A New Secure Android Model Based on Privilege

Tao Zhang(✉) and Zhilong Wang

College of Information Science and Engineering, Ocean University of China,
No. 238 Songling Road, Qingdao, Shandong, China
oucseczt@l26.com, zhilongwang@foxmail.com

Abstract. Android is the most popular smartphone operating system in the world. There are many people who focus on Android security and dedicated to improve android security. There are many android vulnerabilities exposed online and attackers can use these vulnerabilities to steal our private and sensitive information and attack our device. In the paper, we propose a novel secure android model based on privilege. We create three kinds of users and grant different users different permissions. Thus, users can have more freedom and control over their android device in the model we proposed. In our secure model, users can upgrade their android operating system in time, which may enhance the security of their smartphones and protect users' sensitive information better.

Keywords: Android · Security model · Root · Privilege

1 Introduction

1.1 Android Introduction

Led by Google, Android is an open source operating system which based on the Linux kernel. Android is the first open and free mobile operating system that is used widely on the mobile devices.

Google is trying to ensure that Android is a safe and open ecosystem for users and developers [1]. Android is the most popular smartphone operating system in the world. There are about 81.6 percent android device and 15.9 percent iOS devices in the worldwide smartphone operating market share in 2015 [2]. There are more than 1.4 million apps in total dating back to Dec 31, 2015. According to Yahoo Flurry, there is a 14 percent increase in the android apps, and a 332 % boost in some special area. Users can download and install applications from Google play and other third party application markets. However, there are some malicious applications and applications with vulnerabilities in these application markets.

1.2 Android Security Status

As more and more people use smart phones, phones are not just phones. We can use smart phones to chat with others by WeChat, MSN and other applications, to record our

J.J. (Jong Hyuk) Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*, Lecture Notes in Electrical Engineering 421, DOI 10.1007/978-981-10-3023-9_29

life by taking photos and to do shopping online. Smartphones store too much sensitive information and personal information, such as contacts, SMS, bank accounts, payment information, photos, GPS information. In a word, security for smartphone is becoming an issue [3].

Android is open source and the AOSP (Android Open Source Project) get broad security review by many interested parties, including hackers, manufacturers and other interested parties. There are many android vulnerabilities exposed online in the past years and attackers can use these vulnerabilities to steal our private and sensitive information, even to attack our android devices. Of these vulnerabilities, some are fatal and can bring great damages to devices and the owner. Take stagefright as example, CVE-2015-1538 [4], it can be abused to control our mobile phones by privilege escalation.

There are many different android versions customized by different customers and it takes months for users to get access to the version of system security upgrade or update. What is worse, there are too many users who have no idea to update their android operating system.

2 Android Security Mechanisms

There are many security mechanisms in android to ensure the security of the most popular mobile operating system, such as permission model, sandboxing and isolation.

2.1 Permission Model

Permission model is an application level security mechanism of android. There is a *AndroidMainfest.xml* file that describes the permission needed during execution in a application's APK package. One can set *uses-permission* to request permission when developing an application.

At runtime, services at this layer enforce the permissions specified in the manifest and granted by the user during installation [5].

Permission and permission enforcement make sure that an application's access to resource is in accord with the description in *AndroidMainfest.xml* file.

2.2 Sandboxing and Isolation

Sandboxing and isolation are kernel level security mechanism of android. All android devices share a common security model that provides every application with a secure, isolated environment known as an application sandbox [1]. Android security model is partly based on application sandboxes. Each application is executed in a separate Dalvik VM machine, which ensures isolation among applications [5].

2.3 Security Enhanced Android

SELinux, means Security Enhancements for Linux, is mandatory access control for Linux, which can confine flawed and malicious applications and prevent privilege escalation.

SE android [6], Security Enhancements for android, is available in Android since version 4.4.3, which enforce SELinux in android to mitigate malicious applications' threat and to prevent privilege escalation, data leakage by applications and bypass of security features.

2.4 Root

Root is the privilege user of android and Linux operating system. Actually, we don't have full control over our android devices [7]. A user or application with root permissions can modify the operating system, kernel files and any other applications' permissions. If we have full control over the device, we can change system settings, uninstall system applications, delete system files, backup system application data, install applications at SD card and so on.

3 Android Security Framework Analysis

3.1 Permission Model

Although many security mechanisms applied in the android operating system, both on Linux kernel layer and application layer. There are many vulnerabilities in android which may be abused and make our private data in danger. Android enforces strict security mechanisms to limit code execution of vulnerabilities, which may introduce new vulnerabilities at the same time. In android, some malicious applications can bypass those security mechanisms by exploiting some vulnerabilities [8].

However, the permission model is all or nothing [9]. Actually, the user need to accept all permission the application required or the application can not be installed successfully. Once successfully installed, the application will permanently have access to the permission described in the *AndroidMainfest.xml* file. However, for example, if an application requires the GPS permission when installed, it can have access to GPS information whenever. Users do not have flexible control over the permission the application required and their sensitive information, such as GPS information.

Permission and permission enforcement make sure that a application's access to resource is in accord with the description in *AndroidMainfest.xml* file. However, the permission model is not as secure as we think. In [10], the author modify the *AndroidMainfest.xml* file, then reboot the smartphone and the application gets the permission he modified before.

3.2 SE Android

There are also some limits in SE android. SE android is SE-Linux's migration and extension in android, which focus on enforcement access to system call and kernel. SE-android provide solution to malicious applications' abuse system call to invoke kernel function in android [5]. But there is no protection on user data which may be stolen and modified [11] and there is not existing plans to mitigate common vulnerabilities [10].

