Search Rank Fraud and Malware Detection in Mobile Apps

Dr. P. Venkateswara Rao[1],

Professor, Department of Computer Science Engineering , Narayana Engineering College, Gudur, AP, India 524101,

Sk.Farheen[2],P.Krishna Lasya[3],P.Thanusha[4],Sk.Ayesha[5]
skfarheen2000@gmail.com, pamidimarrikrishnalasya1820@gmail.com , thanushapamuru@gmail.com,

skayesha9652@gmail.com

Student, Department of Computer Science Engineering, Narayana Engineering College, Gudur, AP, India 524101,

Abstract: The financial success of Android software stores like Google Play, as well as the incentive scheme they provide for popular apps, make them attractive targets for fraud and malicious behavior. Some fraudulent developers deceptively boost the search rank and popularity of their apps while malicious developers use app markets as a launch pad for their malware. In this project, 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 in order to identify suspicious apps. Adversaries can have chances to launch attacks by gathering victim's information continuously. This project shows that an adversary can successfully infer a victim's vertex identity and community identity by the knowledge of degrees within a time period. The project also includes a new supervised clustering algorithm to find groups of data (coarse and finer cluster). It directly incorporates the information of sample categories into the fraud clustering process.

Keywords: -Android market, search rank fraud, malware detection.

I. INTRODUCTION

A social networking service (SNS) is a platform to build social networks orPeople with comparable interests, hobbies, histories, or real-life relationships form social bonds.A social network service comprises of a user's profile, social connections, and a variety of additional services[1][2].Social network sites are web-based services that allow individuals to create a public profile, create a list of users with whom to share connections, and view and cross the connections within the system[3].The most all social network services are web-based and provide users ways to communicate over the Web, such as e-mail and instant messaging. Social network sites are varied and they incorporate new information and communication tools such as mobile connectivity, photo, video, sharing[4][5]. Online community services are sometimes called social network sites, however that isn't always the case. In a larger sense, a social network service is one that is focused on an individual, whereas internet forum services are those that are focused on a group of people[6].Users can exchange ideas, photographs, postings, activities, events, and hobbies with individuals in their network through social networking sites.

II. LITERATURESURVEY CROWDROID: BEHAVIOR-BASED MALWARE DETECTION SYSTRM FOR ANDROID

With the Android platform, there has been a significant growth in the number of smart phones on the market. On track becoming the market leader, malware detection on this platform has never been more important.We use previous methodologies for dynamic analysis of application activity as a way of identifying malware on the Android platform in this study.The detector is part of a larger framework that uses crowd sourcing to gather traces from an infinite number of genuine users. The strategy has been proved to be an effective way of isolating malware and alerting people to malware that has been downloaded.This demonstrates the possibility of preventing the transmission of discovered malware to a bigger population.In the next five years, all market indications point to a tremendous increase in the number of smart phones acquired. This will create a potential for a massive increase in malware generation, and in particular in the sector dominated by the market leader, potentially the Android platform.We have presented a novel methodology for obtaining and analyzing smart phone application activity in this research. In collaboration with the Android user's community, it will be capable of distinguishing between DetectingKnown programs that behave abnormally, as well as malicious apps with the same name and version..In addition, we tested our platform on a variety of test smart phones.as a technique of studying upcoming technologies, have built a proof of concept for We've shown that monitoring system calls is a viable method for detecting malware[7]. This kind of analysis is commonly utilized in the literature. According to the results of the quick poll, there are a variety of techniques to detecting malwareThere are a range of approaches for detecting malware, according to the results of the short poll.One of the more accurate strategies we investigated was monitoring system callsto determine the behavior of Android applications, since they provide detailed low-level information. We recognize that approaches such as API call analysis, information flow tracking, and network monitoring can help.to a more thorough examination of the virus, yielding more helpful information about the infection's activities and more precise results results. On the other hand, more monitoring capability will place a higher demand on the number of resources consumed in the device. The approach we offer for acquiring genuine traces of application activity is the most significant addition of this study[8][9].We've observed in earlier research that artificially manufactured user activities can be used to gain behavior information, or that artificially made user actions can be used to obtain behavior information[10][11][12].We've observed in earlier research that artificially manufactured user activities can be used to gain behavior information, or that artificially made user actions can be used to obtain behavior information[13].

