Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

www.ijarcsse.com

Research Paper

Available online at: www.ijarcsse.com

Application Security in Android-OS VS IOS

Yogita chittoria[1], Neha Aggarwal[2]
Department of Computer Science and Engineering
Amity School of Engineering and Technology, Amity University, Noida, UP, India

ABSTRACT:In modern computer world security is most important facets of Mobile application development. And it is also the most complex and most of the time overlooked facet of mobile computing. It is because Mobile Computing is cloud enabled and highly network in nature. This paper will emphasise on security issues in mobile Apps on Android and IOS. We will give a comparison between security features of Mobile App on Android and IOS.

Security must be importantly considered in every phase of mobile App life cycle. From the development time to, Publish time, Install time and then to live execution. It is not like that we develop the application and then think about the security measures. It must be considered from the very first point of application development to the end of application execution.

Keywords: MOS - Mobile operating system, Android-OS, IOS, App Security

I. INTRODUCTION

Android is an MOS based on the Linux kernel with a user interface, mainly focused touch screen mobile devices such as Smartphone and tablet computers, since Android's source code is provided by Google under open source licence so various Mobile manufacturing companies like Samsung, HCL, etc which uses it as their MOS. IOS is a MOS developed by Apple Inc. and it is available only on Apple hardware such as iPhone, iPad, iPod Touch, and Apple TV.

There are a lot of applications which are developed for both of these MOS. Current number of Android apps in the market are around:1,235,976[23] and for IOS are around 1,567,359[24]. While using these Apps on our mobile device or tablets or IPad, we should be careful about security of our data and personal information stored on the devices. The most commonly spread malware on Android is one where text messages are sent to some unknown numbers without the awareness of the user, and the sending of personal information to unauthorized third parties. When we talk about security in mobile Apps, we should take care of these facts like Root kit Detectors, Process isolation, file and resource permissions and memory protection.

Root kit are malwares which actually provide administrator rights to the attacker of the Mobile. Using Root kit detectors we can detect these malwares. Process isolation is a technology in MOS by which Apps has their own area of execution they will not interfere each other until some authenticated communication is provided. File and resource permission control is provided by every MOS so that we can achieve secure Inter process communication. Memory protection is achieved through Address Space Layout Randomisation (ASLR).

In this paper we are comparing security of Android-OS and IOS adopted by developers, App stores and end-users. Apps attacks can include following areas like, Attack on sensitive data of application, weak or no encryption during transmission of data, unauthorised access to administration interfaces or data stores. Runtime inoculation – This allows an attacker to manipulate and abuse the runtime of an application to bypass security locks, bypass logic checks, access privileged parts of an application, and even steal data stored in memory. Escalated privileges – Exploits a bug, design flaw or configuration oversight in order to gain access to resources normally protected from an application or user[1]

Our paper on comparison on security of Android and IOS is divided in four section. These sections are Section-1 will contain, Security at Application Development in Android and IOS. We will compare what are the checklist and points to consider while developing in Android and IOS app. Section-2 will contain - Security at Publish time in Apps stores of Android and Apple store. How Apple store and Android App store is different from each other. Section-3- will cover what are the Security feature at install time in IOS and Android OS. And to some extent security feature of IOS is good compare to Android-OS. In section-4 we will come security features of both OS while App is running on the Mobile. And our last section conclusion which includes final comparison of our paper.

II. SECURITY AT APP DEVELOPMENT TIME

As we know that no application is perfect. There are many bugs in Application, some are caused during design phase of application, some are occurred due to implementation mistakes done by developers. Due to these bugs user data is compromised and sometimes arbitrary binary data is executed by the application injected by the attackers.

In Android it is the responsibility of the developer to isolate application from other system resources through application sandboxing. For Android, the application sandboxing is based on the Linux kernel platform. It is a complex and robust sandbox model. Application sandboxing in Android is controlled by each application and required permission and approval to continue accessing what the application needed. This will improve and build the security tighter. Each application has its own sandbox directory and the permission is per App.

