INTRODUCTION

1) Context

Nowadays, Android is one of the mobile market leaders, giving more than a million applications on Google Play store. Current the year 2015, in the average, 135 millions of applications per day were installed by Android users. Android applications can be written by any developer and do not need any certification or validation before being made available on the store. Thus, abusive Android applications are available on Google Play [1, 2, 3, 4, 5]. These applications collect user's data and require access to sensitive services not related to their functionalities. To solve this security problem, Google uses a permission system. The permission system method design a list of permission required by each Android application for its installation. The problem is that the permission list is most often ignored by end-users. The fact that the permission list may provide information about the behavior of the application gives the opportunity to automatize the analysis of the applications. The exploitation of this issue may conducted to the classification or the categorization of the Android applications. For this purpose, strong key permission and expected permission requests may be well identified. This work use graph modeling to identify the normal behavior of application and the expected permission requests in order to classify the applications by category. So, each application is assigned to the group (category) which has the same graph pattern.

2) Background

The background of this work concerns Android permission system, application analysis, malware analysis and detection, mobile decision support systems and permission verification tools.

Android permission system

En Android device has permissions and permission verification system that are installed into the Android operating system. Due to the fact that applications has limited right, developer need to add permissions list which allow application to access to sensitive data. An example case of the permission that allow access to Internet is **android.permission.INTERNET**. The permissions recorded throughout Google nexus are 299. This permission can be classified to three categories: normal permission, development permissions and dangerous permissions.

Application analysis

The analysis of the permissions list helps to identify the relation between applications and permissions list and, to detect abusive permissions.

This analysis can be made using Self-Organizing Map (SOM). This was done by Barrera et al. in 2009. Their results show that applications in the same category do not necessarily require the same permissions due to different functionalities. The work done by Chia et al. show that abusive applications exist on the Android market.

Permission-based decision support systems

The aim of this tools is to design a decisional system that provide privacy and risk recommendations to users. This can be made using crowd-sourcing system.

Malware applications analysis

This part focus on malware detection. That is to say, the method or approach used to detect malware. machine learning can by use as approach for this purpose.

Permission verification tools

This kit give to the developers the idea about list of permission necessary or unnecessary to develop.

3) problem and research question

The problem is based on the following questions **Do Android applications of different** categories **require different permission patterns and can be distinguished by patterns? Can a category pattern allow to measure an application risk/privacy level and permit malware detection?**

4) Related works

Reference	Benign application analysis	result
13	Top 50 free Android applications of 2009	Application in the same category do not necessarily require the same permissions
4	most popular and new Android applications	Popular applications requested more permissions than news application
12	unsupervised learning	different permission requests for distinct categories
Reference	**Permission-based decision support systems**	**Usable fot Android applications**
22	crowd-sourcing system	No
23	searching for a justification of permission usage in application descriptions with NLP	Not verified for all
25	risk warning system based on the occurrence of 24 permissions (identified by author as dangerous	Yes
Reference	**Malware applications analysis**	**Tools used**
15	supervised machine learning techniques to categorize 820 applications into 7 categories	Term Frequency (TF) and Inverse Document Frequency (IDF); Bayesian Tree Augmented Naive and Random Forest Algorithms, K-Nearest Neighbor (KNN) methods
16	classified applications into two main categories : games and tools	Chi-Square (CS), Fisher Score (FS) and Information Gain (IG) methods
17	analysed the use of the permissions in a set of 1,227 clean and 49 families of malicious Android applications	Andrubis
18	Extracted permissions and API calls from benign and malicious applications	SVM, RaJ48 Decision Tree and Bagging algorithms
Reference	Permission verification tools	
27	tool informing a developer about unnecessary permissions	different behaviours between benign and malicious applications in permissions usage
5	tool called Permlyzer	
2 3 28 29 30 31,32 33	TaintDroid AppFence SCAndroid AndroidLeaks ScanDal Blue Seal FlowDroid	monitor and fully control data flow in application but not clear if the user is able and willing to adopt those technologies

5) Methodology employed

This paper present the solution of the problem in three steps.

- The first step is the construction of a category pattern to identify the permissions expected for a given application (see figure 1).

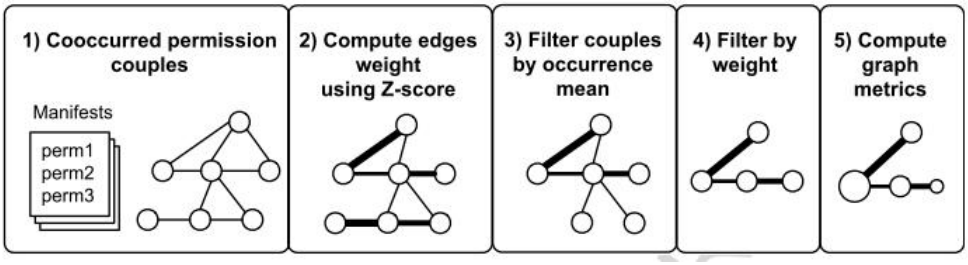

Figure 1: Five steps of pattern construction methodology

- The second step is Application classification to highlight the most relevant patterns. By using different list of permissions this step helps to identify the most pattern which is close to a given application.
- The third step is the risk metrics that aim to detect abusive applications. The overall objective is to provide a warning system that remains within the proposed patterns and risk metrics.

6) Results founds and perspectives

As the main objective of this work is to assign the category to each application based on the permission list using graph modeling, the following results were found:

i. **Personalization category**: Due to the fact this category involves wallpaper, the most permission list use to assign applications to this category is **Set_Wallpaper**. This is follow by the **get_tasks**, the **System_alert_window** and **in_context.**

i. **Health and fitness category**: the permissions use to classify most applications in this category are **Bluetooth** and **bluetooth_admin** . The most applications use here are an **accelerometer, heart and step monitor**, etc.

ii. **Transportation category:** As this category involves map and GPS, the most permission used to classify the applications here is **Access_fine_location.** This is follow by **Access_mock_location** and **Bluetooth_admins.**

iii. **Photography category**: Camera is the main subject in the category. So, the list of permission used are **camera – possibility** and a **device-embedded**. The **set_wallpaper** permission is also used.

7) Future woks

This work can be continued by introducing an application recommendation system that not only takes into account application ratings, but also privacy, using a normal permissions request pattern.

8) Most important publication for this work

- D. Barrera, H. G. Kayacik, P. C. van Oorschot, A. Somayaji, A methodology for empirical analysis of permission-based security models and its application to android, in: Proceedings of the 17th ACM conference on Computer and communications security, CCS ’10, ACM, 2010, pp. 73–84.
- P. H. Chia, Y. Yamamoto, N. Asokan, Is this app safe?: a large scale study on application permissions and risk signals, in: Proceedings of the 21st international conference on World Wide Web, WWW ’12, ACM, 2012, pp. 311–320.