USINGPROBABILISTIC GENERATIVE MODELS FOR RANKING RISKS OF ANDROID APPS

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 inin a "stand-alone" manner that necessitates excessive technical expertise and effort to derive meaningful information.For Android apps, we introduce the concepts of risk score and risk rating 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. The results of experiments utilizing real-world datasets suggest that probabilistic generic models are more effective.outperform previous methods, and that Naive Bayes models provide a good risk grading systemmethod. Android is an open-source software stack for mobile devices that includes an operating system, an application framework, and core applications. The kernel for the operating system is adapted from Linux.

GUILT BY ASSOCIATION: LARGE SCALE MALWARE DETECTION BY MINING FILERELATION GRAPHS

Malicious software is becoming more sophisticated, necessitating new defensive approaches that are more difficult to circumvent and capable of safeguarding users from new dangers.We introduce Aesop, a scalable method that uses Aesop's principle "a man is known by the company he maintains" to find executable files that are hazardous Aesop is a scalable algorithm based on Aesop's idea that "a man is known by the company hemaintains" to detect tight ties between files that often occur together on their workstations and between files to discover malicious executable filesthat reside on their computersAesop employs locality-sensitive hashing to assess the strength of these inter-file linkages in order to create a graph. wide-scale inference by propagating data from labelled files (as benign or malicious) to a large number of unlabeled files.On dataset D, LSH produces numerous bands, each with a different number of bucketsthat contain labeled and unlabeled filesA file only appears once in a band, within one of the band's bins.It's worth noting that the file may come with a varied collection of files depending on the band.In Table 3, for example, file f 5 appears in two bands with file f 4 and in one band with file f 6.In Table 3, for example, file f 5 appears in two bands with file f 4 and in one band with file f 6.

EXISTING SYSTEM

Using FairPlay, the current method attempts to discover malware and search rank fraud subjects in Google Play.This combination isn't coincidental; it's possible that bad developers use search rank fraud to increase the effect of their infection.The proposed system is built on the observation that fraudulent and malicious behaviors leave behind telltale signs on app marketsFairPlay is a Fraud and Malware Detection Approach that uses co-review graphs to simulate user-to-user reviewing relationships.To identify dubious review spikes received by applications, the temporal aspects of review post timings are employed.The linguistic and behavioral information is used to detect genuine reviews from which and then extract user-identified fraud and malware indicators. In the existing system, The Co-Review Graph (CoReG) module identifies apps reviewed groupings of users with considerably overlapping reviews in a contiguous time rangehistoryThe Review Feedback (RF) feature takes advantage of actual reviewer feedback, whilst the Inter Review Relation (IRR) feature takes advantage of relationships between reviews, ratings.

II. PROBLEM DEFINITION

- The preservation of vertex
- Community identities of individuals in a dynamic network is not taken into account.
- It is not carried out a privacy model for securing multi-community identification.
- Subjects alone do not constitute an invasion of privacy.

V.PROPOSED SYSTEM

The details of the planned system are shown below.

- It immediately combines sample category information into the fraud clustering method.
- To quantify the similarity between users, a new quantitative measure is presented that integrates the information of sample categories.
- The suggested technique is based on a novel quantitative metric for determining user similarity.
- Less dense nodes in the graph are removed so users who rating in minimum amount are not treated as fraudsters.

Advantages

The proposed system has following advantages.

- The number of clusters is calculated, and the fraud's relevancy is filtered such that coarse and fine clusters are created.
- Cliques' preparation correctly identifies fraud users.
- In graphs, fraud users with a lot of connections are also tracked.

ALGORITHM

A PCF (Pseudo Clique Finder) is a suggested algorithm that takes use of the fact that fraudsters hired toreview an app are more likely to submit such evaluations in a short period of time (e. g., days).PCF (see Algorithm 1) accepts a set of app reviews arranged by days as input, as well as a threshold value u.PCF output a set of identified pseudo-cliques with r u, that were formed during contiguous time frames.

Algorithm 1: PCF Algorithm Pseudo-Code

Input: The weightedθdays, an array of daily reviews and threshold density.

