International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 5 ǁ 2022 ǁ PP. 42-46

Solution Driven Study on Android Security Concerns

Pratyush Kumar
PG Scholar, Department of MCA,
Dayananda Sagar College of Engineering, Bengaluru, Affiliated to VTU.

Mahendra Kumar
Assistant Professor, Department of MCA,
Dayananda Sagar College of Engineering, Bengaluru, Affiliated to VTU.

Abstract— *Android operating system uses the permission-based model which allows Android applications to access user information, system information, device information and external resources of Smartphone. The developer needs to declare the permissions for the Android application. The user needs to accept these permissions for successful installation of an Android application. These permissions are declarations. At the time of installation, if the permissions are allowed by the user, the app can access resources and information anytime. It need not re- request for permissions again. Android OS is susceptible to various security attacks due to its weakness in security. This paper tells about the misuse of app permissions using Shared User ID, how two- factor authentications fail due to inappropriate and improper usage of app permissions using spyware, data theft in Android applications, security breaches or attacks in Android and analysis of Android, iOS and Windows operating system regarding its security.*

Keywords— *Android; Permissions; Shared User ID; Security; Data Theft; Spyware; iOS; Windows.*

Date of Submission: 12-05-2022 Date of acceptance: 26-05-2022

I. INTRODUCTION

A versatile working framework (OS) is programming that permits cell phones, tablet PCs, and different gadgets to run applications and projects. There are several types of mobile operating system available in the market. The commonly used mobile operating systems are Android, iOS, Windows and BlackBerry OS. The Android working framework is an open source and source code discharge by Google under Apache permit license, based on Linux-Kernel designed for smartphones and tablets. Android is one of the most popular operating systems for smartphones. At the last quarter of 2016, the total number of applications available in Google play store was 2.6 Million [1], and a total number of Android operating system-based smartphones sold was 2.1 Billion [2]. The market share of Android in the first quarter of 2016 was 84.1% whereas iOS, Windows, BlackBerry, and others hold 14.8%, 0.7%, 0.2% and 0.2% respectively [2]. Therefore, it is clear that Android has the widest market when compared to others mobile operating systems. iOS (iPhone OS) developed by Apple Inc. and used only by Apple devices such as iPhone, iPad, and iPod touch. It is the second most popular operating system next to Android [2]. In Android, other than google play store, it is possible to install the applications from unknown sources. But, in iOS, the apps can be only installed from AppStore. It is one of the major security breaches in Android. Due to various security breaches in Android, attackers already regard smartphone as the target to steal personal information using various malware. In 2013, Mohd Shahdi Ahmad et al. [3] indicated the analysis of Android and iOS regarding security and declared iOS more secure than Android. In 2014, A. Kaur et al. [4] indicated that it is possible to revoke granted permissions from android application.

The rest of the paper organizes as Section II describes various security attacks on Android such as permission escalation attack, confused deputy attack, direct collision attack, indirect collision attack and TOCTOU (Time Of Check and Time of Use) attack. Section III describes different types of Android app permissions, over-claiming of app permissions, misuse of app permissions using Shared User ID and failure of two-factor authentication in Android-based smartphones due to spyware. Section IV presents the comparison of security between Android and iOS. Section V presents the proposed method to avoid misuse of app permissions and the conclusion of the paper.

II. SECURITY ATTACKS IN ANDROID

A. Permission Escalation Attack

It allows a malicious application to collaborate with other applications so as to access critical resources without requesting for corresponding permissions explicitly [5][6].

B. Collision Attack

Android supports shared user ID [5][7]. It is a technique wherein two or more application share the same user id so that they can access the permissions which are granted to each other. For example. If application A has permissions to READ_CONTACTS, READ_PHONE_STATUS and B has permissions to READ_MESSAGES, LOCATION_ACCESS, if both the applications use the same user id SHAREDUSERID, then it is possible for application A to use the permissions granted to itself and the permissions granted to B. Similarly, it is possible for application B to use the permissions granted to itself and the permissions granted to A. Every Android application has unique ID that is its package name. Android supports shared User ID. It is an attribute in AndroidManifest.xml file. If this attribute assigned with the same value in two or more applications and if the same certificate signs these applications. They can access permissions granted to each other.

Collision attack has been classified as direct collision attack and indirect collision attack. A direct collision attack is wherein application communicates directly. In Indirect collision attack application communicates via third party application or component.

C. Time of Check and Time of Use Attack

The main reason for TOCTOU Attack is naming collision. No naming rule or constraint is applied to a new permission declaration. Moreover, permissions in Android are represented as strings, and any two permissions with the same name string are treated as equivalent even if they belong to separate applications.

