A SECURITY ASSESSMENT METHOD FOR ANDROID APPLICATIONS BASED ON PERMISSION MODEL

Danyang Jiang, Xiangling Fu, Maoqiang Song, Yidong Cui

School of Software Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
jiangdanyang@qq.com

Abstract: Permission-based security model of Android restricts applications to access specific resources, but malicious applications can invade more easily in such user-centric pattern. Through the analysis of the Android Permission-based security model and the permission features of Android applications, we establish the permission model to quantify the functional characteristics of the application**,** and then provide an assessment method in which we use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not so as to help users choose applications before installation. We test the assessment method on 873 applications available online and do the statistic and analysis of the results to find that this method can do efforts in finding potentially malicious applications.

Keywords: Android; permission; security; malicious application; permission model; visualization technology

1 Introduction

Since the first launch in 2007, the rise of the Android platform has been meteoric. As the first fully open-source mobile equipment comprehensive platform, Android finish most kinds of different functions through the various applications, therefore Android applications is undoubtedly the most critical factor in the rapid development of Android equipment platform. Projections indicate that: the next generation of open operating system platform is no longer a PC or a mainframe, but the small mobile devices [1].

As Mobile equipment play an important role in today's world and have become an integral part of our daily life as one of the predominant means of communication [2], more and more people pay attention to the security of the Android applications, according to a 2011 Juniper Networks report, and follow up press release, they found "a 472% increase in Android malware samples since July 2011" [3]. Although Android permission mechanism set restrictions to access specific resources, it fails to provide users with adequate control over and visibility into how third-party applications use their private data. Meanwhile, the liquidity of application markets also increased the security risks of the applications [4].

978-1-4673-1857-0/12/$31.00 ©2012 IEEE

The paper describes a method to assess the security of Android applications before installation. In this method, we establish the permission model to quantify the functional characteristics of the application based on the analysis of the Android permission-based security model and the permission features of Android applications, and then use the network visualization techniques and clustering algorithm to determine whether the testing application is potentially malicious application or not.

The rest of this paper is organized as follows. Section 2 describes related work, Section 3 describes the design basis of assessment method, Section 4 describes the datasets and the steps of the assessment method, Section 5 displays the assessment result and the analysis of the result and Section 6 discusses the limitations of our method and the future work.

2 Related work

The security of Android applications is a growing concern. A research report: Android whether to go further in the future depends on the security model [1]. In order to reduce the security vulnerabilities of the Android operating system, so as to strengthen the security of the Android mobile device, OS-level protections such as Saint [5] provide enhanced security mechanisms for Android; In Android application security research, Asaf Shabtai analyzed the threats of the Android framework and defined the crisis values to various threats, and then provided some of the proposed solution towards these threats [6]; In the improvement of the Android framework, Machigar Ongtang proposed an improved framework to strengthen internal data communication security by using encryption/decryption technology [7]; William Enck [8] developed a application with a specific rule set to determine whether the application is a malicious one or not In the aspect of Android application installation; In the management of Android application permissions, Nauman put forward Apex [9] which can modify the permissions of installed applications, and Felt presented a method for detecting the over-privileged applications [10]; In the research of malicious behaviors, Delac illustrated the types and motivations of malicious applications beyond mobile platform [11]. Our work mainly focuses on the characteristics of the application and the Android

permission model, in order to find a way to help user find potentially malicious applications.

3 The design basis of the assessment method

3.1 The establishment of the permission model

Tanahashi has done a research on the similarity between the Android applications [12]. His research pointed out that the application needs to apply for a permission to obtain the access to the corresponding resources due to the requirements of Android permission security mechanisms, therefore the functionalities of an application can be reflected by the permissions that applied by it. Thus the similar applications can be defined as: the applications of the same or a similar declaration of permissions.

There are 124 permissions defined in Android 2.3, we can identify each application by a 124-dimensional vector（x_1, x_2, x_3, ..., x_{124}）, where each dimension represents a permission, and measured as $x_i = \frac{p_i}{n} \times log \frac{D}{d}$ by the improved model of the TF-IDF algorithm [13], (i represents the permission that mapped by the i-dimension of the vector, p_i measured as 1 if the application request permission i, and as 0 if does not request; n represents the number of permissions applied by the application; D is the number of applications in the test sample; d indicates the number of applications that apply for permission i).

3.2 Calculation of the similarity between applications

After assigning each application a vector, we can approximate a pair of applications' similarity base on the cosine of the angle values of the vector, $s_{i,j} = \frac{\sum_1^n x_i y_i}{\sqrt{\sum_1^n x_i^2} \times \sqrt{\sum_1^n y_i^2}}$, ($s_{i,j}$ represents the similarity between application I and application j).

