Mechanism for Security Enhancement in Mobile Application Installation

Peng Zhang
Research Institute of Mobile Internet
Xi'an University of Posts and Telecommunications
Xi'an, China
pzhang@xupt.edu.cn

Hanlin Sun
School of Compute Science
Xi'an University of Posts and Telecommunications
Xi'an, China
sunhanlin@xupt.edu.cn

Zheng Yan
Department of ComNet, Aalto University
Espoo, Finland
State Key Laboratory of ISN, Xidian University
Xi'an, China
zheng.yan@aalto.fi, zyan@xidian.edu.cn

***Abstract*—Mobile operating systems (e.g., iOS, Android, Windows Mobile, etc) are becoming powerful platforms on which various applications can be installed and run. Each mobile OS offers application store (e.g., Apple App Store, Android Play, etc) for developers to easily publish applications and earn profits. However, existing mobile platforms provide little means for mobile users to evaluate risks on allowing certain security permissions when installing mobile applications. Since mobile users may not be able to justify the risks on allowing certain permissions required by an application, mobile users may install malware with extra permissions, which leads to security risk for mobile users, e.g., private information leaked, etc. In this paper, we present process and visual User Interface for mobile users to understand and justify the risks on permissions required by mobile applications during installation. We also present two algorithms for calculating the risks.**

Keywords- mobile OS; application installation; trust; permission; User interface

I. INTRODUCTION

Nowadays, mobile operating systems (e.g., iOS, Android, Windows Mobile, etc) are becoming powerful platforms on which various applications have been developed. Each mobile OS offers application store (e.g., Apple App Store, Android Play, etc) for developers to easily publish applications and earn profits. The invention of App Store has enabled the establishment of so-called ecosystem around mobile OS. Mobile users have enjoyed envision of thousands of mobile applications including entrainment applications and business oriented applications. Embracing a large number of applications, mobile OS application stores provide means for mobile users to search and select their preferred applications, e.g., list of top free downloaded games, comments and rating, etc.

However, mobile OS application stores provide little information about the security and trustworthiness of mobile applications. Even though most mobile OS stores have predefined procedures on validating and publishing mobile applications, it is not unusual that the mobile applications downloaded by mobile users contain malware, e.g., stealing users' privacy information without users' notice. Mobile users normally lack enough information to justify the quality and trustworthiness of applications against their preferences when selecting mobile applications. For example, when installing a mobile application in Android device, the user may be asked for various permissions (e.g., access Internet, access private data, etc). Without any clear description why the application needs these permissions, and what are the risks on allowing these permissions, the user can easily get into risky situation losing money and/or private information [1].

Mobile anti-virus software can provide security protection for mobile devices by inspecting mobile applications and monitor the running behaviors of applications. But mobile anti-virus software may hardly protect users' privacy information, e.g., an application may upload users' private information without users' full awareness. Moreover, since anti-virus software causes heavy load on CPU and power consumption, mobile users may run anti-virus software less frequently to save battery life. Thus, in order to protect mobile users from malware, it is necessary to evaluate risks during application installation, and not install an application seen as risky, e.g., an application asks for some permission that are not seen needed.

In this paper, we present mechanisms for improving permission evaluation during application installation. When an application asks for a number of permissions during installation, a UI is shown for users to view the details of the required permissions and the details of corresponding features that require the permissions, users can give their rates on these risks and features, then a summarized evaluation result UI helps users justify the risks implied by the permissions. This process and visual User Interface will help mobile users to understand and justify risks on permissions required by mobile applications during installation.

The rest of the paper is organized as follows. In section II, we give related work. In section III, we outline the mechanism for evaluating the risks on permissions required by applications. In section IV, we present two algorithms for risk assessment and conclude the paper in final section.

II. RELATED WORK

Most prior art has focused on usable security in human computer interaction including visualization of security process and enforcement [2-3]. Some existing work addressed security and privacy issue in mobile devices [4-5].

Published by Atlantis Press, Paris, France.
© the authors

Reference [6] presented a system and method for preventing malware, spyware and other undesirable applications from affecting mobile communication devices. A mobile device uses a server to assist in identifying and removing undesirable applications. When scanning an application, a device transmits information about the application to a server for analysis. But the solution does not address the risks on permissions required by applications.

Reference [7] presented a policy enforcement framework for Android that allows a user to selectively grant permissions to applications as well as impose constraints on the usage of resources. Reference [8] introduces the basics of Andrioid, not giving any solutions for solving the permission problem. Reference [9] provided a solution to ensure the execution of a program against declared permissions.

Reference [10] provided a solution to detect an application's trustworthy by comparing its declared permissions against the application's category. But it does not include parts in UI and user evaluation. Reference [11] discovered the permission problem, not solving the problem.

Overall, there still lack effective solutions to help mobile users understand and justify the risks on permission required by applications.

