Decision Support Systems 93 (2017) 62–76

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Android application classification and anomaly detection with graph-based permission patterns

Karina Sokolova[a,*], Charles Perez[b], Marc Lemercier[a]

[a] University of Technology of Troyes, 12 rue Marie Curie, Troyes, France
[b] PSB Paris School of Business, Chair Digital Data Design, 59 rue nationale, Paris, France

ARTICLE INFO

Article history:
Received 2 July 2015
Received in revised form 12 August 2016
Accepted 10 September 2016
Available online 20 September 2016

Keywords:
Android
Permission patterns
Classification
Anomaly detection
Risk warning
Graph analysis

ABSTRACT

Android is one of the mobile market leaders, offering more than a million applications on Google Play store. Google checks the application for known malware, but applications abusively collecting users' data and requiring access to sensitive services not related to functionalities are still present on the market. A permission system is a user-centric security solution against abusive applications and malware that has been unsuccessful: users are incapable of understanding and judging the permissions required by each application and often ignore on-installation warnings. State-of-the-art shows that the current permission system is inappropriate for end-users. However, Android permission lists do provide information about the application's behavior and may be suitable for automatic application analysis. Identifying key permissions for functionalities and expected permission requests can help leverage abnormal application behavior and provide a simpler risk warning for users. Applications with similar functionalities are grouped into categories on Google Play and this work therefore analyzes permission requests by category.

In this study, we propose a methodology to characterize normal behavior for each category of applications, highlighting expected permission requests. The co-required permissions are modeled as a graph and the category patterns and central permissions are obtained using graph analysis metrics. The obtained patterns are evaluated by the performance of the application classification into categories that allow choosing the best graph metrics representing categories. Finally, this study proposes a privacy score and a risk warning threshold based on the best metrics. The efficiency of the proposed methodology was tested on a set of 9512 applications collected from Google Play and a set of malware.

1. Introduction

Mobile applications are extensively used worldwide and new applications are added every day to mobile markets — platforms for the distribution of mobile applications. Recently there has been a spate of interest in Android — one of the market leaders that offers more than 1.5 million applications on its official store named Google Play (June 2015). In the average, 135 millions of applications per day were installed by Android users in 2015.

Android applications can be written by any developer and do not require any certification or validation before being made available on the store. As a result, poor, malicious or simply abusive applications coexist with benign Android applications on Google Play [1–5]. Google Bouncer (antivirus system proposed by Google) now checks applications for malicious code, but no other validation process takes place for applications arriving to the store.

Google Play provides certain information helping users to decide which application to install: screenshots provide a graphical user interface overview; users' ratings and comments reflect the stability and usability of an application. Although users are looking for appealing and useful applications with no bugs, they should take into account other factors before installing an application such as security and privacy and particularly in the context of BYOD (Bring Your Own Device). Furthermore, previous studies, such as [6], show that users are concerned about privacy and security and are willing to have applications with greater respect for privacy.

One of the Android platform security mechanisms and a principle user warning system of Android is the permission system. Each application has very limited capabilities by default, and needs to require permissions to access sensitive data or services. Users are prompted with a list of the permissions required by an application just before the installation. This list is supposed to warn users about hazardous and abusive applications, but, unfortunately, permission lists have

* Corresponding author.
E-mail addresses: karina.sokolova@utt.fr (K. Sokolova), c.perez@psbedu.paris (C. Perez), marc.lemercier@utt.fr (M. Lemercier).

http://dx.doi.org/10.1016/j.dss.2016.09.006
0167-9236/

been shown to be ineffective for this purpose. First, users see permission lists as a repeated warning or a license agreement that must be accepted to obtain a service. Permission lists are only shown in the final step before installation when other criteria for the user's decision have been met, and therefore the permission list is considered an obligation rather than a decision factor [6,7]. Second, users often do not have enough background to understand the meaning of permissions and their possible harm. Third, permissions are shown entirely out of the context, which prevents the user from understanding their purposes. Finally, some permissions are so frequently required that users do not pay any attention to them [8,9].

It can be seen that there is at present no system that helps Android users to take a decision aimed at more privacy-respecting and secure Android applications. Users must either rely on the community with comments and ratings, which rarely refer to possible security problems, or manually verify permissions and rely on their personal knowledge and understanding. Authors of previous studies, such as [10], highlight the need for a new security and privacy indicators for mobile users.

