APP RANK FRAUD AND MALWARE DETECTION USING FUZZY LOGIC

A. Blessy[1] | T.B.Dharmaraj[2]

[1](UG Student, Christ the King Engineering College, blessyantony209@gmail.com)

[2](Professor, Christ the King Engineering College, bellidhramaraj@gmail.com)

Abstract— *FairPlay using Fuzzy Logic is introduced, a novel system that discovers and leverages traces left behind by fraudsters, to detect both malware and apps subjected to search rank fraud. FairPlay correlates review activities and uniquely combines detected review relations with linguistic and behavioural signals gleaned from Google Play app data (87K apps, 2.9M reviews, and 2.4M reviewers, collected over half a year), in order to identify suspicious apps. FairPlay achieves over 95% accuracy in classifying gold standard datasets of malware, fraudulent and legitimate apps. We show that 75% of the identified malware apps engage in search rank fraud. FairPlay discovers hundreds of fraudulent apps that currently evade Google Bouncer's detection technology. FairPlay also helped the discovery of more than 1,000 reviews, reported for 193 apps, that reveal a new type of "coercive" review campaign: users are harassed into writing positive reviews, and install and review other apps.*

Keywords— Fuzzy Logic, Android Market, Malware Detection, Rank Fraud

1. INTRODUCTION

The commercial success of Android app markets such as Google Play and the incentive model they offer to popular apps, make them appealing targets for fraudulent and malicious behaviours. Some fraudulent developers deceptively boost the search rank and popularity of their apps (e.g., through fake reviews and bogus installation counts) , while malicious developers use app markets as a launch pad for their malware . The motivation for such behaviours is impact: app popularity surges translate into financial benefits and expedited malware proliferation. Fraudulent developers frequently exploit crowdsourcing sites (e.g., Freelancer, Fiverr, Best App Promotion to hire teams of willing workers to commit fraud collectively, emulating realistic, spontaneous activities from unrelated people (i.e., "crowdturfing"), see Figure 1 for an example. We call this behaviour "search rank fraud". In addition, the efforts of Android markets to identify and remove malware are not always successful. For instance, Google Play uses the Bouncer system to remove malware. However, out of the 7, 756 Google

Play apps we analysed using Virus Total, 12% (948) were flagged by at least one anti-virus tool and 2% (150) were identified as malware by at least 10 tools. Previous mobile malware detection work has focused on dynamic analysis of app executable as well as static analysis of code and permissions. However, recent Android malware analysis revealed that malware evolves quickly to bypass anti-virus tools .In this paper, we seek to identify both malware and search rank fraud subjects in Google Play.

2. RELATED WORK

System models, We focus on the Android app market ecosystem of Google Play. The participants, consisting of users and developers, have Google accounts. Developers create and upload apps, that consist of executable (i.e., "apks"), a set of required permissions, and a description. The app market publishes this information, along with the app's received reviews, ratings, aggregate rating (over both reviews and ratings), install count range (predefined buckets, e.g., 50-100, 100-500), size, version number, price, time of last update, and a list of "similar" apps. Each review consists of a star rating ranging between 1-5 stars, and some text. The text is optional and consists of a title and a description. Google Play limits the number of reviews displayed for an app to 4, 000.

Adversarial model, We consider not only malicious developers, who upload malware, but also rational fraudulent developers. Fraudulent developers attempt to tamper with the search rank of their apps, e.g., by recruiting fraud experts in crowdsourcing sites to write reviews, post ratings, and create bogus installs. While Google keeps secret the criteria used to rank apps, the reviews, ratings and install counts are known to play a fundamental part (see e.g., [21]). To review or rate an app, a user needs to have a Google account, register a mobile device with that account, and install the app on the device. This process complicates the job of fraudsters, who are thus more likely to reuse accounts across jobs. The reason for search rank fraud attacks is impact. Apps that rank higher in search results, tend to receive more installs. This is beneficial both for fraudulent developers, who increase their revenue, and malicious developers who increase the impact of their malware.

2.1 Android Malware Detection

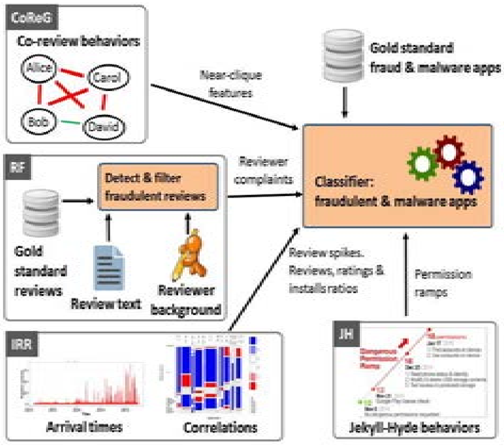
Zhou and Jiang collected and characterized 1, 200 Android malware samples, and reported the ability of malware to quickly evolve and bypass the detection mechanisms of anti-virus tools. Burguera et al. used crowdsourcing to collect system call traces from real users, then used a "partitioned" clustering algorithm to classify benign and malicious apps. Shabtai et al. extracted features from monitored apps. (e.g., CPU consumption, packets sent, running processes) and usedmachine learning to identify malicious apps. Grace et al. [15] used static analysis to efficiently identify high and medium risk apps.