3.3 Root

As a privilege user, root has full access to all applications and all data on the device. Basebridge malware can use HTTP protocol to communicate with central server and convey our private information. DroidKongFu can collect IMEI, device ID, SDK versions and other information which will be sent to a specific remote server [12]. Androidkungfu use *NPROC_RLIMIT* vulnerability and FramaRoot that were reported before to acquire root privilege, then control the victims' smartphones [8]. Root access gained by exploiting a kernel bug or security hole can bypass bootloader unlock mechanism protection. Once malicious applications or applications with vulnerabilities run with root privilege, all your information on the devices will never be yours.

Most of these vulnerabilities exploit vulnerabilities to gain root access or escalate privilege, which are among the largest threats to android operating system.

4 A New Android Security Framework

4.1 Root-Admin-User

There should be a balance between the security of android and users' flexible control over smartphones. Android users want to get more control over their android devices, so some of them try to root their devices and there are many kinds of rooting tools available. However, if we can not protect the root privilege properly, which may cause great damage to android devices and their owners. And there are many malicious applications which use vulnerabilities to get root privilege or escalate privilege.

Like Linux user model, we propose a new security model and divide users into 3 kinds: root, admin and user.

Root: In our model, when users get their android device, they should not get root privilege again unless they format or upgrade their smartphones. We will grant most of the root privilege to the admin and normal user, for example, we give normal user the privilege to change fonts of their android operating system. To assure the security of the android operating system, some of the key system files are not allowed to be changed in our model, such as the *.lib* file.

Admin: We assign the admin user most of the privileges which are belonged to root before. The android operating system run as normal user initially. If normal user need to get root privilege, it will switch to get admin privilege in our model.

User: User is the ordinary android user with no privilege in our model. However, we give owner of the android device more control over the device as a normal user. For example, the owner of the device can uninstall the pre-installed applications.

4.2 Classify System File and Rearrange Privilege

We are to give users more freedom to control and set their android devices. What's more, we provide a more secure and privacy-protected framework of android, so we can protect user data and privacy better on these android devices.

We divide system files into 3 parts based on privilege, then enforce the access control mechanism.

The 1st part is the key files that should not be changed. The 2nd part is the files that can be changed by users freely, and the 3rd is the rest files. We give these 3 kinds of files different privileges, the 1st kind of files is owned by root, which should not be changed in the security model we proposed. The 3rd is owned by admin that can be changed intentionally and users can access and change these files only in some condition. The 2nd kind of files is those files owned by user and the owner of the device can change these files freely, such as fonts, background.

4.3 Upgrade Mechanism

Android is open source and allow device manufacturers to customize devices and introduce diversity. In the android security ecosystem, Google is far away from end users [1], because the android system running on our devices is accustomed by device makers and we can not improve our android's security although Google released the security improvements.

Android is an open operating system and openness will strengthen security [1]. All interested parties can join to improve android's security levels. But there should be some organizations like Google or android union who be responsible for the upgrade of the core system.

The upgrade of operating system is responsible for different parties, such as device manufacturers, vendors and an organization like Google and Android Security Unions. End users can upgrade their android directly after the organization released security upgrade, even their device customers did not provide such security upgrade.

The device customers' upgrade is about the customized part of system and this kinds of upgrade does not have an effect on the upgrade of kernel.

5 Conclusions

In the paper, we propose a more secure and privacy-protected android model which changes the permission model so that users can have better control over their android devices and make sure their data and private information get better protection. We create three kinds of users and grant different users different privilege. In our secure model, users can upgrade their android operating system in time, which will enhance the security of their smartphones and privacy.

References

1. Google. Google android security 2015 report. http://static.googleusercontent.com/media/source.android.com/en//security/reports/GoogleAndroid. Security 2015 Report Final.pdf
2. Statista. Android os market share of smartphone sales to end users from 2009 to 2015. http://www.statista.com/statistics/216420/
3. Han, K.S., Lee, Y., Jiang, B., Im, E.G.: Android permission system violation: case study and renement. Int. J. E-Entrepreneurship Innov. (IJEEI) **4**(1), 16 (2013)
4. The Latest on Stagefright: CVE-2015-1538 Exploit is Now Available for Testing Purposes. https://blog.zimperium.com/the-latest-on-stagefright-cve-2015-1538-exploit-is-now-available-for-testing-purposes/
5. Merlo, A., Costa, G., Verderame, L., et al.: Android vs. SEAndroid: an empirical assessment. Pervasive Mob. Comput. **30**, 113–131 (2016)
6. Smalley, S.: The Case for SE Android. National Security Agency (2011)
7. Zhang, H., She, D., Qian, Z.: Android root and its providers: a double-edged sword. In: ACM Sigsac Conference on Computer and Communications Security, pp. 1093–1104. ACM (2015)
8. University of Chinese Academy of Sciences National Computer Network Emergency Response Technical Team Coordination Center of China State Key Laboratory of Integrated Services Networks Xidian University Beijing Electronic Science and Technology Institute. Survey of Android Vulnerability Detection. J. Comput. Res. Dev. (2015)
9. Nauman, M., Khan, S., Zhang, X.: Apex: extending android permission model and enforcement with user-defined runtime constraints. In: ACM Symposium on Information, Computer and Communications Security, ASIACCS 2010, Beijing, China, April 2010
10. Yang, C., et al.: Utilization pattern based android root vulnerability analysis. Comput. Sci. **41**, 343–346 (2014)
11. Xiang, W.H., Yu-Jun, L.I., Hou, M.S.: Research on privacy protection based on SEAndroid. Comput. Sci. **42**, 329–332 (2015)
12. Zhou, Y., Jiang, X.: Dissecting Android malware: characterization and evolution. In: IEEE Symposium on Security & Privacy. IEEE (2012)