In spite of ineffectiveness of permission list warnings, the Android permission system seems a valuable source of information: 80 permissions are available to third-party applications, and this number doubles for system applications; more permissions appear with each new Android version. Information about required permissions is embedded into each application and is always available.

Instead of asking users to verify permission lists manually an automatic analysis can be used to detect expected permissions and anomalies. With less repetitive warnings and easier indicators, users would be able to use permissions as a decision factor.

Many studies analyzes permission usage of Android application but limit their studies to the detection of correlation between permission requests and other application attributes, such as price and rating [4,11-13]. Many studies focus on Android malware detection by defining malware-specific behavior [14–19]. The detection, thereby, is limited to the known malware; unknown malware and applications that abusively require permissions would stay undetected. Also, most of the previous studies on Android application risk detection provide a binary vision of an application security: normal or risky. In our knowledge, only two studies provide solutions helping users to judge and to compare security and privacy levels of Android applications [20,21]. While existing studies have established ranking systems for Android applications regarding permission requests, multiple limits of the proposed methodologies should be addressed. The methodologies proposed by the authors are based on the assumption that the risk of an application would increase with the number of permissions used. However, many malicious applications use very few permissions and some very popular and functional applications may request many permissions. Also, the final score of one application in the proposed studies depends on all other applications in the dataset and all scores must be recalculated when one application is added or deleted from the dataset. It is not clear if such methodology can be scaled for constantly evolving Google Play.

The purpose of our study is to propose a new methodology that evaluates the risk of a given Android application and detects abnormal applications. We propose to calculate the risk of an application based on the proximity of its permission request with a pre-calculated normal behavior. Therefore, even if an application requests few permissions but they deviate from the expected request in the given category, it would be considered abnormal. This paper describes three phases of our research:

First, we analyze a large set of applications collected from Google Play. For each category, we build a graph of 'normal' permission requests and compute graph metrics obtaining behavioral patterns.

Second, we verify which of the obtained patterns characterize categories best. We build pattern-related features and apply machine learning algorithms to classify applications into categories. The patterns containing betweenness centrality and weighted degree metrics showed better classification performances than others. Therefore, we conclude that those patterns are the most descriptive and representative for categories.

Finally, we propose to measure a privacy level of an application regarding its category and the previously obtained patterns. We suppose that normal applications will request permissions following the pattern, and applications that deviate from the pattern would more likely be abnormal: wrongly categorized, abusive or even malicious. We propose to warn users of those abnormal requests and define a threshold permitting to separate normal and abnormal applications. We evaluate the performance of proposed warning system in malware detection by injecting malicious applications into the initial dataset. Our method outperform the most recent and relevant work on Android application risk evaluation [21].

By this study, we test and verify multiple hypothesis:

1. Application category contains similar applications that would use similar permissions. Therefore, an average or 'normal' category application could be represented by a permission pattern.
2. Different application categories contain different applications. Therefore, the patterns that characterize one category should differ from the pattern characterizing another category. In this case, patterns should permit to identify the category of an application by permissions this application requires.
3. A pattern characterizing 'normal' applications of a category should permit to measure the risk level of an application and to detect abnormal applications: applications abusively requiring permissions, bad-quality applications, applications from wrong categories and malware. By hypothesis, the more applications request permissions that are not normally observed in the category, the higher is its risk score.

The following research questions emerge from the highlighted hypothesis and are answered by this work:

- 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?

The remainder of the paper is organized as follows. Section 2 presents the actual Android permission system and studies related to permissions requests analysis, permissions verification, mobile application risk evaluation and malware detection. Section 3 presents the dataset used for the evaluation of the study and our methodology for permission patterns construction, privacy score computing and warning threshold evaluation. Section 3 also presents the methodology we used to evaluate the obtained patterns. Section 4 presents the results and performances of our approach with respect to the state-of-the-art. Section 5 discusses the results and considers future works. The paper ends with a conclusion.

2. Background and related works

In this section, we provide the background to the Android permission system and outline the state-of-the-art related to our work such as benign application analysis, malware analysis and detection, mobile decision support systems and permission verification tools.

2.1. Android permission system

The Android permissions and permission verification system are embedded into the Android operating system. By default, applications have limited rights and whenever an application