D. Spyware

Spyware is a type of malware. It is an apk file which is downloaded automatically when the user visits malicious website and apps installed from unknown sources. In Android, other than google play store, it is possible to install the applications from unknown sources. Spyware is one of the main reasons for major security threats in Android operating system.

III. UNDERSTANDING PERMISSIONS

The Android operating system uses the permission-based model to access various resources and information. These permissions are not requests; they are declarations. These permissions are declared in AndroidManifest.xml file. Once the permissions are granted, the permissions remain static for Android versions less than 6 [8][9]. But, in Android versions,
7.0 and higher the app permissions are classified into normal permissions [10] and dangerous permissions [11].

A. Normal Permissions

Normal permissions don't specifically hazard the client's privacy. Normal permissions need not be declared in the AndroidManifest.xml file. These permissions are granted automatically. Example:

KILL_BACKGROUND_PROCESSES SET_WALLPAPER UNINSTALL_SHORTCUT
WRITE_SYNC_SETTINGS

B. Dangerous Permissions

Dangerous Permissions can access critical resources of the mobile. Dangerous permissions can give the app access to the user's confidential data. If app lists a normal permission in its manifest, the system grants the permission automatically. If app list a dangerous permission, the user has to explicitly give approval for the app for the successful installation of the app. Example:
CONTACTS
READ_CONTACTS, WRITE_CONTACTS, GET_ACCOUNTS
LOCATION ACCESS_FINE_LOCATION,
ACCESS_COARSE_LOCATION SMS
SEND_SMS, RECEIVE_SMS, READ_SMS, RECEIVE_WAP_PUSH, RECEIVE_MMS
STORAGE READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE

Android Marshmallow 6.0 has classified the permissions into normal and dangerous permissions. Whenever the app needs to use dangerous permissions, it explicitly asks the user to confirm with the permission. Thus, Android 6.0 and higher versions provide explicit permission notification to access critical resources. But, Marshmallow is available only on 1.2 percent of Android devices [9]. The Android operating system updates are not available for most of the older devices. Therefore, security threats related to app permissions are still not solved.

C. Application Sandboxing

Android uses application sandboxing which is used to limit the application to access the resources. If an app needs to access the resources outside of its sandbox, it needs to request the appropriate permission.

D. Over-claiming of application permissions

The permissions which may not be required for the app, but the application request for the particular permission, this is called over claiming of permissions. It is the declaration to use irrelevant permissions that are not at all required for the application. It is the main reason for data theft in android application. The information is collected and sent to the concerned people. The developer's of the app makes money by selling this information. Several third parties buy this information for various reasons like data mining etc., For example, in FlashLight Android app permission is given for full internet access. It is irrelevant for flashlight application to have internet access. Ashmeet Kaur et al. [4] developed a framework wherein it is possible to remove the unnecessary permissions from the app, once the app has been successfully installed.

E. Misuse of App permissions and failure of two- factor authentication

Due to misuse of various app permissions, it is possible for various security threats. Among various threats, it is possible for Android applications to read messages, send messages. SMS is a common and basic functionality in traditional mobile and smartphone. All confidential information based on two-factor authentication has been sent as a text message. For example, various banks, online websites, etc., use two-factor authentications. The main objective of two-factor authentication is to increase the security and integrity for the users and to avoid various security attacks that are based on traditional username and password approach. But, even this method fails, if malware installed in a smartphone or due to over claim permission apps. If the hacker hacks username and password of the user using various hacking techniques, the first level of authentication are compromised and then the OTP (One Time Password) is being sent to the user. If the application or malware that is being installed in Smartphone then it is possible for the app or malware to read messages and send the information to the hacker without the knowledge of the user. So, even two-factor authentication fails.

IV. COMPARISON OF ANDROID AND IOS

A. Application Downloads

The Android applications can be downloaded from google play store and unknown sources. Android uses crowdsourcing [12] which is based on user comments and rating of the app. If enough users complain about the app, then it will be removed and deactivated remotely. The iOS applications can be downloaded only from iOS AppStore. It is not possible to download and install iOS applications other than AppStore. All the applications available in iOS have been properly checked for various security issues in the source code and after verifying it then it is available in the AppStore.

B. Signing Technology

Self Signing [13] is used in Android. The Android discharge framework requires that all applications introduced on client gadgets are carefully marked with declarations whose private keys are held by the designer of the applications. The endorsements permit the Android framework to recognize the creator of an application and set up trust connections amongst designers and their applications. The endorsements are not used to control which applications the client can and can't introduce. Code signing [14] [15] used in iOS. It app assures users that it is from a known source and the app hasn't been modified since it was last signed. Before publishing an app, the app has to be submitted to Apple Inc. for approval. Apple signs the app after checking the code for any malicious code. If an app is signed then, any changes to the app can be easily tracked.