III. MECHANISM

In this section, we present the mechanism for mobile users to understand and justify the risks on permissions required by applications during installation. We firstly introduce the process and logical diagram of permission evaluation, and then we present UI design for permission evaluation.

A. Process and logical diagram of permission evaluation

Figure 1 shows the process for mobile users to evaluate the risks on permissions during application installation. Starting with an application installation, the application asks for a set of permissions. The permissions and corresponding feature details are shown. User rates Risk Level of permission and Desire Level of the corresponding feature. Then, a permission evaluation result UI is shown for users to decide whether to allow the required permissions.

Figure 2 shows the logical diagram of permission evaluation. Permissions required by features imply risks. Permissions are required by features. Both risks and features have impact on users' expectation. Mobile users decide to allow permissions based on expectation against features vs. risks.

B. Visual UI design

Figure 3 illustrates visual UI of permissions asked by an application. Permission includes two sub-parts: permission details including risks, and feature description for what the permission is needed. If Details/Features cannot be shown completely in this space, mobile users can click Details/Features and see the complete details/features description in separate windows. Users can set Risk Level as they perceive, e.g., given a number between 5 and 1, users may select 5 as most severe in their opinions. Users also set Desire Level of the feature in their opinions, e.g., given a number between 5 and 1, user may select 5 as the most favorable feature. Note that the Risk/Desire level can be represented in other UI forms, e.g., rating stars, color bars.

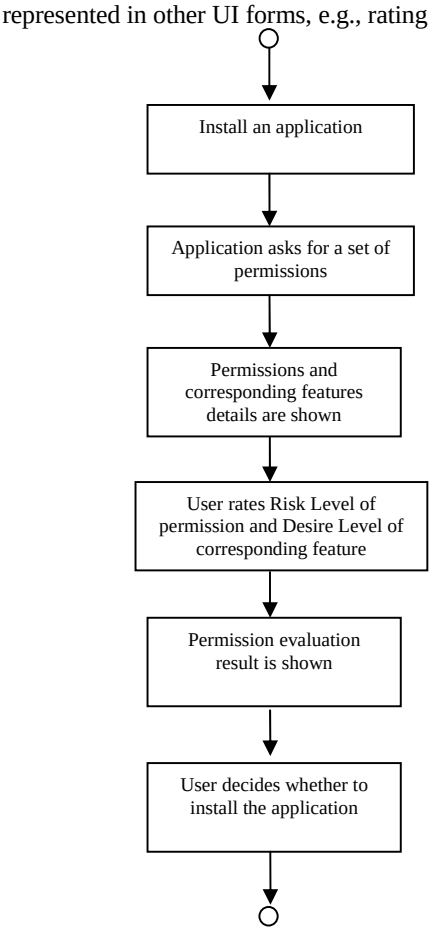

Figure 1. Process on permission evaluation during application installation

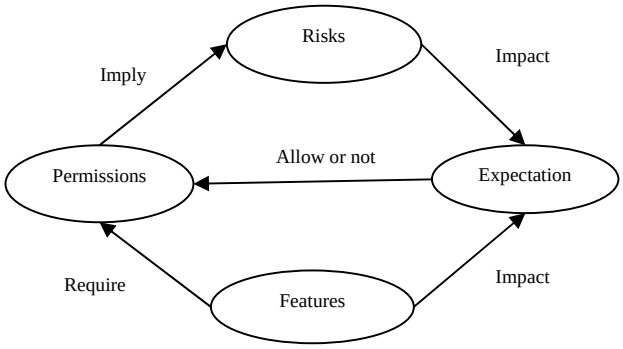

Figure 2. Logic diagram of permission evaluation

Moreover, the default value of Risk Level can be configured by mobile device manufacture, OS provider or Security Service provider. For example, a Security Service Provider (SSP) can provide a service for mobile devices to get the default Risk Level values of permissions. SSP can provide support for mobile users to evaluate risks on

© the authors

permissions, e.g., set default Risk Level of permission, show user rating on Risk Level, percentage on an application installation and removal, etc.

Figure 4 illustrates the values of Risk Level, Desire Level, and their corresponding numeric values.

Permissions asked by an application		
Permission 1	Details Risk level: x1	Feature Desire level: y1
Permission 2	Details Risk level: x2	Feature Desire level: y2
Permission 3	Details Risk level: x3	Feature Desire level: y3

Figure 3. Visual UI of list of permissions

Value	Risk level	Desire level
5	Very high	Most favorable
4	High	Like
3	Neutral	Neutral
2	Minor	Dislike
1	N.A.	N.A.

Figure 4. Values of risk level and desire level

Figure 5 shows visual UI of evaluation result. The result includes evaluation values with color bar and details.

Figure 6 shows the color codes of evaluation results. If the evaluation result is good, i.e., Go, the color code is green; if the evaluation result is moderate, i.e., Neutral, the color code is yellow; if the evaluation is negative, i.e., No Go, the color code is red. With this way, users can visually see the risk of permission against their expectation.

