International Journal of Security and Its Applications
Vol. 9, No. 3 (2015), pp. 7-12
http://dx.doi.org/10.14257/ijsia.2015.9.3.02

Studying Security Weaknesses of Android System

Jae-Kyung Park* and Sang-Yong Choi**

**Chief researcher at Cyber Security Research Center, Korea Advanced Institute of Science and Technology,*
Seoul 135-854, Korea
E-mail: wildcur@kaist.ac.kr
***Chief researcher at Cyber Security Research Center, Korea Advanced Institute of Science and Technology,*
Daejeon 305-701, Korea
E-mail:csyong95@kaist.ac.kr

Abstract

As smartphones are generalized, various technologies and services have been introduced and are in wide use. From simply using calling or texting services, Internet banking and transaction system that require sensitive personal information emerged. Google's Android, one of the representative OS of smartphones, was developed based on an open source, having various weaknesses and exposed to security threats. In this paper, we study the types and characteristics of these weaknesses as well as the risk elements, introducing a safer usage of smartphones.

Keywords: *Android, Smartphone, Malware, Security, Weaknesses*

1. Introduction

Usage of smartphones has been generalized and utilization of it is not limited to calling and texting services. Due to development of smartphones, various tasks such as emails, internet, e-book, Internet banking, and payment can be processed. And now, the usage of smartphones has become a daily life, widening its area to all fields of life. However, in the midst of consistently growing and developing new functions, the security issues to protect user information are not keeping up with the pace of development. Although various security technologies are researched and developed to grow consistently, there are still many threats. Therefore, we intend to find resolutions by researching and analyzing present weaknesses.

This paper is organized as follows: the weaknesses of Android system are presented in Chapter 2, and we introduced the current countermeasures against weaknesses of Android system in Chapter 3. In Chapter 4, attack scenario using vulnerable points and response measures are suggested. Lastly, we conclude the paper in chapter 5.

2. Security Weaknesses of Android System

2.1 Inadequate Management of Authority

2.1.1 Inadequate Management of User Authority

Android shows the authority information required by the system when installing application. User can check the permission required by the application and select whether to

ISSN: 1738-9976 IJSIA

install or not. However, there is a difficulty for general users to check all permission during installation. This will eventually lead to user responsibility in case of a problem [7].

As this authority information can be checked only during installation, this can be a security threat for those without knowledge or interest in the concept of authority. Malicious applications can make use of the address book, SMS and mobile phone information using the authority information. Also, it can manipulate or delete location information and personal information or even extort password by phishing. For example, a text message of “Free Coupon Gift”, falsely representing a mobile coupon event, was widely distributed. This text message was distributed to random smartphone users without gaining authorization in the similar way, and included shortened URL at the end. By clicking on this URL address, the “tj.apk” file is downloaded in the download folder of the user. This file can be installed by clicking on it, and the installation authorization screen for the malicious application will appear. This authorization includes services that can charge the user. If the user doesn’t carefully read the installation authorization, the corresponding application will perform micropayment fraud upon completion of the installation [8].

2.1.2 Acquisition of Authority to Share User ID

Android supports “shared user ID” to allow mutual interlocking of applications. “Shared user ID” refers to the technique where different applications share an ID to gain access to each other’s files and processes. By sharing “shared user ID”, data and authorities are also shared. Using this weakness, malicious applications can gain access to unauthorized resource by sharing the ID. For example, although the malicious application doesn’t have the authority for location information, if it can abuse the API of an application that can access the location information, it can rob of the location information of the user. Another example corresponds to cooperative attack between applications. In this case, two or more applications are written by using personal information leakage application and external leakage application. This method cooperatively attacks through the IPC of two applications. In this case, it is hard to detect the basic authorization as excessive authorization request is not necessary. Also, malicious application can attempt malicious attacks using the same signature as a normal application to share “shared user ID” and its authorities [9].

2.2 Attacks by Capturing Android Authority Through Rooting

Rooting refers to a method of acquiring root authority of smartphones using the weaknesses of the OS. Although it is used by user to control all parts of the system, if maliciously used, its damages include personal information leakage and charging, and it can result in serious damage to the system. Also, by installing a boot, the system can be abused in DDos attack. One of the methods for acquiring administrator authority from Android terminal is GingerBreak rootkit. This method obtains administrator authority through message hooking method operated by the init process of the Linux shell. GingerBreak performs a permanent rooting by copying the falsified su file to the system. Malicious codes integrated with GingerBreak are distributed through 3rd party market and data including personal information and phone numbers are leaked to a specific server.

2.3 Weakness of Master Key

The weakness of master key is related to not precisely checking the encrypted signature. If there is an entry file with the same name in the APK file, the attacker can create an arbitrary file that doesn’t violate the encrypted signature. HashMap has a weakness of overwriting the relevant value by not permitting duplication of files. In this case, APK file can be used without signature.

