International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Android Operating System Security Issues

Khushboo Lokhande, Vishal Patil
[1]Anuradha Engineering College, Chikhli Sant Gadge Baba Amravti University, Maharashtra, India
[2]Asst. Prof., Anuradha Engineering College, Chikhli Sant Gadge Baba Amravti University, Maharashtra India

ABSTRACT

Androids are operating systems, which are mostly used. Android are the primarily used touch screen devices. It is an open source operating system that is it has no limitations regarding cost and access to data.The android operating system permits the access to resources and information anytime. Android operating system permits the access to all kind of data required for successful installation of an android application. Whenever user install an application the user accepts the permissions ,which are actually declarations . If it is accepted the application can access data anytime. This paper tells about the misuse of app permissions using Shared User ID, how authentications fail due to inappropriate and improper usage of app permissions using spyware, data theft in Android applications, security breaches or attacks in Android , iOS and Windows operating system regarding its security.

Keywords : Android Operating, Security Issues, iOS, Fake ID, SOP Vulnerability, GinMaster

I. INTRODUCTION

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.In most cases, however, these warning don't help much. If a user installs an app that promises to make it easier to communicate with others, it's obvious that he or she will press OK after reading these warnings. It might help if Android could alert the user at the moment the app starts sending text messages, but no such functionality is available in the current release. For the time being, anti-malware software can provide some protection against Android app security risks.In android os,we can install applications from play store as well as from unknown sources. it is one of the major security breaches in android operating system. Due to various security breaches in Android, attackers already regard smartphone as the target to steal personal information using various malware.

The rest of the paper organizes as Section II describes various security attacks on Android. Section III describes different types of Android app permissions. 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.

Security attacks in Android

1.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].

2.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.

3. 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.

4. Spyware is a type of malware that's hard to detect. It collects information about your surfing habits, browsing history, or personal information (such as credit card numbers), and often uses the Internet to pass this information along to third parties without you knowing. Keyloggers are a type of spyware that monitors your keystrokes.

II. REAL-WORLD ATTACKS

This section details some real vulnerabilities that have been found in the Android OS.

1. Fake ID

All android applications have their own unique identity, and there was a vulnerability that allowed identities to be copied so that one application could impersonate another. This â€œFake IDâ€ breach allowed malicious applications to be recognized as a trusted one by the user without the user knowing about it. This could potentially allow malicious software to steal user information from a trusted application and even take control of the security mechanisms on a device [FakeID].The problem arises from the Android package installer not verifying the validity of a chain of certificates. Normally an applicationâ€™s certificate is verified before installing it or updating a version. However, an identity can claim to be issued by another identity, providing a certificate that could potentially be malicious as well as the verified one. The malicious certificate will be ignored since the certificate chaining verification was not done properly, and this will allow that malicious application access to the trusted application.

2. SOP Vulnerability

The Same Origin Policy is a security mechanism that is necessary for web application security. The policy allows scripts that originate from the same site to access information from that site found in the DOM or elsewhere, but not to access any information found on pages from different sites. This prevents a

web application from reading information that is open in another tab that a user might be using at the same time. There was a vulnerability with the Android browser AOSP that allowed hackers to bypass the SOP, so they could access sensitive information open in a userâ€™s email tab when the user is on a different site. This was done by sending a malformed JavaScript: URL handler with a null byte, which led to the SOP not being enforced. This has been fixed as the AOSP browser is actually no longer available on the Android devices. Google has released mobile versions of Chrome that do not have this issue so the problem has been resolved [Hoog11].

3.GinMaster

Android GinMaster (short for GingerMaster) is a Trojan application family that is primarily distributed through Chinese third party stores that infects Android devices. It was originally named GingerMaster as it attacked Android version 2.3 which was named Gingerbread. The attacks were first found in 2011 and continued for over 2 years. The newer variants of GinMaster were able to avoid detection by most anti-virus software in order to get into devices. Using polymorphic tecchniques, the program would obfuscate class names for infected objects and randomize package names and certificates for applications. Other functionality of this malware was to steal confidential information, gain more permissions on the device, and install applications without user approval [Yu13]. There are many other similar Trojan or other malicious applications and GinMaster is just a single example of this type of problem. Other general categories of Android malware are Rootkit, Trojan spy, Malicious downloader, Click fraudster, Data stealer, and Premium service abuser. Users should always double check an application's permissions and whether or not they are getting the application from a trusted source before downloading.

4. Master Key

The Master Key vulnerability was found by researchers at SophosLabs in 2013. Usually, when an application is installed, the Android Package, or APK, verifies that all of the necessary files and certificates for installation correctly check out. If one were to put two files with the same name into the APK (which would normally serve no purpose), the Android device verifies the first file, but then installs and uses the second one. This means that if you were able to borrow another party's package, programs, and other data, you could run and install something that this party has never approved or seen before. In that sense, it works sort of like a master key although it does not actually crack any cryptographic keys in the Android system. This was a flaw with the Android OS as it should not have been doing the check in this way but after it had been identified it was patched.

