

Personal Data Leakage: Android Case Study

Yoichi Saito
University of Aizu
Tsuruga, Ikki-machi Aizu-Wakamatsu,
Fukushima, Japan 965-8580
+81-242-37-2603
s1210196@u-aizu.ac.jp

Vitaly Klyuev
University of Aizu
Tsuruga, Ikki-machi Aizu-Wakamatsu,
Fukushima, Japan 965-8580
+81-242-37-2603
vkluev@u-aizu.ac.jp

ABSTRACT

Android has gained popularity explosively in these days. Android has many security problems. Because of this, various incidents have occurred. A typical example is the leakage of personal information. This incident occurred in the Skype application. Android security features are important to prevent the leak of sensitive information. This study characterizes security issues in Android OS.

Categories and Subject Descriptors

K.6.5 [Management of Computing and Information Systems]: Security and Protection.

General Terms

Security

Keywords

Android Security, Information personalization

1. INTRODUCTION

It is difficult to resolve all possible security issues while developing Android apps because the cost of development should be low. However, if the incident with the application occurs, it is more difficult to recover from losses [1]. Security risks are spreading on the Internet so developers should pay more attention to the problem. In this study, we characterize key security mechanisms.

2. ANDROID SECURITY MECHANISMS AND COMPONENTS

2.1 Android Security Mechanisms

Android security architecture is composed of certificates, user ID, permission and access permission to the files. By combining these elements, Android OS presents the secure structures as follows [2]:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWAIT'15, Oct. 8–10, 2015, Aizu-Wakamatsu, Japan.
Copyright 2015 University of Aizu Press.

- Applications cannot access reciprocal data directly.

- Applications can access determinate parts within the functions of Android.

2.1.1 Cryptography

Android disk encryption is based on dm-crypt, which is a key feature that works at the block device layer. Because of this, encryption works with Embedded MultiMediaCard and similar flash devices that present themselves to the kernel as block devices. Encryption is not possible with YAFFS, which talks directly to a raw NAND flash chip [3].

2.1.2 IPC

Binder and Messenger are classes. They provide the very useful interface. They are preferred for Android IPC. A developer should not design the interface that requires the special permission because objects of the binder and Messenger are not declared within the manifest. As a result, the developer can't change the permission of them. They inherit the permission of the Service or Activity so their permissions need to be set appropriately [4].

2.1.3 SandBox

Android applications follow the rules SandBox framework an application. If it needs to access the resources and data that SandBox doesn't provide, it must declare the access to OS. This declaration is set in the AndroidManifest.xml by the permission [2].

2.2 Components

Components are the elements that compose the application.

- Activity

It provides the function of the user interface [2].

- Service

It doesn't have the user interface. It keeps processing in the background even if other applications are started [2].

- BroadcastReceiver

It receives the broadcast from OS or the other application [2].

- ContentProvider

Thanks for ContentProvider, an application can abstract the entity of the data when it exposes the data outside [2].

3. Data Leakage from the Telephone Directory

Some useful mechanisms are prepared for developers but there are issues to concern. In this section, we characterize these issues.

3.1 Problem Description

When the users exchange the telephone numbers, there is a way that the telephone number is exported to the QR code by the application and the camera of the receiver takes the QR code and gets the telephone number. When this application is installed, “reading the data of the telephone directory” permission is displayed. The user does not expect that this application has the Internet access.

This application may collect data away safe it to the data based on the Internet. This database is the source for data leakage [2].

3.2 How to Get the Personal Information

If the third party analyzes this application, they may know that it is possible to access the ContentProvider. He or she who is malicious develops the application to steal the personal information by using fragilities. Suppose it is an application to download the wallpapers. When this application is installed, “Network Communication/Complete Internet Access” permission is displayed naturally. The user installs this application without any suspicion. But this application does not only download the wallpapers but also pick up the personal information and send them to the server. In Android OS, the application process is defended against the other application, so the other processes do not know how data is send. Any security software cannot detect such a movement [2].

3.3 The Matter to Be Considered

The user has the personal information stolen by the application. To solve this problem, the contentprovider that accesses the personal information should set the appropriate access permission. [2].

A contentprovider is declared in <provider> element of the AndroidManifest.xml file. We characterize two distributions below:

3.3.1 *android:enabled* distribution

This distribution assigns whether this contentprovider is used or not. If the value of “true” is set, this contentprovider can be used. If the value of “false” is set, this contentprovider can’t be used. The default value is “true” [2].

3.3.2 *android:exported* distribution

This distribution controls the setting of the contentprovider extension and assigns whether this contentprovider can be used by the other applications. If the value of “true” is set, all applications can use this contentprovider. If the value of “false” is set, the other application can’t use it, and this application itself or the application that has a same user ID by shareUserId assignment only access this contentprovider [2].

In general, the declarations of the access permission of the other components are done in the same way.

4. CONCLUSION

Android devices are very popular nowadays. They are under attacks from the Internet. Various tools are developed to defend them. Access permission to files is used for android security as well as PC security. In addition, an Android security architecture has the AndroidManifest.xml file. The appropriate definitions in this file make the application secure. However, the attention of developers is more important because they cannot resist attacks unless they notice security risks. Every developer should clearly understand that security techniques have limitations.

5. REFERENCES

- [1] 情報セキュリティ読本 (Information Security Reader, in Japanese). Information-technology Promotion Agency, Jikkyou Publisher, Japan, 2012.
- [2] Taniguchi G., 安全なアプリケーションを作るために (Development of the Secure Application, in Japanese). Impress Japan, Tao Software Corporation, 2011.
- [3] Encryption. Retrieved August 26 , 2015 from Android Open Source Project:
<http://source.android.com/devices/tech/security/encryption>
- [4] Security. Retrieved August 26 , 2015 from Android Open Source Project:
<https://source.android.com/devices/tech/security/>