Search Rank Fraud and Malware Detection In Google Play

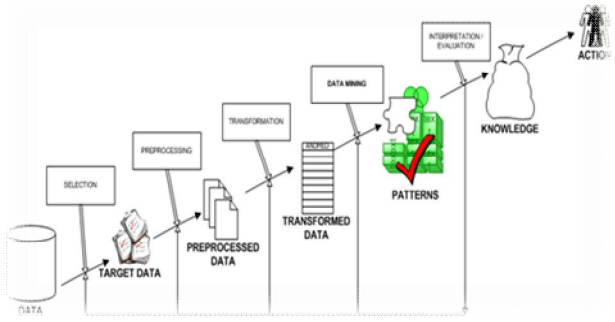
P. Sudha, R.Saranya, U.Suganya[1], Dr. R. Manivannan M.Tech[2], Mr. K.Preveen kumaR[3]
[1]Dept of CSE
[2, 3]Assistant Professor, Dept of CSE
[1, 2, 3] E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India.

Abstract- *Fraudulent behaviors in Google Play, the most popular Android app market, fuel search rank abuse and malware proliferation. To identify malware, previous work has focused on app executable and permission analysis. In this paper, we introduce FairPlay, 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 behavioral signals gleaned from Google Play app data (87 K apps, 2.9 M reviews, and 2.4M reviewers, collected over half a year), in order to identify suspicious apps. FairPlay achieves over 95 percent accuracy in classifying gold standard datasets of malware, fraudulent and legitimate apps. We show that 75 percent 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- Malware, fraudulent, Reviews, Rating.

I. INTRODUCTION

What is Data Mining?



Structure of Data Mining Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

How Data Mining Works?

While large-scale information technology has been evolving separate transaction and analytical systems, data mining provides the link between the two. Data mining software analyzes relationships and patterns in stored transaction data based on open-ended user queries. Several types of analytical software are available: statistical, machine learning, and neural networks.

Generally, any of four types of relationships are sought:

- **Classes**: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.
- **Clusters**: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- **Associations**: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- **Sequential patterns**: Data is mined to anticipate behavior patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

Data mining consists of five major elements:

1) Extract, transform, and load transaction data onto the data warehouse system.
2) Store and manage the data in a multidimensional database system.
3) Provide data access to business analysts and information technology professionals.
4) Analyze the data by application software.
5) Present the data in a useful format, such as a graph or table.

Different levels of analysis are available:

- **Artificial neural networks**: Non-linear predictive models that learn through training and resemble biological neural networks in structure.
- **Genetic algorithms**: Optimization techniques that use process such as genetic combination, mutation, and natural selection in a design based on the concepts of natural evolution.
- **Decision trees**: Tree-shaped structures that represent sets of decisions. These decisions generate rules for the classification of a dataset. Specific decision tree methods include Classification and Regression Trees (CART) and Chi Square Automatic Interaction Detection (CHAID). CART and CHAID are decision tree techniques used for classification of a dataset. They provide a set of rules that you can apply to a new (unclassified) dataset to predict which records will have a given outcome. CART segments a dataset by creating 2-way splits while CHAID segments using chi square tests to create multi-way splits. CART typically requires less data preparation than CHAID.
- **Nearest neighbor method**: A technique that classifies each record in a dataset based on a combination of the classes of the *k* record(s) most similar to it in a historical dataset (where *k*=1). Sometimes called the *k*-nearest neighbor technique.
- **Rule induction**: The extraction of useful if-then rules from data based on statistical significance.
- **Data visualization**: The visual interpretation of complex relationships in multidimensional data. Graphics tools are used to illustrate data relationships.