C. Interprocess Communication

Android supports interprocess communication among its applications. Apple iOS does not support inter-process communication among its applications.

D. Open Source and Closed Source

Android is open source. In this guideline, open source programming implies the source code is made accessible on an all inclusive level. The thought is to open up the product to the general population, making a mass coordinated effort that outcomes in the product being continually upgraded, settled, enhanced, and developed. Apple's iOS is closed source. With closed source software, the source code is firmly watched, regularly in light of the fact that it's viewed as a prized formula that makes shortage and keeps the association aggressive. Such projects accompany limitations against changing the product or utilizing it in courses intended by the first makers.

E. Memory Randomization

It is a technique wherein the information about the application is stored on the disk in the random address which has been generated. This reduces the security threats since malicious code and attacker needs to find the exact location where the information is being stored. This technique is used by both iOS and Android OS.

F. Storage

Data of application is stored either in internal storage or external storage. For Android, the information can be stored in both built in storage and external storage. But, iOS does not support external storage. It has only internal storage to reduce various security threats and faster processing.

V. PROPOSED METHOD

Android shared user ID is one of the major reasons for misusing app permissions. Due to shared user ID permissions granted to one app can access permissions granted by another app if and only if both has the shared user ID value set same and signed by the same certificate. The users are not aware of which applications are misusing the permissions. In the proposed method, an Android security tool is developed.

This procedure includes six steps

- List all the applications based on its app ID that is its package name.
- List all the applications for which shared User ID is set.
- Compare all the applications with every shared User ID set app.
- List the finalized apps.
- Provides explicit notification to the user when the shared User ID app tries to access the permissions with other apps.
- Display the resources used by shared user ID apps by the security tool app.

VI. CONCLUSION

Android is most widely used mobile operating system. Improvising the security of an Android OS is very important to safeguard the user's privacy and confidential information. In this study, it was shown how to avoid misusing app permissions.

REFERENCES

2014. [Online]. Available: http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store.

[1]. "Number of Google play store apps 2016 | statistic," Statista,
[2]. "Smartphone users worldwide 2014-2020 | statistic," Statista, 2016. [Online]. Available: https://www.statista.com/statistics/330695/number-of- smartphone-users-worldwide.
[3]. M. S. Ahmad, N. E. Musa, R. Nadarajah, R. Hassan, and N. Othman, "Comparison between android and iOS operating system in terms of security," 2013 8th International Conference on Information Technology in Asia (CITA), Jul. 2013.
[4]. A. Kaur and D. Upadhyay, "PeMo: Modifying application's permissions and preventing information stealing on smartphones," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), Sep. 2014.
[5]. Z. Fang, W. Han, and Y. Li, "Permission based Android security: Issues and countermeasures," Computers & Security, vol. 43, pp. 205–218, Jun. 2014.
[6]. Y. Zhou and X. Jiang, "Dissecting Android malware: Characterization and evolution," Proc. - IEEE Symp. Secur. Priv., no. 4, pp. 95–109, 2012.
[7]. L. Xing, X. Pan, R. Wang, K. Yuan, and X. Wang, "Upgrading Your Android, Elevating My Malware: Privilege Escalation Through Mobile OS Updating," IEEE Symp. Secur. Priv., 2014.
[8]. L. Whitney, "Almost no one is using Android marshmallow, still," CNET, 2016. [Online]. Available: http://www.cnet.com/news/almost-no-one-is-using-android- marshmallow-still.
[9]. V. Savov, "Only 7.5 percent of Android phones are running marshmallow," The Verge, 2016. [Online]. Available: http://www.theverge.com/circuitbreaker/2016/5/4/11589630/ android-6-marshmallow-os-distribution-statistics.
[10]. "Normal Permissions,". [Online]. Available: https://developer.android.com/guide/topics/security/normal-

permissions.html.
[11]. "Dangerous Permissions,". [Online]. Available: https://developer.android.com/guide/topics/security/permissio ns.html#normal-dangerous.
[12]. J.-K. Park and S.-Y. Choi, "Studying security weaknesses of Android system," International Journal of Security and Its Applications, vol. 9, no. 3, pp. 7–12, Mar. 2015.
[13]. "Sign your App,". [Online]. Available: https://developer.android.com/studio/publish/app-signing.html#certificates-keystores.
[14]. A. Inc, "About code signing," 2012. [Online]. Available: https://developer.apple.com/library/mac/documentation/Secur ity/Conceptual/CodeSigningGuide/Introduction/Introduction. html.