ReviewMiner: Malware Detection of Apps Using Review Processing

Sarat chandran N[1], Rahul MV[2], Elizabeth Isaac[3]

[1,2] Dept. of Computer Science and Engineering Engineering, MACE, Kerala, India
[3] Assistant Professor, Dept. of Computer Science and Engineering Engineering, MACE, Kerala, India

---

Abstract - *Misguiding reviews and malpractices in Google play store, results in proliferation of fraud and malware apps. The user usually relies on star ratings before installing an app. The Mass rating can sometimes give a fraudulent app, a positive outlook. To identify malware, previous works focused on app executable data and permission analysis. ReviewMiner is a novel system that identifies traces left behind by fraudsters and thus detects malware apps based on text processing of reviews. Sentimental analysis of collected reviews distinguishes between fake and genuine reviews. ReviewMiner combines detected review relations with lingual and behavioural signals gathered from Google play app data to identify suspicious apps. It also reveals coercive reviews campaign in which users are harassed into writing positive reviews and install and review other apps. In addition to this, it also checks whether the permission policy of a particular app violates its genuine requirements.*

***Key Words*: Android store, reviews, malware detection**

1. INTRODUCTION

Android today is the largest installed base of any mobile platform and is rapidly growing. Every day more than a thousand new Android devices are activated worldwide and start looking for apps, games and other digital content. A world-class platform is provided by android that can be used for creating apps and games for Android users all over the world, and an open marketplace for distribution. With contributions from the open-source Linux community and at least 200 hardware, software and carrier partners, Android has emerged as the fastest-growing mobile operating system. Its openness has made it a favourite for consumers and developers alike, boosting app consumption. Android users download billions of apps and games each month from Google Play[1] – the premiere marketplace for selling and distributing Android apps. When one publishes an app on Google Play one automatically reaches the huge installed base of Android.

With millions of Android apps to download from Google play store, it is understandable that the user might feel a bit confused as to what to download and install. To help in these situations, a number of bloggers and sites have assumed the role of advisers, providing reviews on apps and scoring or ranking them. The commercial success of apps in marketplaces like Google Play and the incentive model these markets offer to popular apps have caused these markets to attract fraudulent behaviours. This includes posting fake reviews[2] and false installation counts to deceptively boost the search rank and popularity of apps which could ultimately translate into financial benefits. Some developers use app markets as a launch pad for malicious software which leads to expedited malware proliferation [3],[4],[5]. One method of such fraudulent activity is to hire teams of willing workers who commit fraud collectively, emulating realistic and spontaneous activities from unrelated people. This behaviour i.e "crowdturfing" [6] is generally termed as search rank fraud.

Continuous efforts have been made by the Android market to identify and remove malware. However these have not been completely successful. To prevent malicious apps from becoming a part of the official Android app store (Google Play), Google introduced a security service- Bouncer[7]. It automatically scans developer accounts, both new and previously uploaded apps in Google Play with its reputation engine and cloud infrastructure. Similar tools are available. Most of these tools for mobile malware detection focus on static analysis of code and permissions and the dynamic analysis of app executables. But recent study showed that malware evolved quickly to bypass these anti-virus tools [8]. This project aims at identification of malware apps in Google Play Store based on text processing of reviews and permission access policies. The mass positive reviews given to a particular app boosts its positive outlook thereby prompting user to install it.It has been observed that such fraudulent behaviours leave behind a number of telltale signs. For example, the high cost of setting up of more than one valid Google Play accounts results in fraudsters reusing their accounts during their review writing jobs, resulting in more apps bearing reviews from same user accounts. Also when reviews are examined one can sometimes see the unpleasant experiences reported by legitimate users affected by the app malware. One indication of safe to malware (Jekyll-Hyde) transition is the increased number of requested permissions from one version to the next, also referred to as "permission ramps". Observations like these have been used for this project proposing the system ReviewMiner, to detect Google Play fraud and malware.

ReviewMiner downloads all the reviews of a particular app which is to be identified as fake or genuine. This is done by the use of unique id of each app provided by google. It is followed by sentimental analysis of these reviews. This process classifies the reviews as positive, neutral and negative and also assigns a numeric score as 1, 0, and 1 respectively. Each review is given a star rating from 1-5 based on this processing of reviews. Even though this rating is similar to the star rating in google play store, it is more reliable since it is based on genuineness of reviews. Star rating of 1-5 for each review is given as in fig 3 and in the final display, the count of each star rating is provided. For user benefit the tool also displays the same in the form of a histogram, with the star rating along the X-axis and the

number of reviews indicating that particular star rating along the Y-axis.