3. Android Security OS Countermeasure

Android is a Linux based platform that has a unique security mechanism modified to suit mobile environment. As it is a Linux based OS, preemptive multitasking, effective shared memory, user ID and group ID of Unix, and file accessing authority are inherited for use. Also, in Figure 1, by using the characteristic of Java, it also features security functions such as the sandboxing [4].

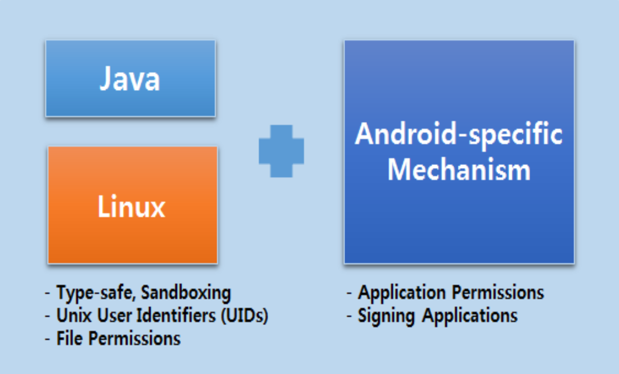
Java
Linux
Android-specific Mechanism
- Type-safe, Sandboxing
- Unix User Identifiers (UIDs)
- File Permissions
- Application Permissions
- Signing Applications

Figure 1. Android Security Model

3.1 Android Security Module Framework

Android security module framework allows developers and users to use new security tools without a separate administrator authority, and allows 3rd party application developers to provide the latest security function of their applications real time. Latest security technology can be utilized without waiting for regular updates by Android. Latest security functions can be quickly applied by deviating from a system that depends on regular updates [5].

3.2 KNOX, Samsung

KNOX is a security software launched by Samsung Electronics featuring mobile platform focusing on security. Currently, it is optimized by Google Android 4.3 Jelly Bean OS and higher versions. By installing virtually fortified security space in a single mobile device, it can be used for personal use and business use. Although similar to privacy locking function, Knox fortified its security by encrypting the security space. In other words, Knox container, the security space and general user space cannot transmit and receive data in simple way. When connected to USB cable, only charging function is available. By using this technology, exposure to hacking and virus is blocked. Therefore, there are also areas for improvement such as complicated replacement of kernel. Of course, there have been custom Android security solutions for business use before KNOX. However, it is meaningful that KNOX expanded its area to general products [1, 6].

3.3 System Protection Technique

By improving Android permission, a segmented permission model was suggested. However, these research present disadvantages of being hard to apply as the Android framework needs to be modified. To complement this, a technique that can apply new permission system at the application level was introduced. Also, analyzing permissions used by applications and techniques that can estimate the possibilities of malicious application were suggested. Although these filtering based on permissions have advantages of being light and quick, there is still a disadvantage of low accuracy in identifying malicious application and normal application [2, 3].

4. Android Security Threat Scenario

Android OS manufactured on the basis of Linux uses a user authority system the same as Linux, while root means the authority acquisition at Linux which is used uniformly even in Android OS. The relevant application hides the code that forcibly routes the terminal. When it is infected, the mobile phone specific information and other malicious programs can be uploaded onto an infected device, and it's possible to steal product ID, detailed information of a mobile phone model, and the data like the information and language of a communications company as well in Figure 2.

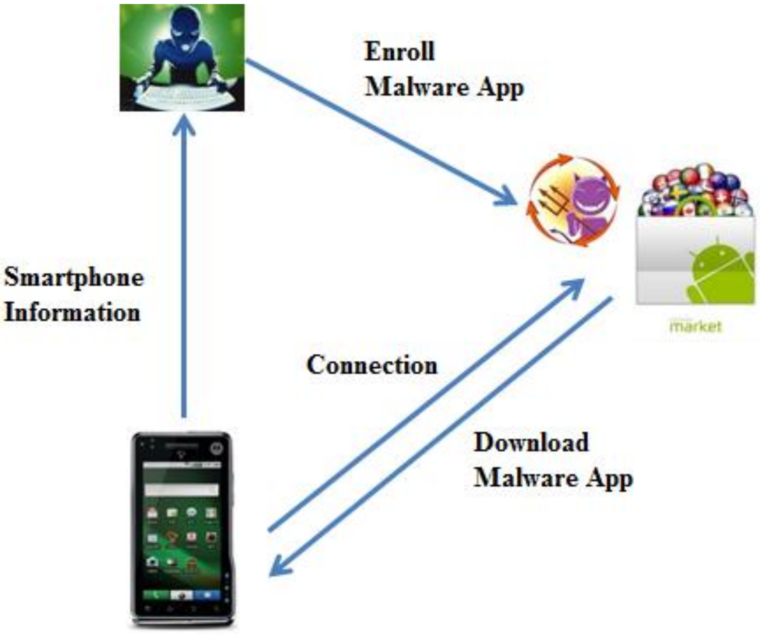

Figure 2. Attack Scenario

4.1 Black Market Registration & Installation Guidance Phase

① An attacker registers a charged altered malicious application, which is available free of charge, at the black market.

