Analysis of Android Security with the Help of Component Granting Permission

Gajanan V. Jaybhaye
Department of CSE, Government College of Engineering, Amravati, India.

Pushpanjali M. Chouragade
Department of CSE, Government College of Engineering, Amravati, India.

ABSTRACT
Nowadays, the mobile has operating system for which we required security. It works like operating system of personal computer, generally mobile uses android base operating sytem. will granting a permission break the phone's security? Do the access permission assignments to an application's components put the phone or the application at risk? Android currently provides no means of answering these questions. We developed an enhanced installer and security framework to answer a variant of these questions—namely," does an application break some larger phone-wide security policy?" Our tool, called Kirin extracts an application's security policy from its manifest file to determine if the requested permissions and component permission assignments are consistent with the stakeholders' definition of a secure phone. The existing reviews extensively cover the smart phone OS security. However, we believe that the security of Android, with particular focus on malware growth, study of anti-analysis technique and existing detection methodologies needs an extensive coverage.

Keywords: component label, API, SDK, APP

1. INTRODUCTION
Android is an open source mobile operating system developed based on Linux system and introduced by Google and its open handset alliance.[1] Currently, Android has been widely used in mobile phones, tablet PCs, laptops and other smart mobile devices. According to the data report released by market research firm Canalys on May 30, in the first quarter of 2013, the global smart mobile equipment shipments were 308.7 million, up 37.4% from a year earlier. Android operating system of Google accounts for 59.5% of the share in the first quarter of the global smart mobile devices market; Followed by Apple's IOS, 19.3%; Microsoft Windows and Windows Phone is 18.1% .[2] This shows that the Android dominates the mobile market. It's essential that this next generation of platforms provides a comprehensive and usable security infrastructure. Developed by the Open Handset Alliance (visibly led by Google), Android is a widely anticipated open source operating system for mobile devices that provides a base operating system, an application middleware layer, a Java software development kit (SDK), and a collection of system applications. One of Android's chief selling points is that it lets developers seamlessly extend online services to phones. The most visible example of this feature is, unsurprisingly, the tight integration of Google's Gmail, Calendar, and Contacts Web applications with system utilities. Android users simply supply a username and password, and their phones automatically synchronize with Google services. Other vendors are rapidly adapting their existing instant messaging, social networks, and gaming services to Android, and many enterprises are looking for ways to integrate their own internal operations (such as inventory management, purchasing, receiving, and so forth) into it as well.

2. ANDROID SECURITY MECHANISMS
Android system supports multi-platform operation, which uses the version of the kernel of Linux 2.6, and uses Dalvik virtual machine as an APP runtime environment. Android system has a layered architecture[4]. Form the bottom to the top, there are five layers, which are the Linux kernel, the local library, the Android runtime environment, the APP framework and the APP. During designing and developing the Android operating system, Google not only inherits the designing idea of Linux, but also sets a corresponding security mechanism in each layer. Moreover, Google also sets two kinds of Android specific security mechanisms: signature and APP permission control.

2.1 Signature Mechanism
All Android APPs must have a digital certificate, due to that the system will not install an APP that doesn't have a digital certificate. Unlike other platforms, Android APP signature not only indicates the publisher of the APK, but also provides validation of the integrity and reliability of the program. For those who attempt to tamper with the APK file, the system will force them to re-sign the APK. Under the condition that the author's signature private key does not leak, the fake signature is almost impossible exactly the same as that of the original signature which has uniqueness. Signature mechanism plays a protective role in the APP update. Only under the circumstance that the two signatures are exactly the same, system allows the update operation. Otherwise the system will prohibit this update to further protects the security of the system.

2.2 App Permission Control Mechanism
Permission control is the key of Android APPs security mechanism. Android deals with security problems by means of implementation of security policies based on permission control, i.e., using permission control to restrict the APP installation, so that the APP can only access API and resources within the permission. Android defines 135 kinds of the system permissions which are divided into four protection levels[5], which are normal, dangerous, signature and signature or System, respectively. All the permissions and related functions can be seen in the development document of the Android system [6]. By default, Android APPs don't have any permissions. Permissions involved in the APP runtime need to be declared in the label of uses-permission in the AndroidManifest.xml of APK file. At the time of installation, Android APP package manager will prompt the user of the application of the APP permissions only with the

authorization of the user, the installation can begin, otherwise, installation will be cancelled. After successful installation, the system will answer the requests for program to access resources according to the solidification permission information when APP runs. If there are corresponding permissions, access successes; otherwise the APP will be forced to shut down by system.

3. COMPONENT TYPES

Android defines four component types:

Activity - components define an application's user interface. Typically, an application developer defines one activity per "screen." Activities start each other, possibly passing and returning values. Only one activity on the system has keyboard and processing focus at a time; all others are suspended.

Service - components perform background processing. When an activity needs to perform some operation that must continue after the user interface disappears (such as download a file or play music), it commonly starts a service specifically designed for that action. The developer can also use services as application- specific daemons, possibly starting on boot. Services often define an interface for Remote Procedure Call (RPC) that other system components can use to send commands and retrieve data, as well as register callbacks.

Content provider- components store and share data using a relational database interface.Each content provider has an associated "authority" describing the content it contains.Other components use the authority name as a handle to perform SQL queries (such asSELECT, INSERT, or DELETE) to read and write content. Although content providerstypically store values in database records, data retrieval is implementation specific forexample, files are also shared through content provider interfaces.

