© 2014 IJIRT | Volume 1 Issue 6 | ISSN : 2349-6002

Design of Authenticity Evaluation Metric For Android Applications

Pranshu Sharma , Satish Anand Arjunan

***Abstract*— For enforcing security, Android platform uses authorizin g system which grants permission per application at install-time. W ith authorized privilege, user applications can modify and delete us er's personal information. Therefore, inspection of granted permiss ion usage can be used to detect security vulnerabilities. ISO/IEC 25 010 defines software product security characteristic and provides g uidelines to evaluate software product quality. Among sub-characte ristics of security, Authenticity is related to Android permission sys tem. In this paper, we present authenticity metric for android appli cation. This metric can quantify the permission usage of application and measured information can be used to classify the malware app lications. To verify the applicability of metric, we perform evaluati on to benign and malware application and compare its results.**

***Index Terms*- security; metric; android; permissions; least privilege; authenticity**

I. INTRODUCTION

In 2013, Android is accounting for more than 78% of smartphone sales to end users by operating system [1]. With the growing popularity of Android platforms, users can download many applications from various android markets (Google Play Market, secondary market, etc.). Unsurprisingly, Android has attracted a huge number of attacks. And there is a lot of malware which is not certified. Therefore, application analysis is needed to bring software security to a higher level. Especially, most of android application security issues are related to abusing personal information. Malware sends information to anonymous server or sends so many SMS in order to billing unnecessary charges. Android platform adopts install-time permissions to protect sensitive resources from un-trusted apps. However, an install-time permission system is ineffective if developers routinely request more permissions than they require [2]. Thus, a lot of malware attacks occurred by this weakness of android permission system. To address these issues, many analysis techniques were researched [3, 4].

Traditional static analysis techniques extract features from large amount of malware sample and classify new applications by feature similarity. They can sense vulnerability of applications fast, but most of static methods are difficult to identify permission usage because they are based on code pattern analysis only. And it is hard to represent the security characteristics of applications.

Among characteristics which ISO25010 defines in its product quality model, security has authenticity sub-

characteristic. Authenticity is the identity of a subject or resource that can be proved to be the one claimed and it is necessary to ensure that the data, transactions, communications or documents are genuine.

Android platform grant permissions to each application when it is installed. Thus, Authenticity is needed to evaluate to the permission usage of an application.

We propose a metric of authenticity evaluation for android application. The metric is a measure of permission usage and it can be used to provide quantitative measurement which checks whether the application is over-privileged or not. Because it can find security vulnerabilities of permissions, it will be a basis for authenticity evaluation and can be used to shorten security analysis time.

The rest of the paper is organized as follows. Section II describes overview of android security system, security characteristic of ISO/IEC25000 and relations between each other. Section III, then, presents characteristics of the authenticity metric we suggested in detail and evaluation of it by experiment is attached. Finally, Section IV concludes and explains future works.

II. RELATED WORKS

A. *Android Application Security*

In Android, application security is based on isolation and permission control to protect user data and system resources [6, 7]. Each applications runs in its own *sandbox*. Application isolation is provided by Linux kernel level security architecture. So separated application is reaching only limited resources and it doesn't share any resources.

In addition, an application is granted permission at install time by declared contents in its AndroidManifest.xml file. Only permitted application can use API to access personal information. It takes advantage of control properly, but sometimes application requests more permissions than what they actually required [2].

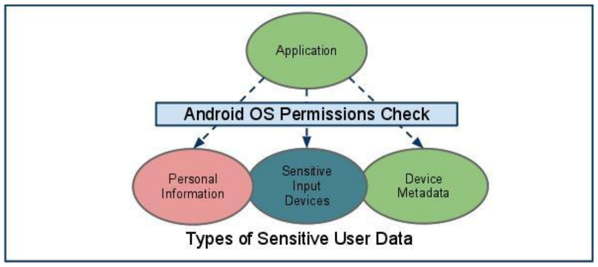

Figure 1. Overview of Android Permission system

It's because users do not pay attention to permissions when they install an application [8]. Extra permissions unnecessarily condition users to casually accept dangerous permission and needlessly exacerbate application vulnerabilities.