Characteristics of Data Mining:

- **Large quantities of data**: The volume of data so great it has to be analyzed by automated techniques e.g. satellite information, credit card transactions etc.
- **Noisy, incomplete data**: Imprecise data is the characteristic of all data collection.
- **Complex data structure**: conventional statistical analysis not possible
- **Heterogeneous data stored in legacy systems**

Benefits of Data Mining:

1) It's one of the most effective services that are available today. With the help of data mining, one can discover precious information about the customers and their behavior for a specific set of products and evaluate and analyze, store, mine and load data related to them
2) An analytical CRM model and strategic business related decisions can be made with the help of data mining as it helps in providing a complete synopsis of customers
3) An endless number of organizations have installed data mining projects and it has helped them see their own companies make an unprecedented improvement in their marketing strategies (Campaigns)
4) Data mining is generally used by organizations with a solid customer focus. For its flexible nature as far as applicability is concerned is being used vehemently in applications to foresee crucial data including industry analysis and consumer buying behaviors
5) Fast paced and prompt access to data along with economic processing techniques have made data mining one of the most suitable services that a company seek

Advantages of Data Mining:

1) Marketing / Retail:

Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign…etc. Through the results, marketers will have appropriate approach to sell profitable products to targeted customers.

Data mining brings a lot of benefits to retail companies in the same way as marketing. Through market basket analysis, a store can have an appropriate production arrangement in a way that customers can buy frequent buying products together with pleasant. In addition, it also helps the retail companies offer certain discounts for particular products that will attract more customers.

2) Finance / Banking

Data mining gives financial institutions information about loan information and credit reporting. By building a model from historical customer's data, the bank and financial institution can determine good and bad loans. In addition, data

mining helps banks detect fraudulent credit card transactions to protect credit card's owner.

2) Manufacturing

By applying data mining in operational engineering data, manufacturers can detect faulty equipments and determine optimal control parameters. For example semi-conductor manufacturers has a challenge that even the conditions of manufacturing environments at different wafer production plants are similar, the quality of wafer are lot the same and some for unknown reasons even has defects. Data mining has been applying to determine the ranges of control parameters that lead to the production of golden wafer. Then those optimal control parameters are used to manufacture wafers with desired quality.

4) Governments

Data mining helps government agency by digging and analyzing records of financial transaction to build patterns that can detect money laundering or criminal activities.

5) Law enforcement:

Data mining can aid law enforcers in identifying criminal suspects as well as apprehending these criminals by examining trends in location, crime type, habit, and other patterns of behaviors.

6) Researchers:

Data mining can assist researchers by speeding up their data analyzing process; thus, allowing those more time to work on other projects.

II. LITERATURE SURVEY

1) Crow droid: Behavior-Based Malware Detection System for Android

AUTHORS: IkerBurguera, UrkoZurutuza

The sharp increase in the number of smart phones on the market, with the Android platform posed to becoming a market leader makes the need for malware analysis on this platform an urgent issue. In this paper we capitalize on earlier approaches for dynamic analysis of application behavior as a means for detecting malware in the Android platform. The detector is embedded in a overall framework for collection of traces from an unlimited number of real users based on crowd sourcing. Our framework has been demonstrated by analyzing the data collected in the central server using two types of data sets: those from artificial malware created for test purposes, and those from real malware found in the wild. The method is shown to be an effective means of isolating the malware and alerting the users of a downloaded malware. This shows the potential for avoiding the spreading of a detected malware to a larger community.

2) Andromaly: a Behavioral Malware Detection Framework for Android Devices

AUTHORS: AsafShabtai, Uri Kanonov

This article presents Andromaly—a framework for detecting malware on Android mobile devices. The proposed framework realizes a Host-based Malware Detection System that continuously monitors various features and events obtained from the mobile device and then applies Machine Learning anomaly detectors to classify the collected data as normal (benign) or abnormal (malicious). Since no malicious applications are yet available for Android, we developed four malicious applications, and evaluated Andromaly's ability to detect new malware based on samples of known malware. We evaluated several combinations of anomaly detection algorithms, feature selection method and the number of top features in order to find the combination that yields the best performance in detecting new malware on Android. Empirical results suggest that the proposed framework is effective in detecting malware on mobile devices in general and on Android in particular.