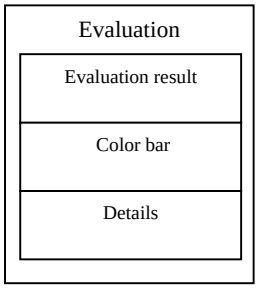

Figure 5. Visual UI of permission evaluation result

Evaluation	Color
Go	Green
Neutral	Yello
No go	Red

Figure 6. Color code

IV. ALGORITHMS

In this section, we present two algorithms for calculating the evaluation result.

Given:

$X = \{x_i, i \in [1, \ldots, K]\}$, x_i is the risk level value of i_{th} permission, K is the number of permissions/features.

$Y = \{y_i, i \in [1, \ldots, K]\}$, y_i is the desire level value of i_{th} feature corresponding to i_{th} permission, K is the number of features/permissions.

Thus, we get $E = \{y_i - x_i, i \in [1, \ldots, K]\}$, and $\bar{E}$ is the mean value of E.

Algorithm 1 is shown as follows.

> Algorithm 1:
> Given a positive value θ,
> If $\bar{E} > \theta$, the Evaluation Result is GO, i.e., color code = Green.;
> If $\bar{E} \in [-\theta, \theta]$, the Evaluation Result is Neutral, i.e., color code = Yellow;
> If $\bar{E} < -\theta$, the Evaluation Result is NO GO, i.e., color code = Red.

Algorithm 2 is shown as follows.

> Algorithm 2:
> Given a threshold of Risk level C_x, a threshold of desire level C_y.
> For $i \in [1, \ldots, K]$,
> If any $x_i > C_x \ AND \ y_i < C_y$, the Evaluation Result is NO GO, i.e., color code = Red.;
> If all $x_i < C_x \ AND \ y_i > C_y$, the Evaluation Result is GO, i.e., color code = Green.;
> Otherwise, the Evaluation Result is Neutral, i.e., color code = Yellow.
> C_x and C_y can be configurable, e.g., mobile user can select Security level=High/Medium/Low, which sets different values of C_x and C_y.

Note that the evaluation can be performed not only during application installation, but also used as security indication after application installation. That is, after an application is installed, the system can perform risk evaluation based on users' behaviors, and monitor whether the application breaches security permissions against the features.

V. CONCLUSIONS

Mobile operating systems provide platforms for mobile users to download and install applications. However, mobile users may lack enough information to understand and justify the risks on permissions required by applications. In this paper, we present a mechanism to help mobile users evaluate risks against their expectation. The mechanism consists of process and UI design for evaluating risks on the permissions. We also present two algorithms for permission evaluation.

ACKNOWLEDGMENT

© the authors

This work is supported by Natural Science Foundation of Education Department of Shaanxi Province, China (Grant No. 11JK1018) and Shaanxi International cooperation key project (Grant No. 2012KW-03-01).

REFERENCES

[1] A. Rodriguez, "Android's permission problems", http://www.pcworld.com/article/251824/androids_permission_problems.html. 2012

[2] M. A. Sasse and I. Flechais, "Usable Security: Why Do We Need It? How Do We Get It?" In: Cranor, LF and Garfinkel, S, (eds.) Security and Usability: Designing secure systems that people can use. O'Reilly: Sebastopol, US. pp. 13-30.

[3] J Rode, C Johansson, P DiGioia, and K Nies. "Seeing further: extending visualization as a basis for usable security". Proceedings of the second symposium on usable privacy and security, 2006, pp. 145-155, doi: 10.1145/1143120.1143138.

[4] D. Ferebee and D. Dasgupta, "Security visualization survey". Proceedings of the 12th colloquium for information systems security education, Jun. 2008, pp. 119-126.

[5] R. Balebako, P. G. Leon, H. Almuhimedi, P. G. Kelley, J. Mugan, and A. Acquisti et al. "Nudging users towards privacy on mobile devices". The human-computer interaction(CHI) 2011 workshop , Vancouver, BC , May 2011.

[6] K. P. Mahaffey, "System and method for server-coupled malware prevention". US patent application, 2011

[7] M. Nauman, S. Khan, and X. Zhang, "Apex: extending android permission model and enforcement with user-defined runtime constraints". Proceeding of the 5th ACM symposium on infomration, computer and communication security, 2010, pp. 328-332, doi: 10.1145/1755688.1755732.

[8] A. Zlotnick, "Verification system and method for accessing resources in a computing environment", US patent, US20070294530, 2007.

[9] K. E. Corby, A. Goldfeder, and J. M. Hawkins, "Method and system for ensuring that computer programs are trustworthy", US patent, US20060090192, 2006.

[10] R. Belani and A. Higbee, "Methods and systems for rating privacy risk of applications for smart phones and other mobile platforms", US patent, US20120072991, 2012.

[11] A.P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: user attention, comprehension, and behavior." Technical report UCB/EECS-2012-26, University of California at Berkeley (2012)

© the authors