For IOS, the application sandboxing has been defined by Apple as a set of fine-grained control that limits the application access to the file system, network and hardware. IOS also has a robust sandbox model where all applications shared a same sandbox model which is more secure and less open to the crowd.

IOS is much better and more secured since it is only allowed users to access the system file in the root and the settings of the phone not in each application. But Android relies more on user because it required user to set the security for each application during installation time.[14]

The protection of data stored on a mobile device is perhaps one of the most important issues that an application developer has to deal with. It is imperative that developers protect sensitive data that is stored client-side in a secure manner. As previously noted, developers wishing to encrypt sensitive content on the device should employ the Data Protection API. Client-side data can be stored in a number of forms, including but not limited to: All of these may contain sensitive data that should be protected if the handset were lost or stolen. This data will generally be stored within the application's sandboxed container.

.IOS uses file system encryption and hardware encryption where individual files and key chains can be encrypted using data protection API's. This uses keys which are derived from device pass code given by user. Third party applications if want to encrypt sensitive data must use Data Protection API.

Developers must carefully specify required permission of resources since every resource has read-write-execute permissions. Giving loose permission will create loop holes in the system. Data format should be provided correctly at development time In IOS same permission should also be considered carefully. Memory Protection is required on a system to prevent one App escalating the privilege to write in the address space of another App memory area. This will cause to corrupt the address space of another app running in the system also in some cases will insert malicious binary data which in turn will execute by the app in which the code is inserted.

With the existence of ASLR developers also must follow defensive programming techniques like preventing buffer overflow and other memory corruption that may occur this should be taken care by both MOS.

In Android OS memory randomization is fully applied to Jelly Bean release. In IOS, memory randomization has been applied since IOS 4.3 which is earlier then the Android operating system. IOS has added more secured technology where IOS has code signing technology which is a process required to allow unauthorized applications running in a device.

In Android, the data can be stored in external storage and built in internal storage. External storage SD card does not have permission by default in Android. All the Apps has read permissions to the storage so only read the files. Android also implements standard crypto libraries to secure the storage. But this method is a password policy only. The root kit malware can have easy access to the unwanted code to find the encryption keys in the memory. An App can access all the files throughout the device without any restriction thus may spread the malware directly to the storage.

In the case of IOS, the devices does not have external storage or memory. Only built in storage is provided in IOS. So to access or manipulate App requires permission to the data. The DPA (Data Protection API) API s built into IOS core. It is combined with a complex pass code which provide more security of data protection in the OS. So if we compare Android and IOS storage is more in IOS then in Android and thus makes the App difficult to access the data in the data storage.

III. SECURITY AT PUBLISHING APP IN APP STORE

Publishing means that once our application is completely finished and tested. Now is the time to distribute it to the users to use it. The most common ways to publish an app on Android or IOS is using their corresponding App store like Google Play and Apple store.

Before publishing every App must be signed. Signing is a process in which App is assigned a key. This is done with asymmetric encryption algorithm. There are two types of keys one is Debug and one is Release.

The process of publishing an Android App on the Google Play store, the developer needs access to the Developer Console. The developer console is a web based set of tools that allows the developer to publish and monitor their apps. Android has a flaw that Google App store is not a single place where application can be released. User can choose or install an App from a website or email by modifying their mobile settings to install the App from unknown source.

When building a debug version of the app, the signing is done automatically with a debug key. The debug key does not provide any security of verification. Developer Console does not allow developers to upload apps that are signed with them. Instead, the developer is required to create their own key pair and certificate. The certificate has to meet strict requirements[10]:

1. The certificate has to be valid until at least 22 October 2033
2. It has to represent the developer or company that created it
3. It has to be encrypted with either RSA or ElGamal based Digital Signature Algorithm (DSA)[12]

In IOS at developer end Xcode uses signing identity to signup app during the build process. The signing identity issued by Apple contains public and private key pair. The private key is stored in keychain and used by cryptographic functions to generate the signature. The public key identifies developer as the owner of the key pair. The certificate is stored both in keychain on the Mac and in the developer account.