3.3 Clustering features of application categories

The applications are defined as different types according to their functionalities in the Android markets, and the applications that belong to the same category have most of characteristics of this type of clustering. In other words, they have a higher similarity. In order to better explain the close correlation between applications in the same application category, we download the first 100 applications of lifestyle application type in third-party application market called AnZhi, Figure 1 is derived from network visualization techniques based on the permission model described above.

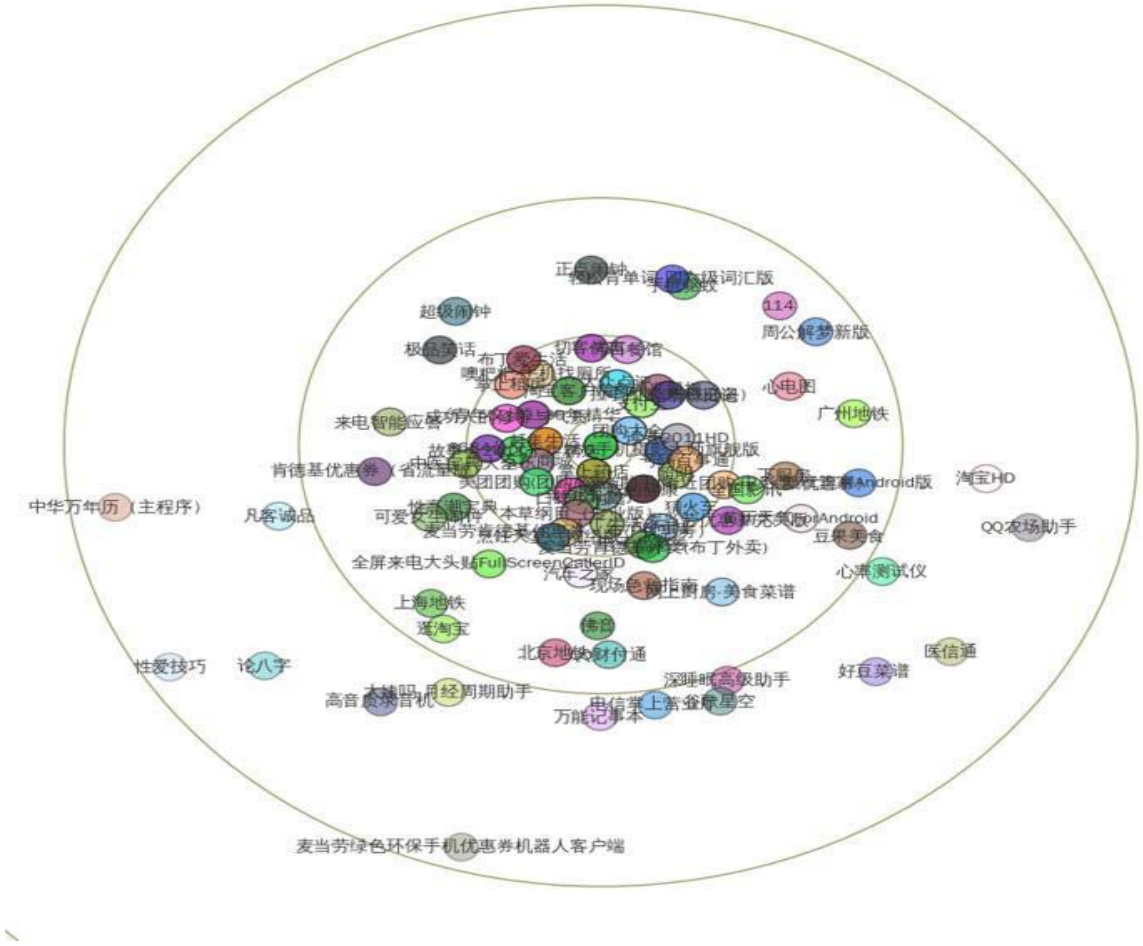

Figure 1 Distribution of applications belong to lifestyle category

In Figure 1, each small circle represents the specific application and the name of application is located next to the circle. The dot matrix arrangement is drawn by the Edge-Weighted Force-Directed algorithm; the distance between small circles indicates the similar of the two applications, the greater the similarity, the closer the distance between circles. It can be seen from Figure 1, most of them have high similarity, but there are several applications staying in the peripheral, such as the application named "The Chinese Calendar", the phenomenon of low similarity towards other application due to that it apply special permissions, such as SDCard storage, phone state and reading the contact. Two conclusions can be made: the application has personalization features or it is a potentially malicious application.