B. ISO/IEC25000 Security

In 2011, ISO announced the new standard for software product quality ISO/IEC 25010 [5]. The big difference with existing standard is software security is defined as the main quality characteristic. Security characteristic is defined as "degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization". Security has five sub-characteristics and it is based on the access control and monitoring functions.

- *Confidentiality*: ensures that data are accessible only to those authorized to have access
- *Integrity*: prevents unauthorized access to, or modification of , software or data
- *Non-repudiation*: actions or events can be proven to have taken place
- *Accountability*: actions of an entity can be traced uniquely to entity
- *Authenticity*: the identity of a subject or resource can be proved to be the one claimed

With this product model, to evaluate security of practical system is being researched [9, 10]. In general, these characteristics can be measured by using the internal data and permission usage. However the four characteristics, except for authenticity, are based on platform security for inner workings. So, specific analysis of those characteristics is difficult to represent. But, Authenticity purpose to admin requested permissions and it is related to Android application which requests permissions at install time.

In this paper, we represent authenticity evaluation metric to check granted permission.

III. AUTHENTICITY EVALUATION METRIC

A. Design of metric

Authenticity Evaluation Metric measure that Android system grants appropriate permission to applications which access to system resources, and clarify that granted the privilege is being used to right time.

Analyze AndroidManifest.xml to get count of requested permission. Used permission can count using characteristics of android. According to permission, application can access different APIs. Analyze source code to extract API list and matching it with the permission map from *android-*

For this purpose, using characteristics of android system, defines the metrics. Metrics shall be composed of metric, calculations, and result area. Result area is a constant greater than 0. Less than 1 score means used permission count in code is less than the total number of requested permission. It can be over-privileged status. Over 1 score means application try to use more permission than those requested. It is latent error status.

B. Experiments and inspection.

We built a static analysis tool, Code Analyzer, which analyze an Android application and count set of permissions. With this, we can evaluate Authenticity metric. Main Process is as shown in Figure 2.

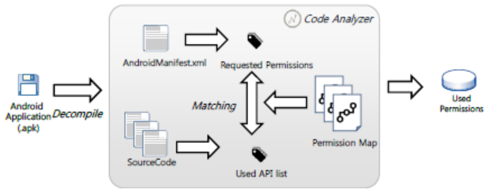

Figure 2. Measurement Environment

permissions.org[11] . If API usage exists in code one or more, matched permission is measured 'in use'.

We evaluated 1133 benign apps from Android market and secondary market, and 1250 malware apps from *android malware genome project*[12]. Because we need to evaluate permission usage, 465 of benign apps and 68 of malware apps which do not request any permission are excluded from result graph.

Figure 3 shows the result of evaluation. We can analysis information as follows:

First of all, 15% of benign applications score 1 which means count of requested and used permission is same. In contrary, only 1% of malware applications score 1. Authenticity of malicious application is lower than normal application's authenticity. Further, about 45% of total malware apps showing less than 0.2 authenticity score, but only 10% of benign apps score lower than 0.2. This means over-privileged permission is likely to be utilized in a malicious action. But, Almost 80% of benign and 99% of malware apps score below 1. A lot of applications request more permissions than they really use. Thus, Android applications are easy to be exposed to security threats.

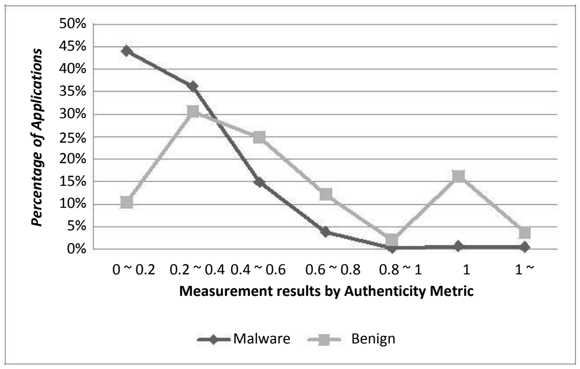

Figure 3. Comparison of Malware and Benign application Authenticity

Table 2 shows sample measurement result of application QuickSettings. It is a variant of DroidKungFu3 malicious application. This app request total 41 kinds of permission but, only 9 permissions are used. In this case, permission re-delegations or malicious behavior can be originated from over-privileged status and it can be potential security vulnerabilities.

