Oct 27th, 3:00 PM - 4:30 PM

Permission-based Privacy Analysis for Android Applications

Erza Gashi
University for Business and Technology, eg32521@ubt-uni.net

Zhilbert Tafa
University for Business and Technology, zhilbert.tafa@ubt-uni.net

Permission-based Privacy Analysis for Android Applications

Erza Gashi[1], Zhilbert Tafa[2]

[1,2]UBT – Higher Education Institution
Prishtina, Kosovo
{eg32521[1], zhilbert.tafa[2]}@ubt-uni.net

Abstract. While Information and Communication Technology (ICT) trends are moving towards the Internet of Things (IoT), mobile applications are becoming more and more popular. Mostly due to their pervasiveness and the level of interaction with the users, along with the great number of advantages, the mobile applications bring up a great number of privacy related issues as well. These platforms can gather our very sensitive private data by only granting them a list of permissions during the installation process. Additionally, most of the users can find it difficult, or even useless, to analyze system permissions. Thus, their guess of app's safety mostly relies on the features like rating and popularity, rather than in understanding context of listed permissions. In this paper we investigate the relationship between the features collected from Android Market API 23 (such as Popularity, Total Number of Permissions, Number of Dangerous Permissions, Rating and Package Size) to app's privacy violation. To show the influence of each feature we use linear regression and R squared statistics. The conducted research can contribute to the classification of mobile applications with regards to the threat on user's privacy.

Keywords: android, applications, permission, privacy.

Introduction

With the increased number Mobile Applications, privacy has become a major threat for smartphone users. Two main market stores, that share more than 90% of market, are Android and IOS [1]. And, as the number of users increase, the privacy and security threats become more serious and dangerous.

Today, there are lots of free mobile apps in Android official Market that are used for advertisement and similar purposes, but these applications can also be used for personal data identification.

Permission control is one of the major Android privacy/security mechanisms. When an application is to be installed, a user has a choice whether to allow specific permissions or not. One problem is that most of the users are not informed about the permission system and the way permission can be misused. On the other hand, even if a user would be informed about the permission system, the user's denial of permissions would disable the application installation. This implies that the user does not actually have much control over the permission system.

Various applications use much more permissions than needed. Some of these permissions are recognized as more dangerous, which categorizes them as being dangerous in the privacy/security sense. The study presented in this paper aims to identify the relation between the user perceptions and this category of applications.

The rest of this paper is organized as follows. Section 2 gives some background in related permission-based privacy analysis from literature. Section 3 introduces the methodology and the dataset structure. The experiment and the evaluations are described in section 4, while section 5 concludes the paper.

Background and Motivation

A great amount of work is done in analyzing smartphone apps in the sense of privacy leakage.
Before installing an application from Google Play store, a user is presented with a list of permissions and a short description for each permission. These permissions cannot be changed once they are declared by app developer in manifest file without install updates. In addition, descriptions of each permission inform a user with functionalities and resources an app wants to access in order to perform as intended. Thus, in order for a user to be able to use downloaded application, he or she must grant all requested permissions to the app. The list of permissions has become longer with later versions of Android. In [2] authors show that Dangerous permissions tend to increase over a course of 3 years. They study a dataset of 237 apps with 1,703 versions collected from Google Play Market API 3 to 15 and conclude that apps tend to use of more permissions over time.
In studies about android permission models, authors in [3]and [4], show that permission warnings do not help users make proper decisions. In trying to identify the granularity of expression for permission descriptions, Barrera et.al analyzed 1,100 Android applications and presented a permission-based security model that improves expressiveness without increasing number of requested permissions.
The results from the study [4] that surveys 308 Android users indicate that users pay little attention to permissions during the installation process. Consequently, warnings about permissions despite the expression level do not help a typical user choose between safe and a potential dangerous app.
With introduction of Permission Manager (App Ops) in Android 4.3, users are offered with some type of control over permission selection by enabling them to choose whether a specific permission is tolerable by a user. As a consequence, such control over permissions comes with a trade-off on apps functionality. Similar extensions to offer users with a finer-grained control over permissions are proposed in [5], [6], [7], [8]. For instance, MockDroid Android simulator application [5] allows users to override access of specific properties at startup time and help them better understand the trade-off between functionality and exposure of personal sensitive data. This extension type application provides support of mocking couple of permissions. However, it is limited to only mocking five types of permissions and their functionalities. TISSA [7] is yet another Android application that allows users to customize privacy setting for untrusted apps by deselecting specific dangerous permissions. TISSA bases app trustworthiness by evaluating permissions found in applications that are known to be leaking private information. The application data set is extracted from TraintDroid [6] application, which aims to inform users about misbehaving applications by monitoring sensitive data flow through different sources.
As reported in [9], today a typical smartphone user has 80 installed apps in average. Customizing all permission settings for these apps is frustrating and time consuming. Thus, our goal is to study what visible app characteristics during installation process can reveal information about its privacy level. Our work complements prior work in the area of identifying relationship of permissions and app popularity. It might be most similar to the study [10], which uses community ratings in app markets to identify indicators that reveal privacy risk level of an application. They show strong correlation between popularity of an application and the number of ratings an application has gained. Our study is different in a sense that we check

