Journal of Electronics and Information Science(2017) 2: 56-61
Clausius Scientific Press, Canada

Permission Analysis Based Detection Method for Android Malicious Application of Privacy Leakage

Zheling Zeng, Bogang Lin, Yitao Ni, Yongjia Chen

Mathematics and computer science college, Fuzhou University
{zheling, linbg95, yitao_ni, ChenYJ}@fzu.edu.cn

Keywords: Android system; malicious application; detection method; resource access.

Abstract: Nowadays, leakage of Android private information security is protected by application audit mechanism in formal application store and the Android mobile security mechanism. But because of insufficient of these two security, mechanism leakage of privacy incidents happened frequently. In this paper, we get the sensitive resources permission mapping table and sends the data permission mapping table according to the relationship between resources and permission and characteristics of sensitive resources. The judging rules and the detection process for the malicious application of privacy leakage are developed from these two mapping tables. Finally, 884application samples are tested and the experiment result demonstrates the advantages and disadvantages of the method.

1. Introduction

The Android phone has become one of the most popular smartphones and many applications are widely used in the meanwhile. The statistics of the IDC shows that, the google application store has published more than 120 million Android applications. Nowadays, many cell phone manufacturers, such as Huawei, Xiaomi and Samsung, are still work on the development of the Android phone. The Symantec regard those applications that haven't been authenticated by the formal manufacturer as the "gray applications" [1], which have occupied one third of the whole application market. In consequence, there contains great potential dangerous in these gray applications.

At the same time, there are many loophole the legality of the certification application in formal application stores. For example, the google application store only verify whether the identity of the registrant is legal to make sure the legality of the applications, but the subsequent applications from the verified registrant would not be verified again.

Though the sandbox technology and the authorization mechanism are adopted to guarantee the reliability of the kernel, many process communicating with each other will make the private date leak. In today, a malicious attacker could locate the position of the user through the location function in the application, look over the private content in the message, or eavesdrop on conversations to steal and leak privacy data of the users. They make Android information security problems become more severe. In this paper, three works are investigated mainly as follows:

(1) According to the relationship between resources and authority and the characteristics of sensitive resources summed up the sensitive resources - permission mapping table and sending-data resources - permission mapping table.

(2) Based on the two mapping tables, inducing three rules and testing procedures.

(3) We tested 884 samples of applications in the domestic third-party Android application market, and analyze the effectiveness and shortcomings of the method based on experimental results.

2. The ways of checking the applications reveal privacy data

This section is divided into three parts, the first part describes the institution of resource permission mapping table. The second part describes the rules for checking the applications reveal privacy data. The third part introduces the process of checking the applications reveal privacy data.

3. Resource Permission Mapping Table.

The institution of resource permission mapping table is divided into four steps:

(1) Summarizing sensitive resources. Sensitive resources are resources that can store and transfer the privacy data or the valuable one [7,8]. According to the characteristic of privacy data in Android, there are eight resources including messages, contacts, call records, SD card, GPS locator, Bluetooth, camera and microphone

(2) Summarizing outbound data resources. By viewing the instructions in the Android Developers website, you will be able to summarize the resources including the call, message, net-access, E-mail, Bluetooth when the application sends outbound data.

(3) Summarizing the characteristics of the relationship between resources and permission According to the description on the permission in the Android Developers, the relationship between resources and permissions have the following three characteristics:

- The name of the permission contains the name of the resource with which the permission control
- If the permission was registered, the program can only access a resource that is fixed in the phone.
- Several different permissions can be individually access to the same resource.

(4) Getting sensitive resources - permission mapping table, as shown in Table 1 and sending data resources - permission mapping table, as shown in Table 2. First, the resource names of the first and second steps are used as the first column of two tables. Then referring to the relationship between resources and permissions, we manually search on all the resources and add to two tables.

3.1. Rules for checking the applications reveal privacy data.

Based on the two mapping tables obtained in the previous section, the rules for determining privacy data is developed:

Rule1: the permissions in Manifest.XML corresponding to sensitive data - permissions mapping table, the application can steal the privacy data in the phone.

Rule2: the permissions in Manifest.XML corresponding to send data - permissions mapping table, the application can send data out.

Rule3: When and only when the two rules to meet, the event that the application can reveal the privacy of mobile phones is determined.