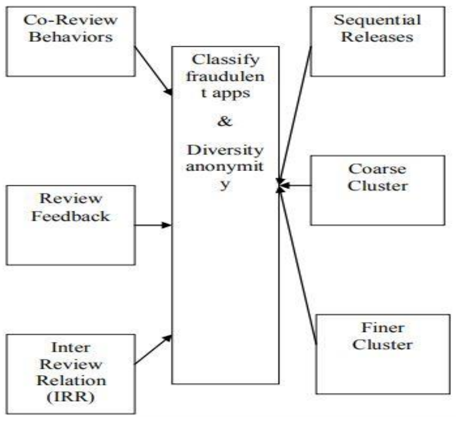
Output: all Cliques, set of all detected pseudo-cliques

```
1. for d :=0 d
2. Graph PC := new Graph();
3. bestNearClique(PC, days[d]);
4.  c := 1; n := PC.size();
5.  for nd := d+1; d n); endfor
6. bestNearClique(PC, days[nd]);
7. c := (PC.size() > n); endfor
8. if (PC.size() > 2)
9. allCliques := allCliques.add(PC); fi endfor
10. return
11. functionbestNearClique(Graph PC, Set
    revs)
12. if (PC.size() = 0)
13. for root := 0; root
14. Graph candClique := new Graph ();
15. candClique.addNode (revs[root].getUser());
16. do candNode := getMaxDensityGain(revs);
17. if (density(candClique  ))
    ∪{candNode}≥θ)).
18. candClique.addNode(candNode); fi
19. while (candNode != null);
20. if (candClique.density() >maxRho)
21. maxRho := candClique.density();
22. PC := candClique; fi endfor
23. else if (PC.size() > 0)
24. o candNode := getMaxDensityGain(revs);
    ))θ
25.  if (density(candClique[candNode) ≥
26. PC.addNode(candNode); fi
27.  while (candNode != null);
28.  return.
```

PCF selects the most promising pseudo-clique for each day when the app has received a review (lines 3 and 12, 1). Begin with each review, then add as many as you can to a potential pseudo-clique; maintain the fictitious clique (of the day) with the highest density. Move on to the following day (line 5) with the "working-in-progress" pseudo-clique: add greedily.

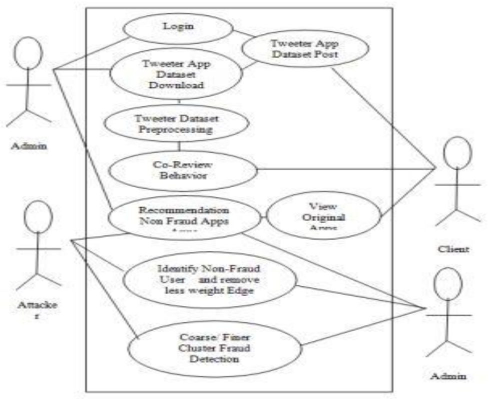
It is noted that if multiple fraudsters target an app in the same day, PCF may detect only the most densely linked pseudo-clique, which corresponds to the most prolific fraudster, while overlooking the less dense ones CoReG Advantages.From the output of PCF, CoReG extracts the following characteristics (see Table 1) I the total number of cliques (ii) the maximum, median andθdensity equals or exceeds standard deviation of the densities of identified pseudo cliques, (iii)The maximum, median, and standard deviation of the node count of discovered pseudo-cliques, normalized by n (the number of reviews for the app), and (iv) the total number of co-review network nodes that are members of at least one pseudo clique, normalized byn.