relationship between the values of apps rating with number of downloads and package size with number of permission. Indicators that we test are most apparent that a typical user experiences during installation process.
Finally, different studies were conducted in efforts to define techniques to predict dangerous applications in terms of permissions they use. In [11], the authors study privacy preferences by looking for patterns of permissions requests in Android and Facebook applications. They use matrix factorization technique and were able to identify 30 common patterns of permission requests. Similarly, our work aims to derive a model by setting three conditions based on previous work.

Methodology and materials

We detail the methodology in the following section and divide this process into Data Collection and Data Processing.

Data Collection

The data set used for this paper is crawled from the official Android market (Google Play) in March 2017. We created index of 1110 apps that were visible to users in Kosovo. More specifically, users of Vala Mobile Network Operator who actually use the Monaco country code +377. This information impacts our dataset in different ways. As in [12], it is known that developers may restrict their apps in variety of ways such as phone compatibility, location and Android version. For example, some apps are only marked compatible with some types of phones or tablets, some are limited to certain countries (e.g. PokemonGo was only available in US for a long time) and some apps require a minimum version of Android.
Because Google has restrictions on the number of purchases with a single credit card [13], we only crawled and analyzed free apps in this paper. Same methodology can be applied in collecting and analyzing paid apps as well.
We customized a crawler with the help of libraries in [14] in order to automate the process of data collection. The whole process of data collection, preprocessing, storing and analysis in chronological order can be seen in Figure 1.

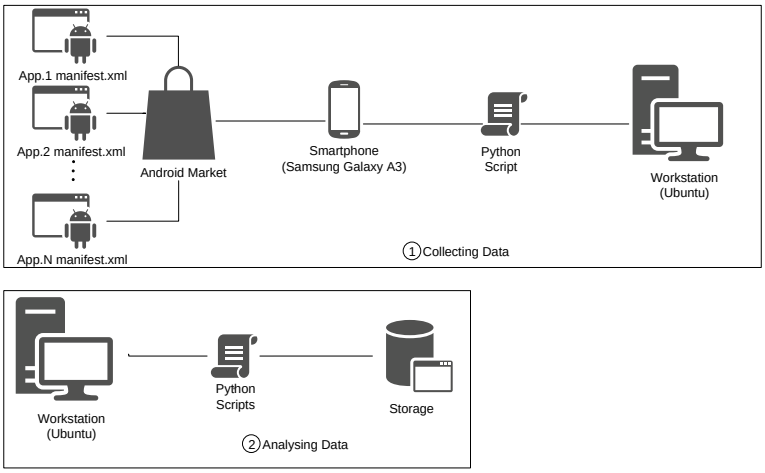

Fig.1 Process of data collection, preprocessing, storing and analysis.

Android version Marshmallow 6.0.1 uses the latest API 23. We accessed and parsed Manifest.xml file of each application through python scripts. This was done without downloading application files locally. We obtained the metadata information of the apps and settings from XML files, including:

Title: is a title given to the app by the developer / creator
Package name: identifies uniquely the application itself.
Version: the current version of application package.
Downloads: number of times one application is downloaded.
Rating: in a scale 1 to 5, users rate an app. It gives the average score.
File size: describes how many MB an application package is.
Rating Count: number of Google users that rated the app.
Creator: name of creator / developer or developing company.

