International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 3

Search Rank Fraud and Malware Detection in Google Play

Prajakta Rane, Priya Mishra, Dr. Archana Chaugule

Department of Computer Engineering, Pimpri Chinchwad College of Engineering and Research (PCCOER), Ravet, Pune, Maharashtra, India

ABSTRACT

The Google Play- the foremost widespread automation app market where rank abuse and malware search has increased rapidly. In this paper, we have a tendency to introduce Fairplay, a unique system that discovers and traces malware left behind by fraudsters. The proposed system's aim is to discover malware and apps subjected to search rank fraud. Fairplay correlates review activities and unambiguously combines detected review relations with linguistic and behavioural signals obtained from Google Play app information. Fairplay achieves gold customary datasets of malware and we go for broad read by applying some technique to each application to gauge its ranking. Our necessity is to create a perfect, fraud less application. Fraudsters create fraud by downloading application through numerous devices and provide fraud ratings and reviews. So, we tend to aforesaid to mine crucial information relating to specific application through reviews that are acquired from comments. Later, these reviews are combined to mine fraud in application ranking.

Keywords: *Android Applications, Fairplay, Fraud rating*

I. INTRODUCTION:

The industrial successes of android app markets like Google Play have increased the appealing targets for dishonest and malicious behaviour. We have a tendency to use activity knowledge to note real reviews and from that we extract user-identified fraud and malware indicators. Review consists of a star rating between 1-5 stars and app developers that extend rating of application by installing the application multiple times. We are introducing a system that discovers and leverages traces left behind by fraudsters to sight every malware and apps subjected to seem rank fraud. We can detect malicious developers as well as dishonest developers. Dishonest developers attempt to tamper with the search rank of their apps. The police investigation, fraud rating and reviews regarding application and trace the malware prior of installation and downloading application on single registration ID. Fair play is employed for organizing the analysis information of the application.

II. MOTIVATION

Fraudulent developers often exploit crowdsourcing sites (e.g., Freelancer, Fiverr, BestAppPromotion) to rent teams of willing workers to commit fraud place, emulating realistic, spontaneous activities. This is called behaviour search rank fraud. In addition, the efforts of automation markets to identify and exclude malware does not appear to be constantly roaring. As an example, Google Play uses the guard system to urge obviate malware. Previous mobile malware detection work has targeted on dynamic analysis of app executables also as static analysis of code and permissions. However, in recent malware automation analysis discovered that it evolves quickly to bypass anti-virus tools.

III. LITERATURE SURVEY

Paper Name: Android Permissions: A Perspective Combining
Authors: Bhaskar Pratim Sarma, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina
Nita-Rotaru, and Ian Molloy.

Year: 2012.
Description: In this paper we exploit earlier approaches for dynamic analysis of application behaviour as a method for detection malware. The detector is embedded associate exceeding overall framework for assortment of traces from an unlimited variety of real users that support crowd sourcing. Our framework has been incontestable by analyzing the information collected within the central server victimization.

Polonium: Tera-scale graph mining and inference for malware detection.
Authors: D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos.
Year: 2011.
Description: In this paper, author developed four malicious applications, and evaluated ability to notice new malware supported samples of renowned malware. Author evaluated many mixtures of anomaly detection algorithms, feature selection technique and also the variety of high options so as to seek out the mixture that yields the most effective performance in detecting new malware in android application. Result shows that the projected framework is effective in detecting malware on mobile devices normally and on android applications specifically.

Fair Play: Fraud and malware detection in Google play
Authors: Mahmudur Rahman, Mizanur Rahman, Bogdan Carbunar, Duen Horng
Chau.
Description: In this paper, author proposes a proactive theme to identify zero-day android malware. Without using malware samples and their signatures, our scheme is actuated to assess potential security risks exposed by untrusted apps. Specifically, we have developed a automatic system referred to a risk ranker to scalably analyze whether a specific app exhibits malicious behaviour (e.g,launching a root exploit or causing background SMS messages).

Paper Name: Discovering opinion spammer groups by network footprints. In Machine Learning and Knowledge Discovery in Databases
Authors: Junting Ye and Leman Akoglu
Year: 2015.
Description: In this paper, we have studied a way to conduct effective risk communication for mobile devices. This has emerged jointly on the quickest growing operative systems. In Gregorian calendar year 2012, Google announced that four hundred million devices are activated, with one million devices being activated daily. The Google Play crossed more than fifteen billion downloads including year 2012, and was adding around one billion downloads per month from December 2011 to December 2012.