3) Riskranker: Scalable and Accurate Zero-day Android Malware Detection

AUTHORS: Michael Grace, Yajin Zhou

Smartphone sales have recently experienced explosive growth. Their popularity also encourages malware authors to penetrate various mobile marketplaces with malicious applications (or apps). These malicious apps hide in the sheer number of other normal apps, which makes their detection challenging. Existing mobile anti-virus software are inadequate in their reactive nature by relying on known malware samples for signature extraction. In this paper, we propose a proactive scheme to spot zero-day Android malware. Without relying on malware samples and their signatures, our scheme is motivated to assess potential security risks posed by these untrusted apps. Specifically, we have developed an automated system called *Risk Ranker* to scalably analyze whether a particular app exhibits dangerous behavior (e.g., launching a root exploit or sending background SMS messages). The output is then used to produce a prioritized list

of reduced apps that merit further investigation. When applied to examine 118,318 total apps collected from various Android markets over September and October 2011, our system takes less than four days to process all of them and effectively reports 3281 risky apps. Among these reported apps, we successfully uncovered 718 malware samples (in 29 families) and 322 of them are zero-day (in 11 families). These results demonstrate the efficacy and scalability of Risk Ranker to police Android markets of all stripes.

4) Android Permissions: a Perspective Combining Risks and Benefits.

AUTHORS: Bhaskar Pratim Sarma, Ninghui Li

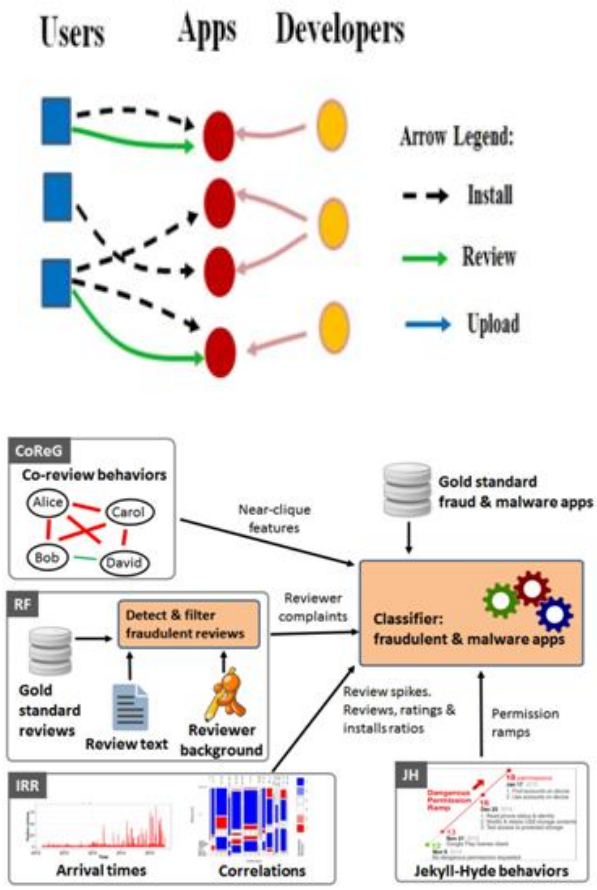
The phenomenal growth of the Android platform in the past few years has made it a lucrative target of malicious application (app) developers. There are numerous instances of malware apps that send premium rate SMS messages, track users' private data, or apps that, even if not characterized as malware, conduct questionable actions affecting the user's privacy or costing them money. In this paper, we investigate the feasibility of using both the permissions an app requests, the category of the app, and what permissions are requested by other apps in the same category to better inform users whether the risks of installing an app is commensurate with its expected benefit. Existing approaches consider only the risks of the permissions requested by an app and ignore both the benefits and what permissions are requested by other apps, thus having a limited effect. We propose several risk signals that and evaluate them using two datasets, one consists of 158,062 Android apps from the Android Market, and another consists of 121 malicious apps. We demonstrate the effectiveness of our proposal through extensive data analysis.