Another restriction faced during data collection is Google security mechanism which does not allow the download of multiple applications at a time from the same IP address and Google account. This made us set a *sleep()* method in our crawler which reduced the performance with regards to the time it takes to get metadata by *30 seconds * time_to_get_data_for_each_app*. The process of collecting the respective information and python files used for each step is given in Figure 2.

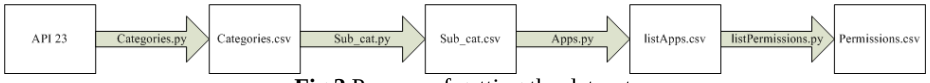

Fig.2 Process of getting the dataset.

A snippet of output after executing *permissionsList.py* script can be seen below. As it can be noted, each line includes the name of application, package name and permission type. This information is stored in a specific file for analysis part.

A snippetfrom file *Permission.csv*.

```
Headspace - meditation;com.getsomeheadspace.android;android.permission.ACCESS_COARSE_LOCATION
Headspace - meditation;com.getsomeheadspace.android;android.permission.ACCESS_FINE_LOCATION
Headspace - meditation;com.getsomeheadspace.android;android.permission.ACCESS_NETWORK_STATE
Headspace - meditation;com.getsomeheadspace.android;android.permission.ACCESS_WIFI_STATE
Headspace - meditation;com.getsomeheadspace.android;android.permission.GET_ACCOUNTS
```

Data Processing and Analysis

In order to perform statistical analysis of the data, Python and associated developed packages are used to get the most accurate results.
After removing rows with missing data, the final file consists of 980 unique apps and counted 965 permissions, including third-party permissions. There are 34 distinct categories and top twenty downloaded applications among these categories are shown in the Table 1 and Table 2.

Table 2. Basic statistics of our dataset – Top downloaded apps (left) and Apps with highest number of permissions (right).

App name	#Downloads
WhatsApp Messenger	1000000000
Google+	1000000000
Google Play Newsstand	1000000000
Instagram	1000000000
Facebook	1000000000
Maps - Navigation & Transport	1000000000
Google Play Games	1000000000
Messenger	1000000000
Chrome Browser - Google	1000000000
Hangouts	1000000000
YouTube	1000000000
Google	1000000000
Google Photos	500000000
Twitter	500000000
Skype - free IM & video calls	500000000
Clean Master (Boost&Antivirus)	500000000
Waze - GPS\t Maps & Traffic	100000000
AliExpress Shopping App	100000000
WPS Office + PDF	100000000
eBay - Buy\t Sell & Save Money	100000000

App Name	#Permissions
RainbowCamera	106
Brightest Flashlight-Multi LED	79
Google	77
CM Security AppLock AntiVirus	70
Tap Emoji Keyboard	69
Flash Keyboard - Emojis & More	69
UC Browser - Fast Download	66
Messenger	61
Kindle	54
Facebook	53
GO Security - Antivirus AppLock	51
CM Launcher 3D-Theme Wallpaper	50
WhatsApp Messenger	50
360 Security - Antivirus Free	49
SOMA free video call and chat	48
Virus Cleaner - Antivirus	48
WhatsCall - Free Global Calls	46
Skype - free IM & video calls	45
video chat & free calls icq	45
Mi Live - Live video streaming	45

Table 3. Basic statistics of our dataset – Number of Apps per category (left) and Number of Permissions per category (right).

Category	#Apps
ENTERTAINMENT	45
TOOLS	44
VIDEO_PLAYERS	44
PHOTOGRAPHY	41
EDUCATION	39
BEAUTY	38
BOOKS_AND_REFERENCE	37
COMMUNICATION	37
PRODUCTIVITY	37
NEWS_AND_MAGAZINES	35
PERSONALIZATION	35
COMICS	34
GAME	34
MUSIC_AND_AUDIO	34
SPORTS	33
ART_AND_DESIGN	32
LIBRARIES_AND_DEMO	32
PARENTING	32
SOCIAL	32
TRAVEL_AND_LOCAL	32