Previous work has also used app permissions to pinpoint malware. Sharma et al use risk signals extracted from app permissions, e.g., rare critical permissions (RCP) and rare pairs of critical permissions (RPCP), to train SVM and inform users of the risks vs. benefits tradeoffs of apps. in § 5.3 we show that FairPlay significantly improves on the performance achieved by Sarma et al. Peng et al. propose a score to measure the risk of apps, based on probabilistic generative models such as Naive Bayes. Yerima et al also use features extracted from app permissions, API calls and commands extracted from the app executable.

Shahs and Khan used features extracted from app permissions and control flow graphs to train an SVM classifier on 2000 benign and less than 100malicious apps. Sanz et al. rely strictly on permissions as sources of features for several machine learning tools. They use a dataset of around 300 legitimate and 300 malware apps. Google has deployed Bouncer, a framework that monitors published apps to detect and remove malware. Overhead and Miller have analysed and revealed details of Bouncer (e.g., based in QEMU, using both static and dynamic analysis). Bouncer is not sufficient - our results show that 948 apps out of 7,756 apps that we downloaded from Google Play are detected as suspicious by at least 1 anti-virus tool.

2.2 Graph Based Opinion Spam Detection

Graph based approaches have been proposed to tackle opinion spam . Ye and Akoglu [24] quantify the chance of a product to be a spam campaign target, then cluster spammers on a 2-hop sub graph induced by the products with the highest chance values. Akoglu et al frame fraud detection as a signed network classification problem and classify users and products that form bipartite network using a propagation-based algorithm. Fair Play's relational approach differs as it identifies apps reviewed in a contiguous time interval, by groups of users with a history of reviewing apps in common. FairPlay combines the results of this approach with behavioural and linguistic clues, extracted from longitudinal app data, to detect both search rank fraud and malware apps. We emphasize that search rank fraud goes beyond opinion spam, as it implies fabricating not only reviews, but also user app install events and ratings.



3. APP PROVIDER MODULE

This is one of the leading topic which is playing important role in detecting fraudulent apps in applications store. The app provider module is where the authorisation and Updation of all the data needed for the app is regulated. The provider has the command over the data visibility to the users. The app user will be added or removed only on the admin's. To provide permission to user easily

access the app for utilization to improve secures. Admin and User login has to authenticated and registered. Then only the application allows the users to request a solution to his problem.

4.VIEW APPS AND PLACE RATINGS

This module is to be designed for providing the user-friendly interface. In the user Interface, for the first time, the user has to give the details such as name, Email ID and mobile number, password. In the settings of the app, the user has to specify the access permission. The admin have a rights to approval the details for the user. The admin have default username and password to login and give access control for the user. In addition ratings, most of the App stores also permit users to write some textual comments as App reviews. Such reviews can indicates the individual perceptions and usage experiences of existing users for particular mobile Apps.

5 .APP POSITIVE NEGATIVE ANALYSIS

To review or rate an app, a user needs to have a Google account, register a mobile device with that account, and install the app on the device. Indeed, review manipulation is one of the most valuable perspectives of App ranking fraud. Furthermore, we investigate three types of evidences, i.e., ranking based evidences, rating based evidences and review based evidences, by modeling Apps ranking, rating and review behavior through statistically prostheses tests. This process complicates the job of fraudsters, who are thus more likely to reuse accounts across jobs.

6.FIND OUT VARYING APP RATINGS

The reason for search rank fraud attacks is impact. Apps that rank higher in search results, tend to receive more installs. This is beneficial both for fraudulent developers, who increase their revenue, and malicious developers, who increase the impact of their

malware. Specifically, before downloading or purchasing a new mobile App, users usually first read its historical reviews to ease their decision making, and a mobile App contains more encouraging reviews may captivate more users to download.

7.APP CLASSIFICATION

In this module, Fuzzy logic algorithm is used to correctly classify apps as either benign, fraudulent or malware. Included in the rule base, fuzzy rules are in the form of "if-then" (for instance, if the temperature is above 86 degree, then adjust the air conditioner to cooling mode) and used to compute output fuzzy functions. The computing process is analyzed qualitatively and resistant to the change of malicious features that only affect the application feature.

8.CONCLUSIONS

We have introduced FairPlay, a system to detect both fraudulent and malware Google Play apps. Our experiments on a newly contributed longitudinal app data set, have shown that a high percentage of malware is involved in search rank fraud; both are accurately identified by FairPlay. In addition,

we showed Fair Play’s ability to discover hundreds of apps that evade Google Play’s detection technology, including a new type of coercive fraud attack.

REFERENCES

[1] Andy Greenberg. Malware Apps Spoof Android Market To Infect Phones.

[2] Best App Promotion. www.bestreviewapp.com/.

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

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

[5] Fiver https://www.fiverr.com/.

[6] Forbes Security, 2014.

[7] Freelancer. http://www.freelancer.com.

[8] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal,

[9] Haitao Zheng, and Ben Y. Zhao. Serf and Turf: Crowdturfing for Fun and Profit. In Proceedings of ACM WWW. ACM, 2012

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

[11] Stephanie Mlot. Top Android App a Scam, Pulled From Google Play. PCMag, 2014.

[12] Zachs Miners. Report: Malware-infected Android apps spike in the Google Play store. PC World, 2014.