5) Using Probabilistic Generative Models for Ranking Risks of Android Apps

AUTHORS: HaoPeng, Chris Gates

One of Android's main defense mechanisms against malicious apps is a risk communication mechanism which, before a user installs an app, warns the user about the permissions the app requires, trusting that the user will make the right decision. This approach has been shown to be ineffective as it presents the risk information of each app in a "tand-alone" ashion and in a way that requires too much technical knowledge and time to distill useful information. We introduce the notion of risk scoring and risk ranking for Android apps, to improve risk communication for Android apps, and identify three desiderata for an effective risk scoring scheme. We propose to use probabilistic generative models for risk scoring schemes, and identify several such models, ranging from the simple Naive Bayes, to advanced hierarchical mixture models. Experimental results conducted using real-world datasets show that probabilistic general models significantly outperform existing approaches, and that Naive Bayes models give a promising risk scoring approach.

III. SYSTEM ARCHITECTURE



Architecture explore the contextual functional point and flow for every role holder process. It Contains role holder, entity, external resource and process flow. Proposed system exists with various role holder like app developer, end user, and admin. Every functionality have individual data set or entity to hold or track and keep the record of functional flow.

User- User want to analyst the app review or post rating/ review before use a app from app portal

Developer- Developer post their app with minimum required information in the portal for common availability.

Admin- Admin has to approve user account for every role holder due to enroll or login for user activity.

App- App treated as entity for all role holder, it carries their review or rating information while applied or user review.

IV. IMPLEMENTATION

MODULES:

- System model
- Adversarial model
- The Co-Review Graph (CoReG) Module
- Reviewer Feedback (RF) Module

MODULES DESCSRIPTION:

System model:

In the first module, of the project we develop the System environment model to evaluate the performance of the our system for Search Rank Fraud. We focus on the Android app market eco system of Google Play. The participants, consisting of users and developers, hae Google accounts. Developers create and upload apps, that consist of executables (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, 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. In this module, we illustrate the participants in Google Play and their relations.

Adversarial model:

In the second module, we develop the Adversarial model for considering the malicious users. 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 crowd sourcing 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.

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.

The Co-Review Graph (CoReG) Module

This module exploits the observation that fraudsters who control many accounts will re-use them across multiple jobs. Its goal is then to detect sub-sets of an app's reviewers that have performed significant common review activities in the past. In the following, we describe the co-review graph concept, formally present the weighted maximal clique enumeration problem, then introduce an efficient heuristic that leverages natural limitations in the behaviors of fraudsters .Let the co-review graph of an app ,be a graph where nodes correspond to user accounts who reviewed the app, and undirected edges have a weight that indicates the number of apps reviewed in common by the edge's endpoint users. The co-review graph concept naturally identifies user accounts with significant past review activities.

Reviewer Feedback (RF) Module

Reviews written by genuine users of malware and fraudulent apps may describe negative experiences. The RF module exploits this observation through a two step approach:(i) detect and filter out fraudulent reviews, then (ii) identify malware and fraud indicative feedback from the remaining reviews.

V. CONCLUSION

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 FairPlay's ability to discover hundreds of apps that evade Google Play's detection technology, including a new type of coercive fraud attack.

Application rating and review declares app quality based on user manipulation and their feedback for every posted app. Review and rating dynamically parsed and generated out while a user review a app from portal.