1. Developer can obtain two types of certificates from Apple using the developer portal: Developer ID certificates (for public distribution) and distribution certificates (for submitting to the Mac App Store).

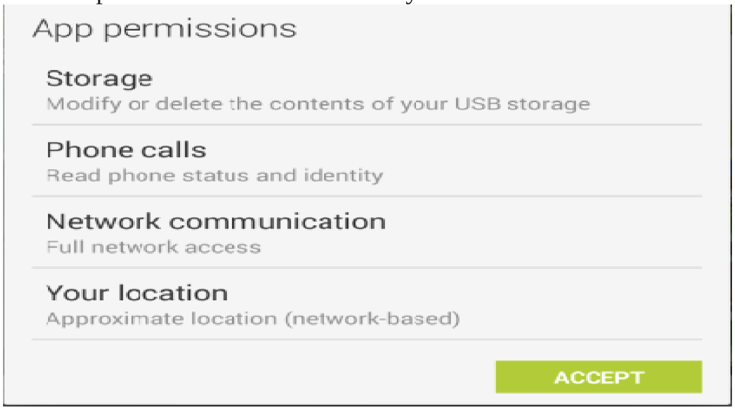
2. The certificate contains An intermediate certificate this is required to be in your keychain to ensure that your certificate is issued by a certificate authority.

IV. SECURITY AT INSTALLATION TIME

At install time user can follow little carefulness to avoid unnecessary problems. If you're careful and you do a little research then you can avoid inadvertently downloading something harmful.

The Android OS has permissions system for individual apps, same for IOS. While installing Android will give us single prompt when we install an App. But IOS allows us to make more decisions.

In Android when we install an App from Android App store or from third party source. Android will display list of resources or permission require by the App to run. So user at install time only declares that the list of permission App required to be given or not. Permission may be for Internet access, Reading USB storage or GPS location etc. So user has only two choices either give all the permissions App require or not to install the App at all. But it is up to the user that he pay attention to the permission while installing and give the permissions or simply deny it to not install the App. But normally users overlook this list of permission which is a security breach.



App permissions

Storage
Modify or delete the contents of your USB storage

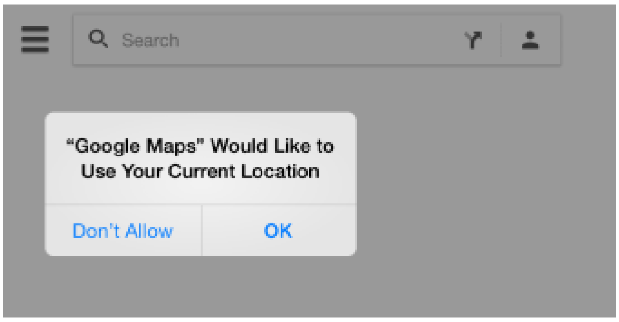
Phone calls
Read phone status and identity

Network communication
Full network access

Your location
Approximate location (network-based)

Fig. 1. Example of permission asked in Android at install time

App permissions in IOS function differently. When installing an App, we will not be asked for any permissions. Certain basic permissions are given to every app however we can modify these basics permissions in IOS for the App we might install. The basic permission might include access to the Internet. So at install time, we are just installing the App but not granting it any special permissions like access to your GPS or contacts list etc So then when user has control of these permissions? Actually these permissions user is asked at the time when the App is actually running and require to use the specific resource. For example , when we install Google Maps or another mapping app, it will prompt with a alert dialog as shown in fig-2 which is asking to use your current location when you first use its mapping features. Same way other Apps also prompt with permission alert dialogs[20]. So here we have more option like one to deny this permission and continue using app anyways.