4 Details of the assessment method

4.1 Data

There are two datasets in this paper: Test Set T and Sample Set S. Sample Set S has 500 APKs of browser, e-book, finance, humanity, input-method, lifestyle, multimedia, system-tool, theme and travel, 50 APKs per category; Test Set T is derived from 876 APKs that were freely available online and each APK belongs to a subject category according to its functionality determined by third-party Android Market called AppChina. Table I shows the number of applications in different categories in Test Set T.

Table I Potentially malicious applications belong to category theme in test set T according to exceptional value in descending order

Application Name	**Exceptional *v*alue**	**Special *p*ermission**	**Potentially malicious *b*ehavior**
Audio-visual class plug-in	0.7429404	ACCESS_WIFI_STATE READ_HISTORY_BOOKMARKS WRITE_HISTORY_BOOKMARKS	Get the user's internet records Modify the state of the wireless network
Touch of beauty	0.74249318	ACCESS_FINE_LOCATION READ_LOGS	Access to the user's location info Read the logs of all applications
Volume console	0.74237786	READ_CONTACTS WRITE_CONTACTS	Free to modify the user's contact
Universal Notepad	0.73987077	SEND_SMS RECEIVE_BOOT_COMPLETED	Get the signal of switching on Send high-fee SMS
Desktop contact plug-in	0. 73231349	RESTART_PACKAGES MOUNT_UNMOUNT_FILESYSTEMS KILL_BACKGROUND_PROCESSES	Restart the package service Delete some background process Login/logout file system
…	…	…	

4.2 Steps of the assessment method

First, we give the definition of the security application rate of Category C, P_C= $\frac{\text{the number of secure applications in category C through out the assessment}}{\text{the number of applications of category C in the Test Set T}}$. And then assess the applications of different categories in Test Set T respectively by using improved KNN (K-Nearest-Neighborhood) algorithm [14]. The test steps are as follows:

1) Take out Application C_A of Category C and define the Application C_A as a 124-dimension vector $(a_1,a_2,\ldots,a_{124})$ based on the permission model mentioned in Section 3.

2) And then calculate the similarities between Application C_A and the other applications in Sample Set S by using vector angle cosine formula, $S_{a,b}=\sum_1^n a_i b_i \Big/ \sqrt{\sum_1^n a_i^2} \times \sqrt{\sum_1^n b_i^2}$ (here n is 124, $(b_1,b_2,\ldots,b_{124})$ represents an application B in Sample Set S).

3) Sort the similarities from highest to lowest and select the first K applications.

4) Figure out the weights of each category of Application C_A, the weight of Category C is defined as W_C= $\sum_{i\in K} s_i \times q_C(i)$ (s_i represents the similarity between Application C_A and applications among the K applications mentioned above, q_J is measured as 1 if application s_i belong to Category J, as 0 if not; n_J indicated the number of applications in the K applications that belong to Category J).

5) Finally, Check whether WC = max{WI, I∈AC}, if does, Application CA is judged secure, if not, Application CA may be a potentially malicious application, and the exceptional value is defined as $E_{C_A} = 1 - \frac{W_C}{\sum_{i\in AC-\{C\}} W_i}$ (AC represents all categories in the Test Set T, here the number of elements in AC is 10).

5 Statistic and analysis of the assessment results

5.1 Statistic of assessment results

We take the value of K as 5 when using the revised KNN algorithm. The value of K is determined by statistic and testing which can be neither too big nor too small. The assessment results are shown in Table II. And we list the several applications belong to Category Theme according to exceptional value in descending order in Table I.

Table II Statistic of assessment results in test set T of different categories

Category	Num. of applications	Num. of *p*otentially *m*alicious *a*pps	Security *a*pplication *r*ate
Browser	63	5	0.920634921
E-book	64	21	0.671875
Finance	83	27	0.674698795
Humanity	94	54	0.425531915
Input-method	49	9	0.816326531
Lifestyle	81	23	0.716049383
Multimedia	116	49	0.577586207
System-tool	111	43	0.612612613
Theme	130	41	0.6846153846
Travel	82	30	0.634146341
Total	873	332	0.6419702176

5.2 Analysis of the assessment results

As can be seen from Table II, Category Browser and Category Input-method have higher security application rate against Category Multimedia and Category Humanity. Then we analyze the assessment results from three aspects: a. Characteristic of applications that identified as potentially malicious; b. the reasons for the difference of security application rate between diverse categories; c. Characteristic of applications that have high exceptional value:

1) Characteristic of potentially malicious applications: a. A part of them in the Android market are classified into the wrong category; b. some of them belong to the categories that have vague classification boundary against other categories such as Category E-book and Category Humanity; c. and some of them apply for far too much permission.