VII. SYSTEM FLOW DIAGRAM

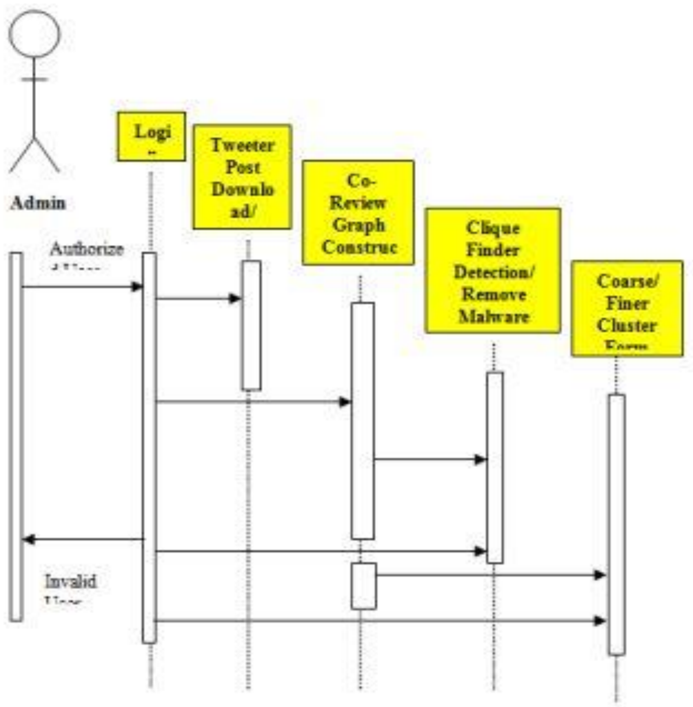


VIII.UML DIAGRAM

(i) Use case Diagram


Login
Tweeter App Dataset Post
Tweeter App Dataset Download
Tweeter Dataset Preprocessing
Co-Review Behavior
Recommendation Non Fraud Apps
View Original Apps
Identify Non-Fraud User and remove less weight Edge
Coarse/ Finer Cluster Fraud Detection
Admin
Attacker
Client
Admin

(ii) Sequence Diagram



IX. CONCLUSION

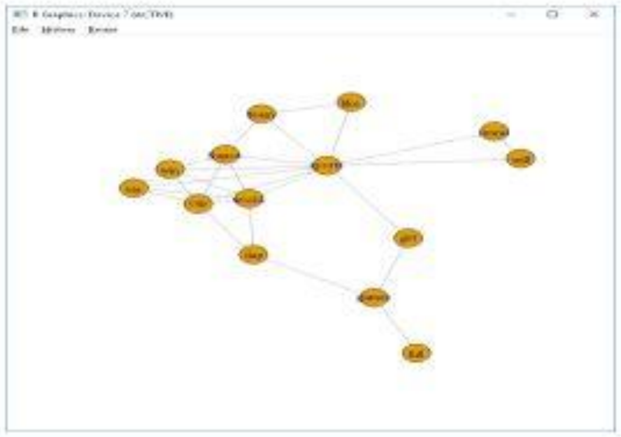
Some dishonest developers artificially enhance their apps' search rank and popularitye. g.through fake reviews and bogus installation counts), while malicious developers use app markets as a launch pad for their malware. The incentive for such actions is impact: increased app popularity translates into cash rewards and faster virus propagation.

This project aims to discover malware as well as search rank fraud in Google Play.This is not an accident: we believe that bad developers use search rank deception to increase the impact of their infection.Unlike other alternatives, our initiative is based on the fact that fraudulent and harmful activities leave telltale traces on app stores.

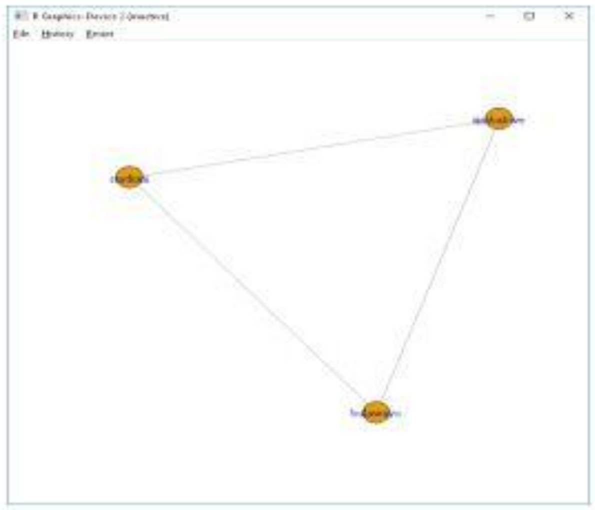
As part of the initiative, FairPlay, a tool for detecting both fraudulent and dangerous Google Play apps, has been installed.Experiments using Twitter tweets indicated a huge number of fake accounts.It also demonstrated FairPlay's capacity to identify non-fraud users.