"Google Maps" Would Like to Use Your Current Location

Fig. 1. Example of permission asked in IOS at execution time

So if we compare Android with IOS Android forces apps to declare more permissions. Android also offers permissions at install time provide seamless run of the App, but on point of security it is venerable to the attack as compare to IOS. The web works like IOS if for example a website wants to access our location, it has to ask. If it wants to access our microphone or webcam, it has to ask. And thus you can choose to deny or allow any of these permissions and still continue using the website.

After installing an app in device, it is verified by the antivirus app or other verification app if user have these apps on his or her device. This makes installation process more secure. So many gadgets-gurus long believed that Android's permission system is an advantage over IOS. But this may be shocking to that IOS is more secure and practical as compare to Android.

V. SECURITY WHILE EXECUTING APP

As we have discussed in above section how android App manage security issues like process isolation , memory protection , usage of resources by providing necessary permission while App is installed by the user. In IOS, this permission based access control achieved at runtime. When an App executes, it requests the use of any one of the protected features such as user's location while using Map App , the OS pops up a dialog box and asks the user is he or she chooses to allow the App access to the demanded resource. Many Apps continue without having this permission and some apps fails if user goes with no option.IOS also provide runtime security through code signing process which attempts to prevent unauthorised App running on the device. Code signing process validates an App signature whenever it is executed and only those apps executes on the device which are signed by a valid and trusted signature.

In IOS, one of the powerful method Hooking the Objective C runtime which includes observing and modifying accordingly the behaviour of an App. IOS achieves this using Mobile Substrate, which provides hooking framework for jailbroken devices.

In android App, developer should build and test security of its app by following android best practice which include whether the App is not performing unsafe files creation, improper data storage, unsafe use of shared preferences, content provider SQL intrusion, inbound SMS listener, APN or proxy modification etc while it executes. Root kit are malwares which actually provide administrator rights to the attacker of the Mobile. If this happens, then nothing prevents the attacker to study or disable the safety features that were avoided, deploy the applications they want, or disseminate a method of intrusion by a Rootkit to a wider audience.[37][38] So It is important to prevent the invasion of a RootKit in the system. If intruded there must be some mechanism to be able to detect them. We can cite, as a defence mechanism, the Chain of trust in iOS. This mechanism relies on the signature of the different Apps required to start the OS, and a certificate signed by Apple. In the event that the signature checks are inconclusive, the device detects this and stops the boot-up.[39] If the Operating System is compromised due to Jail breaking, root kit detection may not work if it is disabled by the Jailbreak method or software is loaded after Jailbreak disables Rootkit Detection.

VI. CONCLUSION

Below is the comparison between IOS and Android based Apps based on various App phase security.

TABLE I

COMPARISON BETWEEN ANDROID AND IOS

Phases	**Android**	**Ios**
Developmen	Application sandboxing in Android is controlled by each application and required permission and approval to continue accessing what the application needed.	In IOS application sandboxing is a set of fine-grained control that limits the application access to the file system, network and hardware.IOS has a robust sandbox model
	Each app has its own sandbox. This improves security tighter	Shares a same sandbox model which is more secure and less open to the crowd.
	Uses file system encryption only	Uses file system encryption and hardware encryption
Publish	No code signing.	Has code signing technology which is a process required to allow unauthorized applications running in a device.
	Account on developer console and App should have a valid certificate.	Account on App store and App must be code signed.
Installation	All types of permission are assigned to the App at installation time.	Minimal set of permission are automatically assigned during installation of App.
	User denies the permission then App installation will be aborted.	User will not be asked for any permission. No deny option is there.
Execution	Seamless execution of App as no permission interruption are there	The user will be asked for permission during execution time when the App actually uses the resource
	Less secure as user will not be aware that the App may be using that resource which it is not intended for	More secure as user will be notified if any resource App want to use

security in mobile devices are not a sole responsibility of developer. It is the responsibility of all the users and people participate from the development phase to end user phase of execution. It also depends on how the App is used on the device as well as App stored on the App stores. As they must allow only secured App to be published to the users, by verifying its signatures and certificates. In conclusion both MOS, Android and IOS provides very good security. Provide a very good App store where users can publish their Apps. But both of them have their pros and cons.