UNDERSTANDING PERMISSIONS

The Android mobile OS uses a permission system to help ensure that apps behave appropriately. The system allows apps to request access to features on the device that can use power, access sensitive data, and incur charges.

This isn't just behind the scenes, though — your choices can affect how users perceive the app. It is worthwhile for you to carefully consider what permissions your app requests, because users can verify the appropriateness of those permissions before installing the app. If they deem a permission to be inappropriate, they might post a negative comment about your app.

1. Permission: "Normal"

The default value. A lower-risk permission that gives requesting applications access to isolated application-requesting level features, with minimal risk to other applications, the system, or the user. The system automatically grants this type of permission to a

requesting application at installation, without asking for the user's explicit approval (though the user always has the option to review these permissions before installing).

2. Permission: "Dangerous"

A higher-risk permission that would give a requesting application access to private user data or control over the device that can negatively impact the user. Because this type of permission introduces potential risk, the system may not automatically grant it to the requesting application. For example, any dangerous permissions requested by an application may be displayed to the user and require confirmation before proceeding, or some other approach may be taken to avoid the user automatically allowing the use of such facilities.

3. Permission: "Signature"

A permission that the system grants only if the requesting application is signed with the same certificate as the application that declared the permission. If the certificates match, the system automatically grants the permission without notifying the user or asking for the user's explicit approval.

4. Permission: "SignatureOrSystem"

A permission that the system grants only to applications that are in the Android system image or that are signed with the same certificate as the application that declared the permission. Please avoid using this option, as the signature protection level should be sufficient for most needs and works regardless of exactly where applications are installed. The "signatureOrSystem" permission is used for certain special situations where multiple vendors have applications built into a system image and need to share specific features explicitly because they are being built together.

Difference between Android and IOS

Android was started by android Inc. But, purchased by Google in 2005.Android is an Open source operating system.Android is based on Linux kernel.Rooting is the term used when it is associated with Android. It is the process that allows users of cellphones or other devices to gain privileged control within Android's Linux system.Android Open Source Project (AOSP) is responsible for managing maintainable and development.Due to its open source nature, There are more than 100000 apps which can be found on Android Market (online app store run by Google).Android Operating System versions are 1.1,1.5 (Cupcake), 1.6 (Donut), 2.0 / 2.1 (Eclair), 2.2 (Froyo), 2.3 (Gingerbread) and 3.0 (Honeycomb) (which will be launched in early 2011).Android os can be run on cellphone, netbook, tablet PC's, Dell streak, samsung galaxy tab, etc.,. It means it can run on any hardware which supports it. First phone to run on Android OS was HTC Dream. World's First TV running on Android is Scandinavia launched by People of Lava. Google's Smartphone running on Android are Nexus One manufactured by HTC and Nexus S manufactured by Samsung.

iOS was launched by Apple on June 29, 2007.iOS is not an open source operating system. It is Apple's Mobile operating system.iOS is based on Mac OS X operating system.Jailbreaking or Jailbreak is the term used when it is associated with iOS. It is the process that allows iPad, iPhone, iPod Touch and Apple TV users to gain root access to the command line of the iOS operating System.iOS is maintained by Apple.There are more than 300,000 iOS applications which can be found at MarketPlace (online app store run by Apple).iOS Operating System versions are 1.x, 2.x, 3.x and 4.x. The current version for iPad, iPod Touch and iPhone is 4.2.1. 4.1 version is the current version for Apple TV.iOS Operating System can only run Apple's hardware (cannot run on third party hardware). It runs on iPad, iPod Touch, iPhone and Apple TV

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:

1. List all the applications based on its app ID that is its package name.
2. List all the applications for which shared User ID is set. 3. Compare all the applications with every shared User ID set app. x List the finalized apps.
3. Provides explicit notification to the user when the shared User ID app tries to access the permissions with other apps.
4. Display the resources used by shared user ID apps by the security tool app.

III. CONCIUSION

In day-to-day life android operating system plays an important role. From this paper we conclude that the updation of android operating system is very important for the safeguard of users privacy and confidential documents .we can avoid misuse of app permission as stated in our study

IV. REFERENCES

[1]. "Smartphone users worldwide 2014-2020 | statistic," Statista2016.[Online].Available: https://www.statista.com/statistics/330695/number-ofsmartphone-users-worldwide.

[2]. Types of android app permissions avaliable: https://discussions.soti.net/thread/what-are-on-demand-permissions.

[3]. "Normal Permissions,".[Online]. Available: https://developer.android.com/guide/topics/security/normalpermissions.html.

[4]. "Dangerous Permissions,".[Online]. Available: https://developer.android.com/guide/topics/security/permissio ns.html#normal-dangerous.

[5]. Difference between android and IOS. avaliable: http://www.differencebetween.co.in/mobiles/difference-between-android-and-ios/

[6]. Attacks in android ,Spyware avaliable : https://www.avast.com/c-spyware

[7]. Real world attacks.avaliable: http://www.cse.wustl.edu/~jain/cse571-14/ftp/android_security/index.html#sec3.1