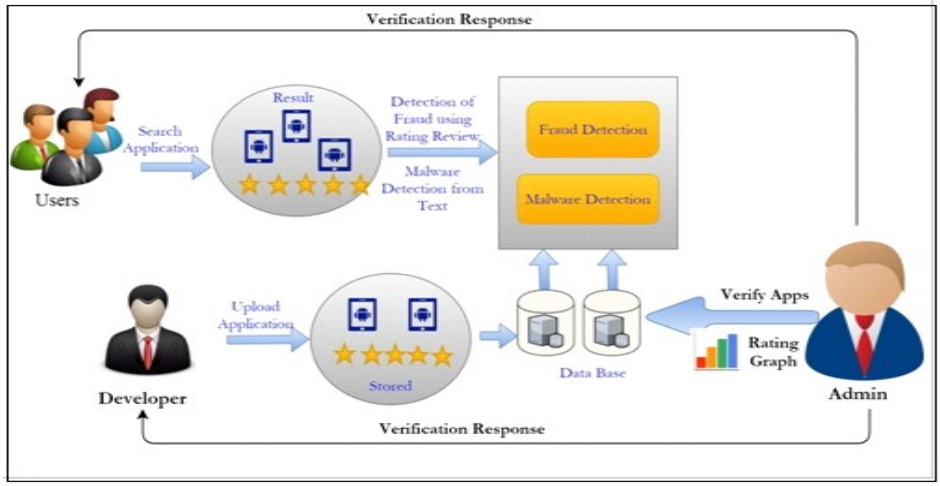
IV. EXISTING SYSTEM

Previous mobile malware detection work has targeted on dynamic analysis of app executables and static analysis of code and permissions. However, recent android malware analysis discovered that malware evolves quickly to bypass anti-virus tools.

- **Disadvantage**

Existing system was not able to detect malware before the installation of application.

V. ARCHITECTURE OF PROPOSED SYSTEM



In proposed system user and developer both have to do the registration. Developer will login into the system and upload the application .This application is stored in the database.Admin has the authority of accessing the database and reviewing accordingly using PCF algorithm. Admin verifies the app through the graph rating. After that user will login and search for the required application. The application uploaded by the developer is viewable to the user. The fraud application is detected using rating review and through this we come to know whether application is fraud or not. Malware detection refers to malicious software that exploits target system vulnerabilities that could be detected in application. Fraud detection detects background server-based processes that examine users and other defined entities access and behaviour patterns, and typically compares this information to a profile of what is expected.

VI. PROPOSED SYSTEM

We propose PCF (Pseudo clique Finder), an algorithm that takes input as the set of the reviews of associate app, organized by days, and a threshold value. PCF outputs a set of known pseudo-cliques and are shaped throughout contiguous time frames. Once the app has received a review , it finds the day's most promising pseudo-clique that begins with each review and then add different reviews to a candidate pseudo-clique. It manages to keep the pseudo set (of the day) with the very best density. With this work-in-progress, pseudo-clique adds different reviews whereas the weighted density of the new pseudo-clique is either equal or it exceeds to previous density. In proposed system user and developer have to register. Developer can login to the system and upload the application. Then user can login and rummage around the appliance. User will see the appliance uploaded by the developer. Once finding application that user needs to transfer user can choose search rank fraud detection and then he can check the malware within the application. Once user is satisfied, he can transfer the application.

➢ Advantages

The proposed system is able to detect malware before the installation.

VII. PROPOSED ALGORITHM

Input: Days, an array of daily reviews, and q, the weighted threshold density.

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

Step 1 for d :=0 d <days.size(); d++
Graph PC := new Graph();
bestNearClique(PC, days[d]);
c := 1; n := PC.size();
Step 2 for nd := d+1; d <days.size() c = 1; d++
bestNearClique(PC, days[nd]);
c := (PC.size() >n); endfor
Step 3 if (PC.size() >2)
allCliques := allCliques.add(PC); endfor
return
Step 4 function bestNearClique(Graph PC, Set revs)
if (PC.size() = 0)
Step 5 for root := 0; root <revs.size(); root++
Graph candClique := new Graph ();
candClique.addNode (revs[root].getUser());
Step 6 do candNode := getMaxDensityGain(revs);
if (density(c and Clique [c and Node) q))
candClique.addNode(candNode);
Step 7 while (candNode != null);
if (candClique.density() >maxRho)
maxRho := candClique.density();
PC := candClique; endfor;
else if (PC.size() >0)
Step 8 do candNode := getMaxDensityGain(revs);
if (density(candClique [candNode) q))
PC.addNode(candNode);
while (candNode != null);
return

CONCLUSION

Hence we developed PCF that reviews pseudo-cliques fashioned by reviewers with considerably overlapping co-reviewing activities across short time windows. We have introduced Fairplay, a system to find each deceitful and malware Google Play apps through search ranking using graph ratings.

REFERENCES

1. Bhaskar Pratim Sarma, Ninghui Li, Chris Gates, Rahul Potharaju, Cristina Nita-Rotaru, and Ian Molloy, "Android Permissions: a Perspective Combining Risks and Benets," in Proceedings of ACM SACMAT, 2012.
2. 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.

3. Mahmudur Rahman, Mizanur Rahman, Bogdan Carbunar, Duen Horng Chau, " Fair Play: Fraud and malware detection in Google play."
4. Junting Ye and Leman Akoglu. "Discovering opinion spammer groups by network footprints." in Machine Learning and Knowledge Discovery in Databases, 2015.
5. Takeaki Uno, "An efficient algorithm for enumerating pseudo cliques ," In Proceedings of ISAAC, 2007.
6. Steven Bird, Ewan Klein, and Edward Loper, " Natural Language Processing with Python," O'Reilly, 2009.
7. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, "Thumbs Up? Sentiment Classification Using Machine Learning Techniques," In Proceedings of EMNLP, 2002.