▶ The altered malicious application is the rootkit using Kernel vulnerable point Futex(Fast Userspace Mutex) and Bug(CVE-2014-3153).

② An attacker puts up black market execution file (.apk) onto famous Blogs and SNS to make it possible for many people to install the altered malicious application.

③ An attacker registers adult films, films, charged games, and vaccine application, etc. that can arouse people's interest to make it possible for a lot more people to download them.

4.2 Download & Installation Phase

① A user downloads the charged application free of charge, or the altered application that arouses interest from the black market for installation.

② In the process of a malicious application's installation at the terminal, confirmation and approval of the terminal H/W resource use authority are required, but most of the users tend to approve without confirmation.

4.3 Dispossession Phase of Terminal Information & Financial Information

① Once a malicious application is installed, a rootkit is installed at a user's terminal, opening a specific Port and sending IP to the first attacker to make it possible for the attacker to access the terminal. In addition, location information is additionally sent to make it possible for the attacker to confirm the location of a user in real time.

② An attacker accesses the relevant user's terminal after confirming IP Number, location information of the terminal, which is installed with a malicious application, from the server of the attacker's own make. (An attacker accesses the terminal using root authority, which could do several attacks and damage on the terminal).

③ An attacker installs a malicious application, which can dispossess financial information using root authority, at a user's terminal.

④ When a user enters the password of the certificate for using Internet banking as usual, the installed malicious application delivers the information to an attacker's server by collecting the certificate password of the relevant terminal, account number needed for an account transfer, and transfer password, etc.

⑤ An attacker dispossesses money by approaching a user's account using the delivered information.

⑥ An attacker increases the number of victims by transmitting URL having the black market installation file to SMS using the phone number stored in a user's terminal in order to do damage to another user.

▶"This is OOO. We are distributing the diagnosis application against Google Store Hacking, so please diagnose your terminal by installing the diagnosis application at your terminal. Application Download http://xxx.xxx.xx/xxx"

4.4 Response Method

We suggest the most basic security method for Android system, as shown below.

① A user is encouraged to update the relevant vulnerable points (CVE-2-14-3153) by the very terminal manufacturer, so the user is supposed to get the manufacturer's OS update.

② A user is forbidden to use the black market, and encouraged to intensify security by installing vaccine that can detect the rootkit.

③ A user is supposed to update the vaccine of the terminal and smart phone in real time.

5. Conclusion

Security weakness of Android system poses threat to safe mobile use of users. In particular, in case of serious weaknesses, there are even mental and property damages. Although many weaknesses are being resolved by endless analysis and research and development of various security technologies, there are still weaknesses. While the fundamental method to resolve weaknesses is required, it is not easy to find such a method. Therefore, researching on security weaknesses is helpful in developing technologies that can resolve weakness issues. By consistently researching and analyzing security weaknesses, it is expected that development of quick response technology to weaknesses can lead to safer use of Android system.

References

[1] S. Kim, S. Kim and D. H. Lee, "A study on the vulnerability of integrity verification functions of android-based smartphone banking applications". in Journal of The Korea Institute of Information Security & Cryptology (JKIISC), vol. 23, no. 4, **(2013)**, pp. 743-755.

[2] Y.-D. Kim, I. Kim and T. Kim, "Analysis of Usage Patterns and Security Vulnerabilities in Android Permissions and Broadcast Intent Mechanism" in Journal of The Korea Institute of Information Security and Cryptology, **(2012)**, pp. 1145-1157.

[3] C.-K. Han, S.-Y. Kang, H.-B. Jang and H.-K. Choi, "Security Vulnerabilities in Android Applications" in Proceedings of the Korea Information Processing Society Conference, vol. 18, no. 1, **(2011)**, pp. 854-857.

[4] http://www.samsung.com/global/business/mobile/platform/mobileplatform/knox/index_de-vicesecurity.html.

[5] http://erteam.nprotect.com/tag/tj.apk.

[6] http://www.androidsecuritymodules.org/index.html.

[7] http://news.softpedia.com/news/More-Android-4-2-Features-Detailed-SELinu-VPN-Lockdown-00320.shtml.

[8] http://en.wikipedia.org/wiki/Android_version_history.

[9] https://www.samsungknox.com/ko/products/knox-workspace/technical.

Authors

Jae-Kyung Park, Ph.D
Chief researcher at Cyber Security Research Center,
Korea Advanced Institute of Science and Technology,
Nonhyun-ro 28gil 25, Gangnam-gu, Seoul, Korea
E-mail : wildcur@kaist.ac.kr

Corresponding Authors : Sang-Young Choi, Ph.D
Chief researcher at Cyber Security Research Center,
Korea Advanced Institute of Science and Technology,
N5 2321, 291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea
E-mail: csyong95@kaist.ac.kr