Category	#Permission
TOOLS	180
PRODUCTIVITY	168
COMMUNICATION	162
ANDROID_WEAR	130
SOCIAL	115
TRAVEL_AND_LOCAL	111
NEWS_AND_MAGAZINES	109
SHOPPING	108
FINANCE	105
BUSINESS	102
VIDEO_PLAYERS	97
MAPS_AND_NAVIGATION	86
HOUSE_AND_HOME	86
HEALTH_AND_FITNESS	84
SPORTS	83
PARENTING	80
FOOD_AND_DRINK	80
WEATHER	78
ENTERTAINMENT	75
PERSONALIZATION	73

Estimating Privacy Risk

As initial condition we set the number of dangerous permissions being requested by an app. As in[2], 66.11% of permission on new versions of app contain of at least one or more Dangerous permission. On the other hand, our dataset has only 14.06% of apps which do not use any dangerous permission.
Second condition includes four-tuple dangerous permissions such as CAMERA; READ_CONTACTS, ACCESS_FINE_LOCATION, READ_PHONE_STATE. Despite the fact that the list of dangerous permissions might be longer [15], intuitively the ones above comprise the most delicate information. At the same time these are part of the most frequent permissions asked from malwares as per Zhou and Xiang study [16].
Third, permissions alone can potentially expose privacy vulnerabilities for users. Yet, they are more dangerous when combined with other, especially communication resource granting permissions. Enck et al. [17]identified a list of vulnerable combinations that can be very risky for the system. Hence, we combined our list from second condition with permission INTERNET. Naturally an app that requests dangerous permissions and has INTERNET access can potentially send information to unauthorized organizations and violate user's privacy.
We have forty-eight apps that satisfy our model based on set conditions.

Condition#1: App has more than one dangerous permissions
*Condition#2***:** Has set of permissions
list_of_privacy_p = ['android.permission.CAMERA',
'android.permission.READ_CONTACTS',
'android.permission.ACCESS_FINE_LOCATION',
'android.permission.READ_PHONE_STATE']
…
Condition#3: Internet use granting permission:
'android.permission.INTERNET'

Final dataset before applying the condition has fields as below.

```
ID,PackageName,Title,cat,Creator,SuperDev,VersionCode,SizeMB,Rat
ing,NumDownloads,TotalPermissions,Danger_P,Privacy_P,Safety

1,aerobicexercise.danceworkout,Aerobics workout weight
loss,HEALTH_AND_FITNESS,AppsBundle,0,1.00,5.00,4.17,1000.00,9.00
,1.00,1.00,0

2,air.com.KalromSystems.FruitDrawPlay,Fruit Draw: Sculpt
Vegetables,ART_AND_DESIGN,Kalrom Systems
LTD,0,1005006.00,28.90,3.99,100000.00,6.00,3.00,2.00,0
```

Experiments and Results

The experiment is focused on finding a correlation between apps and characteristics mentioned the previous section. Specifically, it aims to check for the possible relationships between apps rating, the number of downloads, and the package size on one side, and the number of permissions on the other side.

Correlation between Rating and Number of Downloads

Unlike [10], our experiments aim to show the relationship of apps rating and number of apps installations (downloads). Rating directly expresses the feedback of users regarding apps, sharing their personal experience about apps functionality or user interface. Therefore, they can help the developers to improve their apps and at the same time refer or not the app to new users. Thus, rating is supposed to be important in terms of users, and at the same time is a parameter that impacts the number of downloads. Hence, we suggest that apps with better rating should have larger number of downloads.

InFigure 3a, the experiment involves the whole apps of our dataset. We can see that the rating of apps is distributed around score 4, 3.68 – 4.7. As it can be noted, most of the applications have a high rating value. On the other hand, results show that most of the apps are downloaded less than 100 thousand times.

Figure 3b shows same variables and experiment conducted parallel for both safe and unsafe apps. Same as in Figure 4a, even apps that are potential dangerous have high rating value and downloaded around 100 thousand times. Thus, we can conclude that there is no particular pattern on dangerous apps.

With regards to correlation, one can see from Figure 3b that rating score and number of downloads have a very weak positive correlation value. Yet, our results suggest that high value rated apps are potentially more dangerous than low value rated apps.