3.2. The process of checking the application reveal privacy data.

To detect whether the application can reveal privacy data, this section has developed a specific testing process, that can be divided into the following three steps:

- Finding the resources that the application can access. First, exporting the permissions registration information in the Manifest.XML [2,3,6] as shown in Figure 1 and saving.
- Matching the permission information in the first step with the sensitive-permission table. According to the rule1, if match is successful, the conclusion that the application can obtain the privacy data is determined, followed by the third step. If not, the process stops.
- Matching the permissions one-by-one between the permission in the first step with outgoing data-permission table. According to rule 2, if match is successful, the conclusion that the

application can send data outwards is determined, finally drawing that the application will lead to the disclosure of the privacy data.

Table 1. Sensitive resource privilege mapping table

Resource name	Permission name
Message	SEND_SMS；READ_SMS；RECEIVE_SMS；RECEIVE_MMS
Mail list	READ_CONTACTS；WRITE_CONTACTS；
Call log	CALL_PRIVILEGED
SD Card	READ_CALL_LOG；WRITE_CALL_LOG
GPS	WRITE_EXTERNAL_STORAGE
Bluetooth	ACCESS_FINE_LOCATION；
Camera	ACCESS_LOCATION_EXTRA_COMMAND；
Microphone	BLUETOOTH；BLUETOOTH_ADMIN
	CAMERA
	RECORD_AUDIO

Table 2 Send data resources - access mapping table

resource name	Permission name
Call	CALL_PRIVILEGED
Message	SEND_SMS；WRITE_SMS
	BROADCAST_SMS
Bluetooth	BLUETOOTH；BLUETOOTH_ADMIN
network access	INTERNET
E-mail	AUTO_SEND

```
<uses-permission
android:name="android.permission.INTERNET" />

<uses-permission
android:name="android.permission.ACCESS_WIFI_STA
TE" />

<uses-permission
android:name="android.permission.ACCESS_NETWOR
K_STATE" />

<uses-permission
android:name="android.permission.WRITE_EXTERNA
L_STORAGE" />
```

Figure1. Permission registration information sample

4. Experiment and result analysis

To test the advantages and disadvantages of the detection method of the application of data privacy leak formulated by this paper. This section has carried on the test, and analyze the test results.

4.1. Experimental environment.

Deploy the tools in PC. The experimental environment:

operating system: WIN 10 Pro internal storage: 8.00GB
processor: Intel(R) Core(TM) i7-6700HQ CPU@2.60GHz 2.59GHz
Google Galaxy Nexus Android4.2.2 simulator supplied by Motioning
Python2.7、JDK8

4.2. Experimental subject.

We tested 884 samples acquired from domestic application market. These samples contain game, social, news and other applications. There are 668 applications have been shown to reveal private data. And 216 applications are common application.

4.3. Experimental results and analysis.

We conduct two experiments in the paper. It only performs the first two steps of detection process at the first experiment. The second experiment tested the integrated detection process. And analyze the two experimental results respectively.

4.4. Analysis of experimental results for the first experiment.

The test results of 884 samples as shown in Figure 2. 661 applications leak private data by accessing static resources, 519 applications leak private data by accessing hardware resources. We divide these applications into two classes according to the way to gain private data and analysis.

In 519 applications accessing hardware resources. The number of applications is the largest by using GPS. The second largest is the application accessing camera. Most of them are the social, photography or shopping applications. Them often involves money or monitoring ability. The number of applications is the least by accessing Bluetooth or microphone. As shown in Figure 3.

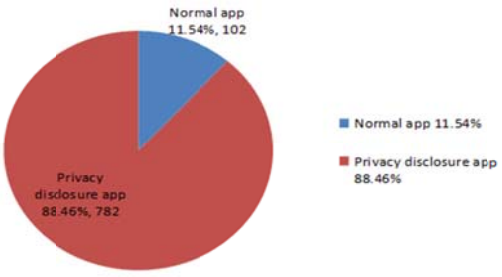

Figure 2 distribution of sample test results

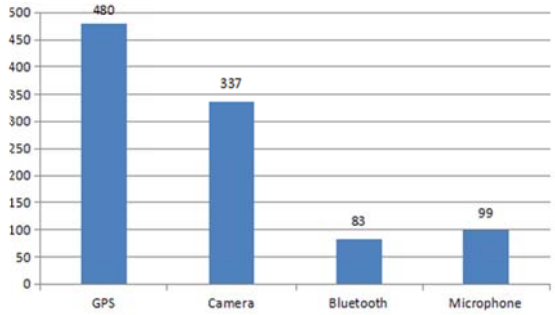

Figure 3 malware distribution map

The number of applications gaining private data by accessing file resources is 611.The number of applications is largest by accessing external storage devices. The second largest is the applications reading the message and address book. They are always the game app and social app. The number of applications by accessing record or system log is the least. As shown in Figure 4.

