Mathematics, pp. 1-14, 2015.

[10] W. H. a. Y. L. Z. Fang, "Permission based Android security: Issues and countermeasures," Computers &Security, vol. 43, p. 205–218, 2016.

[11] "Android data leakage," [Online]. Available: http://www.appstechnews.com/news/2016/oct/25/research-reveals-ios-and-android-app-data-leakage-and-what-it-means-enterprises/.

[12] A. F. a. S. C. C. Marforio, Application Collusion Attack on the Permission-Based Security Model and Its Implications for Modern Smartphone Systems, 2010.

[13] E. Kovacs, "flaw exposes android devices to dos attacks," [Online]. Available: http://www.securityweek.com/wi-fi-direct-flaw-exposes-android-devices-dos-attacks.

[14] S. Z. P. L. a. D. W. H. Huang, "A framework for evaluating mobile app repackaging detection algorithms," in Proc. of the 6th International Conference on Trust and Trustworthy Computing.

[15] G.-J. A. Z. Z. a. H. H. Y. Jing, "Riskmon: Continuous and automated risk assessment of mobile applications," 4th ACM Conference on Data and Application Security and Privacy (CODASPY’14), p. 99–110, 2016.

[16] M. O. a. P. M. W. Enck, "On lightweight mobile phone application certification," 16th ACM Conference on Computer and Communications Security , p. 235–245, 2015.

[17] I. B. a. U. Zurutuza, "Crowdroid: Behavior-Based Malware Detection System for Android".

[18] K. A. G. P. Philip Homburg, "Paranoid Android: Versatile Protection for Smartphones".

[19] L. K. Y. a. H. Yin, "Droidscope:Seamlessly reconstructing the os and dalvik semantic views for dynamic android malware analysis," in 21st USENIX Conference on Security Symposium, 2015.