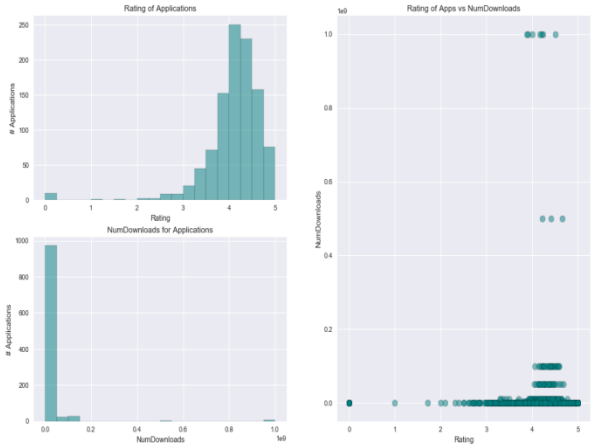

Fig.3a Correlation between Rating and Number of Downloads whole dataset.

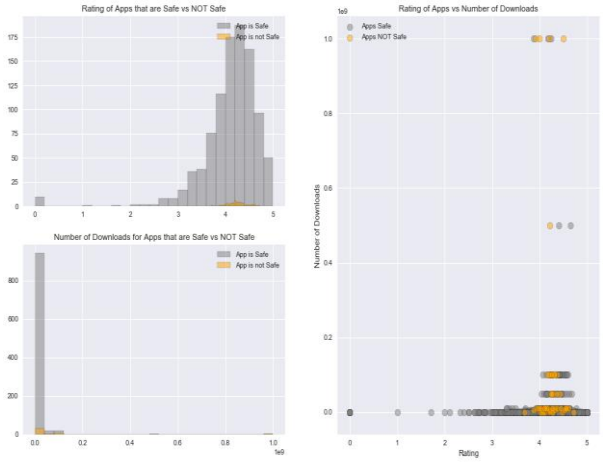

Fig.3b Correlation between Rating and Number of Downloads on Safe/Unsafe apps.

Correlation between Package Size and Number of permissions

In addition to rating value and number of downloads, package size is another very obvious parameter to the user during app installation process. Thus, we try to reveal possible patterns on relationship between package's MB and the number of permissions an app requests.

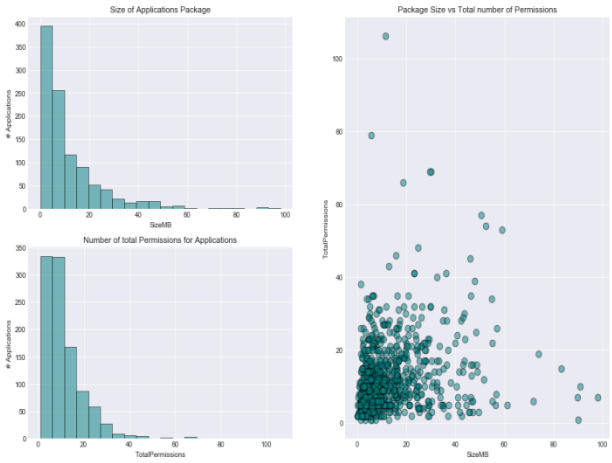

Fig. 4a Correlation between Package Size and Number of permissions.

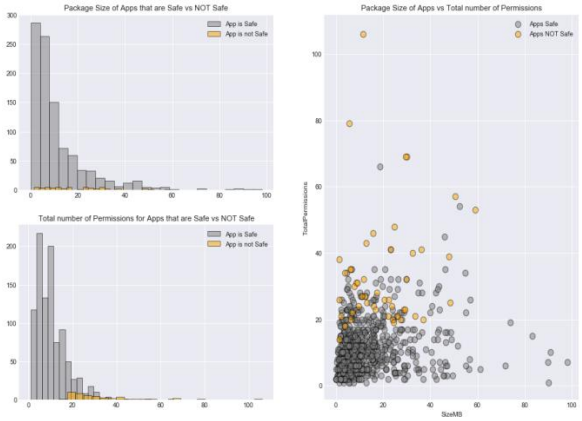

Fig. 4b Correlation between Package Size and Number of permissions on Safe/Unsafe apps.

Similar to the fist experiment, one can see from the Figure C that there are two accomplished tasks: a) relationship between *Package*and *Number of Permissions*for the whole dataset, b) relationship between *Package Size*and *Number of Permissions* for Safe/Unsafe apps.
The observations on this experiment show that most of apps have less than 10 MB size, while they contain 10 permissions in average.
In addition we suggest that package size is not a significant predictors on app's number of permissions, and consequently, not significant on safety.