Broadcast receiver- components act as mailboxes for messages from other applications. Commonly, application code broadcasts messages to an implicit destination. Broadcast receivers thus subscribe to such destinations to receive the messages sent to it. Application code can also address a broadcast receiver explicitly by including the namespace assigned to its containing application.

4. COMPONENT INTERACTION

The primary mechanism for component interaction is an *intent*, which is simply a message object containing a destination component address and data. The Android API defines methods that accept intents and uses that information to start activities (startActivity(Intent)), start services (startService (Intent)), and broadcast messages (sendBroadcast(Intent)). The invocation of these methods tells the Android framework to begin executing code in the target application. This process of intercomponent communication is known as an *action.* Simply put, an intent object defines the "intent" to perform an "action."

5. CONCLUSIONS

In many ways, Android provides more comprehensive security than other mobile phoneplatforms. However, learning how to effectively use its building blocks isn't easy. We'reonly beginning to see different types of applications, and as Android matures, we'll learnhow faulty application policy affects the phone's security. We believe that tools such asKirin and those like it will help mold Android into the secure operating system neededfor next-generation computing platform

6. REFERENCES

[1] J.P. Anderson, 1. *Computer Security Technology Planning Study*, tech. report ESD-TR-73-51, Mitre, Oct. 1972.

[2] M.A. Harrison, W.L. Ruzzo, and. J.D. Ullman, "Protection in Operating Systems,"*Comm. ACM*, vol. 19, no. 8, 1976, pp. 461-471.

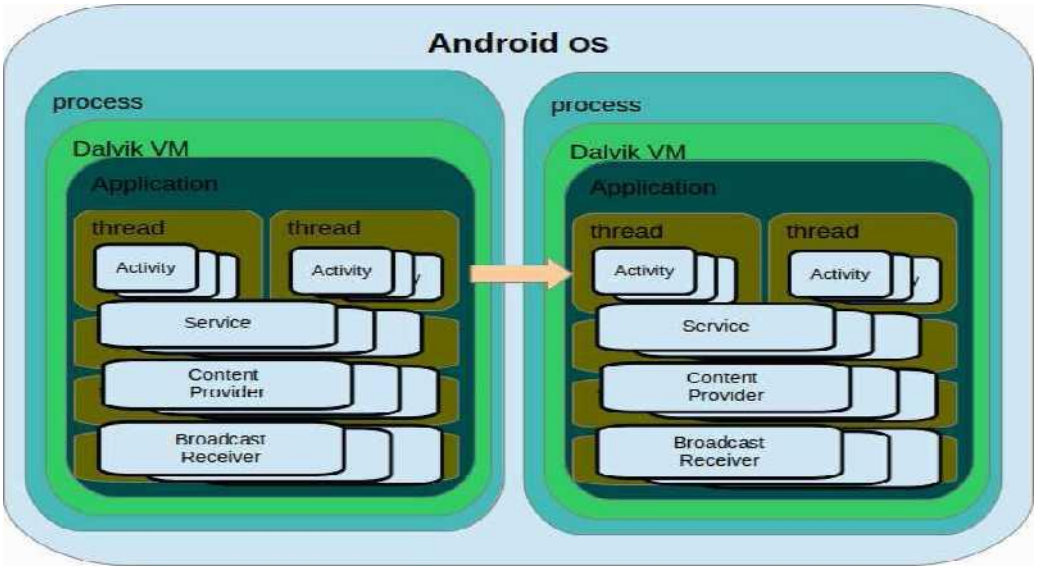

Figure 1: Different Functional Components of Android OS.

[3] L. Badger et al., "Practical Do- 3.main and Type inforcement for UNIX," *Proc. IEEESymp. Security and Privacy*, IEEE CS Press, 1995, pp. 66-77.

[4] J. Saltzer and M. Schroeder, "The 4. Protection of Information in Computer Systems,"*Proc. IEEE*, vol. 63, no. 9, 1975, pp. 1278-1308.

[5] I. Krstic and S.L. Garfinkel, "Bit- 5. frost: The One Laptop per Child Security Model,"

[6] Yan Jin,ShanglangYao.Introduction and practice of Google Androiddevelopment[M].Beijing:Posts&Telecom Press, July 2009,pp.12-18.

[7] Google.AndroidReference:Manifest File-Permissions. http://developer.android.com/guide/topics/manifest/manifest-intro.html[EB/OL].

[8] Google.AndroidReference:Security and Permissions. http://developer.android.com/guide/topics/security/security.html[EB/OL].

7. AUTHOR BIOGRAPHY

Gajanan V. Jaybhaye has received his B.TECH. degree in InformationTechnology from Government College of Engineering, Amravati, India in2012. His research interest includes Android, Data mining, Webmining.At present, Hi is persuing Master of Technology in department ofComputer Science and Engineering at Government College of Engineering, amravati, India.

Pushpanjali M. Chouragade has received her Diploma in Computer Scienceand Engineering from Government Polytechnic, Amravati, India, in 2007,the B.Tech. degree in Computer Science and Engineering from GovernmentCollege of Engineering, Amravati, India in 2010 and her M.Tech. in Computer Science and Engineering from Government College of Engineering, Amravati,India, in 2013. She was a Lecturer with Department of Computer Science & Engineering, in Government College of Engineering, Amravati, in 2010-11. Her research interestincludes Data Mining, Web Mining, Image Processing. At present, she is an Assistant professor with department of Computer Science and Engineering at Government College of Engineering, Amravati, India, since 2011.