2) The main reason for the difference of security application rate is that the classification of categories is in accordance with the requirements of people but not strictly classified by functionality and features of applications. Simultaneously, the value of K plays an important role. Figure 2 shows the influence on Category Theme caused by K.

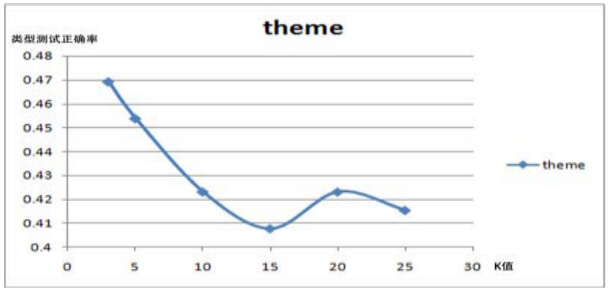

Figure 2 The variation of security application rate of category theme caused by K

3) Characteristic of applications that have high exceptional value: a. Some of them apply for the permissions that is rarely applied by other applications belong to the same category, such as the second application named “touch of beauty” in Table I which applies for a special permission called android.permission.READ_LOG; b. Some of them have litter common characteristic of the category that they belong to.

6 Conclusions and future works

In this paper, we provide an assessment method in accordance with certain logic, reliable theoretical basis and existing research results in order to help users estimate the applications before installation. Via evaluation and statistical analysis, the assessment method can identify a part of malicious applications, mainly for those that apply for far too much permission or have relatively large differences with other applications which belong to the same category.

However, the assessment method is a nondeterministic approach and to a certain extent, the accuracy of the assessment method is on the influence of the Sample Set S. Simultaneously, The applications in the third-party Android market are not classified directly according to the functionality of applications, for example, The applications in Category E-book and Category Humanity have a great deal of similarity in fundamental functionality.

In our future work we plan to improve the Sample Set and then consider adjusting the classification of Android application by combination or division. Category Game is not included in the testing of the assessment method due to the complexity, finding the way to assess the applications belong to Category Game is also listed in the future works of us.

Acknowledgements

Supported by HGJ（Grant No.2012ZX01039004-008）and "the Fundamental Research Funds for the Central Universities".

References

[1] William Enck, M. O., and Patrick McDaniel (2009). "Understanding Android Security." IEEE: 8.

[2] Davi, L., A. Dmitrienko, et al. (2011). Privilege Escalation Attacks on Android Information Security. M. Burmester, G. Tsudik, S. Magliveras and I. Ilic, Springer Berlin / Heidelberg. 6531: 346-360.

[3] Juniper Networks. 2011. Mobile Malware Development Continues To Rise, Android leads The Way. http://globalthreatcenter.com/?p=2492

[4] William Enck, D. O., Patrick McDaniel, and Swarat Chaudhuri (2010). A Study of Android Application Security. In Proceedings of the 20th USENIX Security Symposium.

[5] ONGTANG, M., MCLAUGHLIN, S., ENCK, W., AND ACDANIEL, P. Semantically Rich Application-Centric Security in Android. In Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC) (2009).

[6] Asaf Shabtai, Y. F., Uri Kanonov, Yuval Elovici, Shlomi Dolev, and Chanan Glezer (2010). "Google Android A Comprehensive Security Assessment." Mobile Device Security.

[7] Machigar Ongtang, K. B., Patrick McDaniel (2010). "Porscha: Policy Oriented Secure Content Handling in Android." ACM.

[8] Enck, W., M. Ongtang, et al. (2009). "On Lightweight Mobile Phone Application Certification." Ccs'09: Proceedings of the 16th Acm Conference on Computer and Communications Security: 235-245.

[9] Nauman, M. (2010). "Apex: Extending Android Permission Model and Enforcement with User-defined Runtime Constraints." 10 Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security.

[10] Felt, A. P., Chin, E., Hanna, S., Song, D., Wagner, D. (2011). Android Permissions Demystified. In Proceddings of the 18th ACM conference on Computer and communications security(CCS '11).

[11] Delac, G., M. Silic, et al. (2011). Emerging security threats for mobile platforms. MIPRO, 2011 Proceedings of the 34th International Convention.

[12] Tanahashi, I. R. a. Y. (2011). "Various Approaches in Analyzing Android Applications with its Permission-Based Security Models." IEEE: 6.

[13] Sandeep Tata, Jignesh M. Patel. (2007). “Estimating the selectivity of tf-idf based cosine similarity predicates”. ACM SIGMOD Record: 7-12.

[14] Soucy, P.; Mineau, G.W. (2001) . “A simple KNN algorithm for text categorization.” IEEE.