TABLE II. SAMPLE MEASUREMENT RESULT

Title	QuickSettings
Used Permission	android.permission.ACCESS_FINE_LOCATION
	android.permission.ACCESS_WIFI_STATE
	android.permission.BLUETOOTH
	android.permission.BLUETOOTH_ADMIN
	android.permission.CHANGE_WIFI_STATE
	android.permission.INTERNET
	android.permission.READ_CONTACTS
	android.permission.READ_PHONE_STATE
	android.permission.VIBRATE
Total Count of Permissions	41
Measurement Result	9 / 41 = 0.219512194

Using proposed method, thus, we can quantify security weakness with authenticity score. And it can be used to classify whether the application is malware or not. This evaluation metric reduce number of target applications with fast analysis, and can be utilized to detect malware candidates.

IV. CONCLUSION

In this paper, we presented authenticity evaluation metric for android applications. Evaluation metric is considering the characteristic of android permission system. We could detect over-privileged status with the metric and analyze permission usage rate. We applied this metric to 2283 Android application and found that most of malware applications are over-privileged. Our results show that applications need to permit properly at development cycle. With authenticity score, we can detect potential vulnerabilities. Therefore, it could be used to prohibit over-privileged status for ease of development and to draw malicious candidates.

Limitation of this work is that the analysis results are based on static analysis information. Thus, it cannot detect runtime permission use state. Our future work will focus on this. A static analysis method will be used to reduce number of target application and a dynamic analysis method will be added to analyze android platform inner side. Finally, we will design evaluation metrics for other sub-characteristics of security (confidentiality, integrity, etc.)

REFERENCES

[1] Gartner, "Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013", [OnLine]: http://www.gartner.com/newsroom/id/2665715

[2] FELT Adrienne Porter, et al.., "Android permissions demystified", Proceedings of the 18th ACM conference on Computer and communications security. ACM, pp.627-638, 2011

[3] YANG Zhemin, et al., "Leakminer: Detect information leakage on android with static taint analysis", Software Engineering (WCSE) 2012 Third World Congress on IEEE, pp.101-104, 2012

[4] ZHANG Yuan, et al.. "Vetting undesirable behaviors in android apps with permission use analysis", Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security. ACM, pp.611-622, 2013

[5] ISO, "ISO/IEC FDIS 25010:2011 : Systems and software engineering - (SQuaRE) - System and software quality models", 2011

[6] Android, "Android Security Overview", [OnLine]: http://source.android.com/devices/tech/security

[7] LIEBERGELD Steffen, et al., "Android security, pitfalls and lessons learned." Information Sciences and Systems 2013. Springer International Publishing, pp.409-417, 2013

[8] FELT Adrienne Porter, et al., "Android permissions: User attention, comprehension, and behavior", Proceedings of the Eighth Symposium on Usable Privacy and Security, ACM, pp.3-17, 2012

[9] XU, Haiyun, et al., "A Practical Model For Rating Software Security", In: Software Security and Reliability-Companion (SERE-C), 2013 IEEE 7th International Conference on. IEEE, pp.231-232, 2013

[10] COLOMBO, Regina Thienne, et al., "Prioritization of software security intangible attributes", ACM SIGSOFT Software Engineering Notes Vol.37.6: pp.1-7, 2012

[11] Stowaway, "android-permissions", [Not Available]:http://android-permissions.org/

[12] ZHOU Yajin, et al.., "Dissecting android malware: Characterization and evolution", In: Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, pp.95-109, 2012