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. PC World, 2014.
[4] Stephanie M lot. Top Android App a Scam, Pulled From Google Play. PCMag, 2014.
[5] Daniel Roberts. How to spot fake apps on the Google Play store. Fortune, 2015.
[6] Andy Greenberg. Malware Apps Spoof Android Market To Infect Phones. Forbes Security, 2014.
[7] Freelancer. http://www.freelancer.com.
[8] Fiverr. https://www.fiverr.com/.
[9] BestAppPromotion. www.bestreviewapp.com/.
[10] Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, ManishMohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and Turf: Crowd turfing for Fun and Profit. In *Proceedings of ACM WWW*. ACM, 2012.
[11] Jon Oberheide and Charlie Miller. Dissecting the Android Bouncer. *SummerCon2012, New York*, 2012.
[12] Virus Total - Free Online Virus, Malware and URL Scanner. https: //www.virustotal.com/, Last accessed on May 2015.
[13] Iker Burguera, Urko Zurutuza, and Simin Nadjm-Tehrani. Crowdroid:Behavior-Based Malware Detection System for Android. In *Proceedings of ACM SPSM*, pages 15–26. ACM, 2011.
[14] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and YaelWeiss. Andromaly: a Behavioral Malware Detection Frame work for Android Devices. *Intelligent Information Systems*, 38(1):161–190,2012.
[15] Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou, and XuxianJiang. Riskranker: Scalable and Accurate Zero-day Android Malware Detection. In *Proceedings of ACM MobiSys*, 2012.
[16] Bhaskar Pratim Sarma, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. Android Permissions: aPerspective Combining Risks and Benefits. In *Proceedings of ACMSACMAT*, 2012.
[17] Hao Peng, Chris Gates, Bhaskar Sarma, Ninghui Li, Yuan Qi,Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy. Using Probabilistic Generative Models for Ranking Risks of Android Apps. In *Proceedings of ACM CCS*, 2012.
[18] S.Y. Yerima, S. Sezer, and I. Muttik. Android Malware Detection Using Parallel Machine Learning Classifiers. In *Proceedings of NGMAST*, Sept 2014.
[19] Yajin Zhou and Xuxian Jiang. Dissecting Android Malware:Characterization and Evolution. In *Proceedings of the IEEE S&P*,pages 95–109. IEEE, 2012.
[20] Fraud Detection in Social Networks. https://users.cs.fiu.edu/_carbunar/caspr.lab/socialfraud.html.
[21] Google I/O 2013 - Getting Discovered on Google Play. www.youtube.com/watch?v=5Od2SuL2igA, 2013.
[22] Justin Sahs and Latifur Khan. A Machine Learning Approach to Android Malware Detection. In *Proceedings of EISIC*, 2012.
[23] Borja Sanz, Igor Santos, Carlos Laorden, Xabier Ugarte-Pedrero, Pablo Garcia Bringas, and Gonzalo ´Alvarez. Puma: Permission usage to detect malware in android. In *International Joint ConferenceCISIS12-ICEUTE´ 12-SOCO´ 12 Special Sessions*, pages 289–298.Springer, 2013.
[24] Junting Ye and Leman Akoglu. Discovering opinion spammer groups by network footprints. In *Machine Learning and Knowledge Discovery in Databases*, pages 267–282. Springer, 2015.
[25] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion Fraud Detection in Online Reviews by Network Effects. In *Proceedings of ICWSM*, 2013.
[26] Android Market API. https://code.google.com/p/android-market-api/, 2011.
[27] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.*, 363(1):28–42, October2006.
[28] Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. 3111:260–272, 2004.
[29] Takeaki Uno. An efficient algorithm for enumerating pseudo cliques. In *Proceedings of ISAAC*, 2007.
[30] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly, 2009.
[31] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs Up? Sentiment Classification Using Machine Learning Techniques .In *Proceedings of EMNLP*, 2002.
[32] John H. McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, second edition, 2009.
[33] New Google Play Store greatly simplifies permissions.http://www.androidcentral.com/new-google-play-store-4820-greatly-simplifies-permissions,2014.
[34] Weka. http://www.cs.waikato.ac.nz/ml/weka/.
[35] S. I. Gallant. Perceptron-based learning algorithms. *Trans. Neur.Netw.*, 1(2):179–191, June 1990.
[36] Leo Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.
[37] Ron Kohavi. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of IJCAI*,1995.
[38] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. In *Proceedings of the SIAM SDM*, 2011.
[39] Acar Tamersoy, Kevin Roundy, and Duen Horng Chau. Guilt by association: Large scale malware detection by mining file-relation graphs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge*

Discovery and Data Mining, KDD '14, pages1524–1533, New York, NY, USA, 2014. ACM