REFERENCES

[1] https://viaforensics.com/resources/reports/best-practices-ios-android-secure-mobile-development/

[2] https://developer.apple.com/library/mac/documentation/security/conceptual/Security_Overview/Introduction/

Introduction.html#//apple_ref/doc/uid/TP30000976

[3] http://www.digitaltrends.com/mobile/android-app-security-basics/#ixzz33JTbRJ3U

[4] Android Open Source Project. "Security Overview." Tech Info. N.p., 2012. Web. 18 June 2012. http://source.android.com/tech/security/index.html.

[5] Bilton, Nick. "Apple Loophole Gives Developers Access to Photos." Bits. The New York Times, 28 Feb. 2012. Web. 2 July 2012. http://bits.blogs.nytimes.com/2012/02/28/tk-ios-gives-developers-access-to-photos-videos-location/.

[6] Sacco, Al. "Six Essential Apple iPhone Security Tips." Business Center. PC World, 12 Oct. 2008. Web. 2 July 2012. http://www.pcworld.com/businesscenter/article/152128/six_essential_apple...

[7] Veracode. "Mobile Security for the Rest of Us." 11 June 2012. PDF eBook file.

[8] Google. Permissions. http://developer.android.com/guide/topics/security/permissions.html, June 2013.

[9] Google. Manifest.permission. http://developer.android.com/reference/android/Manifest.permission.html, June 2013.

[10] Google. Signing in release mode. http://developer.android.com/tools/publishing/app-signing.html#releasemode, June 2013.

[11] Google. Google play developer console. https://play.google.com/apps/publish/signup/, June 2013.

[12] D. W. Kravitz. Digital signature algorithm, July 27 1993. US Patent 5,231,668.

[13] K. Casteel, O. Derby, and D. Wilson. Exploiting common intent vulnerabilities in android applications. http://css.csail.mit.edu/6.858/2012/projects/ocderby-dennisw-kcasteel.pdf, December 2012. *10. 2013 8th International Conference on Information Technology in Asia (CITA)*

[14] Qing Li; Clark, G., "Mobile Security: A Look Ahead," *Security & Privacy, IEEE* , vol.11, no.1, pp.78,81, Jan.-Feb. 2013 doi: 10.1109/MSP.2013.15

[15] Khadijah Wan Mohd Ghazali, Rosilah Hassan and Zulkarnain Md Ali, A Network Device Simulator, IEEE ICACT 2013, PyongChang Korea Jan 27-30, 2013, pp.378-381. [13] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith, "*Smudge attacks on smartphone touch screens*," Berkeley, CA, USA, 2010, pp. 1- 7.

[16] Prince McLean.Inside google’s Android and Apple’s iPhone OS as business models.roughlyDrafted Magazine.November 10,2009

[17] Apple Computer Corp., Mac OS X has you covered, Apple – Mac OS x – Security – Keeps safe from viruses and malware.pdf, http://www.apple.com/macosx/security

[18] http://www.howtogeek.com/177711/ios-has-app-permissions-too-and-theyre-arguably-better-than-androids/

[19] Peijnenburg Falco , Security in Android apps, august 16, 2013

[20] http://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Maintaining Certificates/MaintainingCertificates.html

[21] Mobile Security in android and IOS: http://blog.veracode.com/2012/01/mobile-security-android-vs-ios/

[23] Number of android apps in market: http://www.appbrain.com/stats/

[24] http://148apps.biz/app-store-metrics/

[25] Shah, Kunjan. "Top 10 iPhone Security Tips." Top 10 iPhone Security Tips. McAfee, 2011. Web. 2 July 2012. http://www.mcafee.com/us/resources/white-papers/foundstone/wp-top-10-iph...