ReviewMiner also checks whether permission polices of a particular app violates its genuine requirements. For instance, if a picture editor app has permission requirements such as access to audio, it is a violation of its genuine permission requirements because a picture editor app doesn't need the use of audio. This feature is also taken into consideration for overall malware detection. Fig 1 shows the general permission requirements of apps.

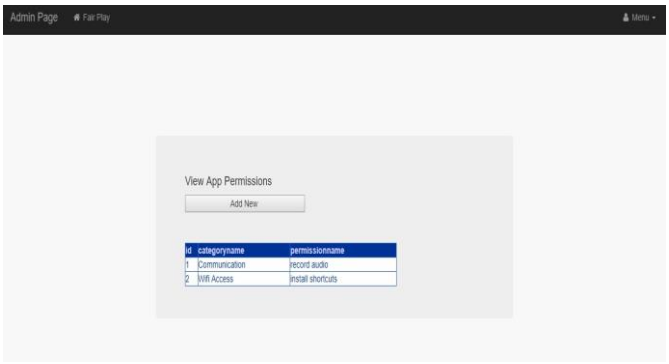

Fig -1: General Permission requirements of Google Play Apps

2. HIGHLIGHTS OF REVIEWMINER

The ReviewMiner system generates relational, behavioural and linguistic features to train supervised learning algorithms. The system also uses temporal dimensions of review post times to identify suspicious review spikes received by apps. It identifies apps with unbalanced review, rating and install counts, as well as apps with permission request ramps.

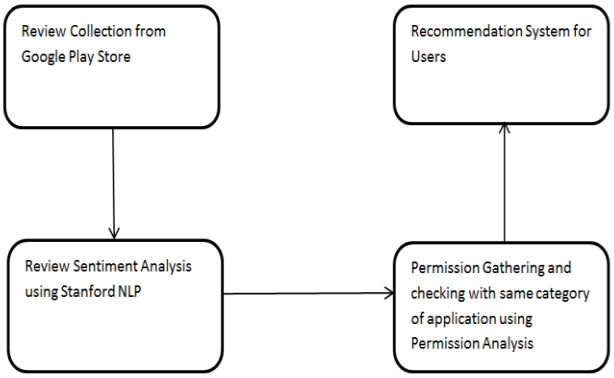

Fig -2: ReviewMiner system architecture

ReviewMiner uses linguistic and behavioural information to detect genuine reviews first. It then extracts user-identified fraud and malware indicators. Tools to collect and process Google Play data (e.g. GPCrawler) collects data published by Google Play for apps, users and reviews automatically.

In contrast to other tools for android malware detection, ReviewMiner employs a relational, linguistic and behavioural approach based on longitudinal app data as against the method of analysing app executables. In its use of app permissions ReviewMiner includes the temporal dimension, e.g., changes in the number of requested permissions, also the dangerous ones. ReviewMiner thus identifies and exploits a new relationship between malware and search rank fraud.

3. SYSTEM DESIGN

The various modules in the project are:

• Review Collection from Google Play store:

The first step involves getting the reviews for a particular app from Google Play store as in fig 3. This is done by typing the unique ID obtained from the URL of the app into the Search bar of this tool. This enables the tool to extract all reviews for the particular app available in Google Play store.

• Review Sentiment analysis using Stanford NLP:

Stanford NLP provides[10] a set of human language technology tools. It can provide base form of words, parts of speech, indicate sentiment and so on. It provides support for a number of major (human) languages and has a broad range of grammatical analysis tools. It can run as a fairly simple web service and is designed to be highly flexible and extensible. Stanford NLP makes it very easy to apply linguistic analysis to a piece of text.

It provides a high class library of words to which words from review comments can be compared in order to determine the sentiment behind the comment- good or bad. Sentiments behind each comment are made a note of. A column in the result display of the tool is labelled as review point. This indicates the sentiment analysis behind each comment. The review point can be any one of three possible values: 1 refers to a good review, -1 refers to a bad review and 0 is a neutral review.

• Permission gathering and checking with same category of application using permission analysis.

Malware[11], fraudulent and legitimate apps request large number of permissions. Android's API level labels 47 permissions as 'dangerous'. Some of the most popularly dangerous permissions are 'modify or delete the contents of USB storage', 'read phone status and identify' and 'access precise location'. Some malicious apps have a deceptive behaviour of attracting users with minimum permissions and later requesting dangerous permissions after installation. These apps generally referred to as 'Jekyll-Hyde apps', succeed when the user is unwilling to uninstall the app just to reject a few new permissions. This tool monitors the permissions requested by the app in an effort to determine malicious and fraudulent behaviour.