X. OUTPUT

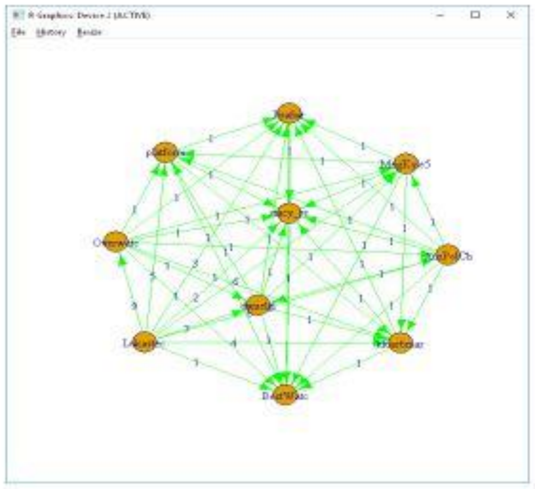
(i) Sample Screen



(ii) CO-REVIEW BACK GROUNDER



(iii) CLIQUE DETECTION



XI. REFERENCES

[1] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani , “Crowdroid: Behavior-based Malware detection system for Android,” in Proc. ACM SPSM, 2011, pp. 15–26.

[2] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, “Andromaly: A behavioral malware detection framework for

Androiddevices,"Intell.Inform. Syst.,vol.38, no.1, pp.161–190,2012.

[3] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, "RiskRanker: Scalable and accurate zero-day Android malware detection," in Proc. ACM MobiSys, 2012, pp. 281–294.

[4] H. Peng, et al., "Using probabilistic generative models for ranking risks of Android Apps," in Proc. ACM Conf. Comput. Commun. Secur., 2012, pp. 241–252.

[5] S. Yerima, S. Sezer, and I. Muttik, "Android Malware detection using parallel machine learning classifiers," in Proc. NGMAST, Sep. 2014, pp. 37–42.

[6] J. Sahs and L. Khan, "A machine learning approach to Android malware detection," in Proc. Eur. Intell. Secur. Inf. Conf., 2012, pp. 141–147.

[7] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas, and G. Alvarez, "Puma: Permission usage to detect malware in android," in Proc. Int. Joint Conf. CISIS12-ICEUTE' 12-SOCO' Special Sessions, 2013, pp. 289–298.

[8]. S. Jyothi, V. Sucharita, D.M. Mamatha " Survey on Computer Vision and Image Analysis based Techniques in Aquaculture", CIIT International Journal of Digital Image Processing, 2013
Castleman, K. (1993). Digital image processing

[9]. V. Sucharita, S. Jyothi ,D.M. MamathaA Comparative Study on Various Edge Detection Techniques used for the Identification of Penaeid Prawn Species ,International Journal of Computer Applications (0975 – 8887) Volume 78 – No.6, September 2013

[10]. Sucharita, V., Venkateswara Rao, P., Bhattacharyya, D., Kim, T.-H. Classification of penaeid prawn species using radial basis probabilistic neural networks and support vector machines International Journal of Bio-Science and Bio-Technology, 2016, 8(1), pp. 255–262

[11]Venkateswara Rao, P., Ramamohan Reddy, A., Sucharita, V.An approach of detecting white spot syndrome of peaneid SHRIMP using improved FCM with hybrid back propagation neural network, International Journal of Pharmacy and Technology, 2016, 8(4), pp. 22351–22363

[12]Mandava Geetha Bhargava, Modugula TS Srinivasa Reddy, Shaik Shahbaz, P Venkateswara Rao, V Sucharita Potential of big data analytics in bio-medical and health care arena: An exploratory study, Global Journal of Computer Science and Technology 2017/8/5

[13]P. Venkateswara Rao , A. Ramamohan Reddy , V. Sucharita, Computer Aided Shrimp Disease Diagnosis in Aquaculture. International Journal for Research in Applied

[14] M.Shanthini, P.Rajasekar and H.Mangalam (2014),"Design of low power S-Box in Architecture Level using GF", International Journal of Engineering Research and General Science Volume 2, Issue 3, April-May 2014