Conclusions

In order to understand whether there is a relation between number of permissions per category and number of applications per category, the number of permissions per each category were extracted and counted only the distinct ones. Surprisingly, the hypothesis is not plausible. Despite some categories have a very large number of applications; they are not ranked among categories that have a very large number of permissions.
In addition, one can expect that applications that use more permission are the ones that most violate user's privacy. Thus, we have chosen to focus and analyze these types of applications in more depth. For example, the so-called application *Rainbow Camera* has the largest number of permissions while, for example, there is no reason for the camera to use the permissions such as Internet access to work properly.
Conducting above experiments we tried to find a pattern of danger apps using rating, number of downloads, package size and number of permissions. As result, we report that high rated apps are potentially more dangerous and that package size cannot be considered as significant variable on apps safety.
Finally, popularity used in Android Market applications is not a reliable indicator of privacy risks. Therefore, user cannot rely on features like number of downloads and rating to decide if an app violates the least privileged principal.

References

1 Egham, "Gartner," 15 February 2017. [Online]. Available:

https://www.gartner.com/newsroom/id/3609817. [Accessed August 2017].

2 X. Wei, L. Gomez, I. Neamtiu and M. Faloutsos, "Permission evolution in the Android ecosystem," in *28th Annual Computer Security Applications Conference*, Orlando, 2012.

3 D. Barrera, G. H. Kayacik, P. C. van Oorschot and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android," *17th ACM conference on Computer and communications security*, pp. 73-84, 04 - 08 October 2010.

4 A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin and W. David, "Android permissions: user attention, comprehension, and behavior," Washington, D.C, 2012.

5 A. R. Beresford, A. Rice, N. Skehin and R. Sohan, "MockDroid: trading privacy for application functionality on smartphones," *12th Workshop on Mobile Computing Systems and Applications*, pp. 49-54, 01-02 March 2011.

6 W. Enck, P. Gilbert and B. G. Chun, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy," Vancouver, 2010.

7 Y. Zhou, X. Zhang, X. Jiang and V. W., "Taming information-stealing smartphone applications (on Android)," Pittsburgh, 2011.

8 J. Jinseong, K. K. Micinski, J. A. Vaughan, A. Fogel, N. Reddy, J. S. Foster and T. Millstein, "Dr. Android and Mr. Hide: fine-grained permissions in android applications," Raleigh, 2012.

9 App Annie, "App Annie," 2017. [Online]. Available: http://go.appannie.com/1705UsageReport_LP01-RegisterUpdated.html?utm_source=mma-uk&utm_medium=partnership&utm_campaign=emea-logo-201705-1705-app-usage-report-push-by-mma-uk&utm_content=report-1705-app-usage-report&sfdcid=7016F000001AXUJ. [Accessed September 2017].

10 P. H. Chia, Y. Yamamoto and N. Asokan, "Is this app safe?: a large scale study on application permissions and risk signals," in *21st international conference on World Wide Web*, Lyon, France, 2012.

11 M. Frank, B. Dong, A. P. Felt and D. Song, "Mining Permission Request Patterns from Android and Facebook Applications," in *IEEE 12th International Conference on Data Mining*, Brussels, Belgium, 2012.

12 Developers. [Online]. Available: https://developer.android.com/google/play/filters.html.

13 J. Lin, B. Liu, N. Sadeh and S. I. Hong, "Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings," in *Proceedings of the Tenth Symposium On Usable Privacy and Security (SOUPS 2014).*, Menlo Park, 2014.

14 dflower, "GitHub. Inc," June 2014. [Online]. Available: https://github.com/dflower/google-play-crawler. [Accessed March 2016].

15 Developers, "Developers," Developers, [Online]. Available: https://developer.android.com/guide/topics/permissions/requesting.html#normal-dangerous. [Accessed May 2017].

16 Z. Yajin and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," in *Trust and Trustworthy Computing*, San Francisco, 2012.

17 W. Enck, D. Octeau, P. McDaniel and S. Chaudhuri, "A Study of Android Application Security," San Francisco, 2011.