There are 641 applications can be detected correctly in 782 malwares leaking private data. So, we can figure out that 141 applications are reported falsely, and false positive rate is 18.03%. 27 applications fall to declare, and false negative rate is 3.05%. And we detect that the 27 applications use side channel attacks to leak private data. So, Accuracy and false negative rate are good, but false positive rate is too high and applications using side channel attacks can't be detected effectively.

4.5. Analysis of experimental results for the second experiment.

To test the rule2，we use 782 applications gaining private data in the first experiment as the samples of second experiment.

There are 707 applications that can send data outward in 782 samples. They almost access net. The number of applications sending message is the second largest and sending email or communication is small. As shown in Figure5.

638 applications are malwares in the 707 applications. We can figure out that 69 applications are misinformed, false positive rate is 9.76%. The three applications are fell to declare. Combining with the first experiment, there are 30 applications are fell to declare, false negative rate increase to 3.39%. And sending message method of the increased 3 applications are using the same GID.

We prove that when we use integrated rule, the result has lower false positive rate than just using the rule1, and false negative rate amplification is small, detection effect of applications has been promoted. The method couldn't detect applications using same GID applications to send data.

5. Relevant work progress

Yajin Zhou [4], concluded the installation feature, the code features, the permission features, malicious behavior activation conditions and operation mechanism of malicious applications. Nauman M [5] et al, developed a defense tools on the framework layer: Apex. The tool can discover the threat in time and don't affect the functions of other application at the same time. Zhang N [9] et al. On application layer developed a tool: GUARDIAN. The tool can protect the application data produced when running from stealing by other applications.

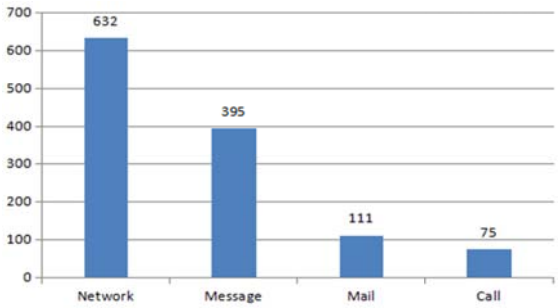

Figure 5 malware distribution map

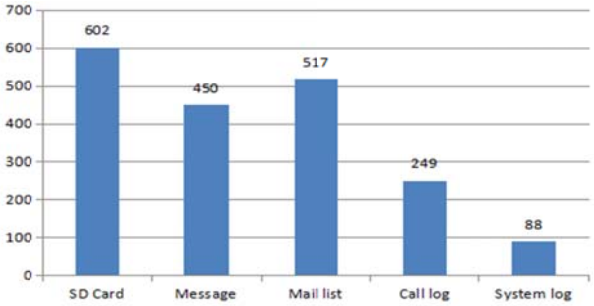

Figure 4 malware distribution map

6. Summary

By concluding the relationship between resources with permission and the characteristics of private resources, we develop a method to detect the malware leaking privacy data. This method can effectively detect most malware in the market. However, it is not very effective to applications using the side channel attacks and applications calling the same GID.

References

[1] Symantec. Twentieth Internet Security Threat Report [J]. Financial Computer of China, 2015, (5):95-95.

[2] Drake JJ, Lanier Z, Mulliner C, et al. Android Hacker's Handbook [J]. 2014.

[3] Elenkov N. Android Security Internals: An In-Depth Guide to Android's Security Architecture [M]. No Starch Press, 2014.

[4] Jiang X, Zhou Y. Dissecting Android Malware: Characterization and Evolution[C]// IEEE Symposium on Security & Privacy. IEEE, 2012:95-109.

[5] Nauman M, Khan S, Zhang X. Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints[C]// 5th international symposium on Acm symposium on information, computer and communications security. 2010:328-332.

[6] Bingquan Xu, Yuan Zhang, Min Yang. GrantDroid: a kind of support Android

[7] Yubin Wang, Chao Li, Nan Cheng. Research on personal sensitive information protection of Internet [J]. information network security, 2014(9):144-148.WANG Y B, LI C, CHENG N. Research personal sensitive information protection on Internet[J]. Information Network Security, 2014(9):144-148.

[8] Ker A, Watt S, Myrhaug H I, et al. An ambient, personalized, and context-sensitive information system for mobile users [C]// European Union Symposium on Ambient Intelligence. ACM, 2004.

[9] Zhang N, Yuan K, Naveed M, et al. Leave Me Alone: App-Level Protection against Runtime Information Gathering on Android [J]. 2015:915-930.