• Recommendation system for users:

A genuine review would mirror the author's experience and are usually informative. Keeping this in mind and analysing

the permissions requested by the app and the sentiments behind its reviews, the tool assigns a ranking for the app on a scale of 1 to 5. Some other factors are also taken into consideration. For example fraudsters are likely to post reviews within short intervals of time. They may use the same accounts to review more apps in common than regular users. The tool also uses general features such as the average rating of the app, its total number of ratings, reviews and installs. There is also the assumption that since no app is perfect, there would be a balanced review that contains both positive and negative sentiments and there would be a relation between the dominant sentiment on the app and its overall rating.

Also in certain cases the users themselves have indicated the fraudulent or malicious behaviour of the app in their reviews. The project was able to prepare a general list of words based on such reviews and check reviews for these words. For example malware indicator list had words such as risk, fraud, hack, corrupt, fake, malware, blacklists, ads. Words such as cheat, hideous, complain, wasted and crash made it to the fraud indicator list.

One more prominent factor was the app harassing the user to rate the app. These are generally termed as coercive review campaigns. Examples cite the users posting reviews such as "I could not proceed to second level without rating the app" or "Rate me pop-up keeps appearing while I'm playing, forcing me to post this". Words that make this list include make, ask, force. Some of these apps keep giving popups until the user has given it all 5 stars. Taking into consideration all such factors, a final ranking is displayed for the app. This final rating indicates the star rating for the app. In other words, for each review a star rating of 1-5 is given as in fig 3 and in the final display, the count of each star rating is provided. For user benefit the tool also displays the same in the form of a histogram, with the star rating along the X-axis and the number of reviews indicating that particular star rating along the Y-axis.

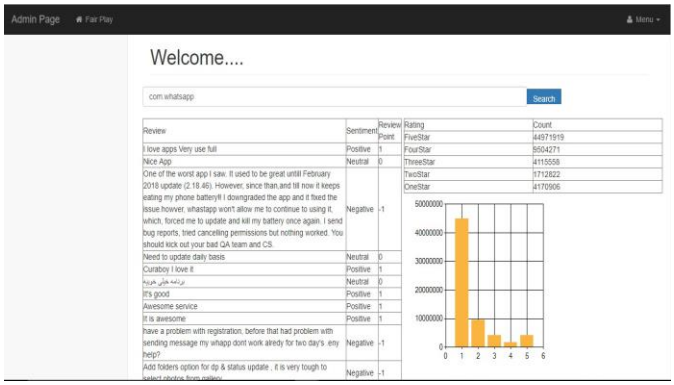

Fig -3: Review Extraction from Google play and calculating count of rating

4. CONCLUSIONS

In the current scenario, it is really essential to avoid people from being misguided by fraud and malware apps. Moreover they should be made aware of malware apps.

This system effectively helps in identifying malware apps based on reviews. It distinguishes between fake and genuine reviews and thereby proves that rating based on text processing is more reliable than usual star rating. It also checks whether the permission policy of a particular app violates its genuine requirements and also tries to identify those apps for which the users are forced to give positive reviews. Those reviews, called coercive reviews, are usually generated for fraudulent apps. Since it is based on language processing, it has high accuracy than star ratings.

REFERENCES

[1] Google Play. https://play.google.com/.

[2] Ezra Siegel. Fake Reviews in Google Play and Apple App Store. Appentive, 2014.

[3] Zach Miners. Report: Malware-infected Android apps spike in the Google Play store. PCWorld, 2014.

[4] Daniel Roberts. How to spot fake apps on the Google Play store. Fortune, 2015.

[5] Andy Greenberg. Malware Apps Spoof Android Market To Infect Phones. Forbes Security, 2014.

[6] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and Turf: Crowd- turfing for Fun and Profit. In Proceedings of ACM WWW. ACM, 2012.

[7] Jon Oberheide and Charlie Miller. Dissecting the Android Bouncer. SummerCon2012, New York, 2012.

[8] Yajin Zhou and Xuxian Jiang. Dissecting Android Malware: Characterization and Evolution. In Proceedings of the IEEE S&P, pages 95–109. IEEE, 2012.

[9] https://stanfordnlp.github.io/CoreNLP/api.html

[10] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas, and Gonzalo A´ lvarez. Puma: Permission usage to detect malware in android. In International Joint Conference CISIS12-ICEUTE´ 12-SOCO´ 12 Special Sessions, pages 289–298. Springer, 2013.