International Journal of Scientific Research in Computer Science, Engineering and Information Technology

Analyzing Fake Ranks In Google Play

M. Kishore[1], P. Shirisha[2]

[1]Student, Department Of Mca, Rcr Institution Of Management &Technology, Tirupati, Andhra Pradesh , India

[2]Assistant Proffesor, Department Of Mca, Rcr Institution Of Management &Technology Tirupati, Andhra Pradesh , India

ABSTRACT

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 false and malicious behaviors. 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. 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 (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.

I. 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 behaviors. 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 behaviors is impact: app popularity surges translate into financial benefits and expedited malware proliferation. Fraudulent developers frequently exploit crowdsourcing sites (e.g., Freelancer, Fiverr, BestAppPromotion) to hire teams of willing workers to commit fraud collectively, emulating realistic, spontaneous activities from unrelated people (i.e., "crowdturfing"). We call this behavior "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 analyzed using VirusTotal, 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 executables as well as static analysis of code and permissions. However, recent Android malware analysis revealed that malware evolves quickly to bypass anti-virus tools.

II. EXISTING SYSTEM

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. Previous mobile malware detection work has focused on dynamic analysis of app executables as well as static analysis of code and permissions. However, recent Android malware analysis revealed that malware evolves quickly to bypass anti-virus tools.

Disadvantages:

- ✓ Can't detect genuine reviews.
- ✓ Can't identify fraud users and malware indicators.
- ✓ Time taking process with executing app and analysis of code permission methods.

III. PROPOSED SYSTEM

In this, 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 syntactical and behavioral signals gleaned from Google Play app data, in order to identify doubtful apps. FairPlay achieves over 95% accuracy in classifying gold standard datasets of malware, fraudulent and rightful 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 forceful review operation. Malicious acts by picking out such trails. For instance, the high cost of setting up valid Google Play accounts forces fraudsters to reuse their accounts across review writing jobs, making them likely to review more apps in common than regular users. Resource constraints can compel fraudsters to post reviews within short time intervals.

Advantages:

- ✓ Can detect genuine reviews
- ✓ Can identify fraud users and malware indicators.
- ✓ Identifies forceful reviews operation.

Algorithm:

Algorithm 1 PCF algorithm pseudo-code.

```
Input: days, an array of daily reviews, and
       θ, the weighted threshold density
Output: allCliques, set of all detected pseudo-cliques
for d :=0  d < days.size(); d++
  Graph PC := new Graph();
  bestNearClique(PC, days[d]);
  c := 1; n := PC.size();
  for nd := d+1; d < days.size() & c = 1; d++
    bestNearClique(PC, days[nd]);
    c := (PC.size() > n); endfor
  if (PC.size() > 2)
    allCliques := allCliques.add(PC); fi endfor
return
function bestNearClique(Graph PC, Set revs)
  if (PC.size() = 0)
    for root := 0; root < revs.size(); root++
      Graph candClique := new Graph ();
      candClique.addNode (revs[root].getUser());
      do candNode := getMaxDensityGain(revs);
        if (density(candClique ∪ {candNode}) ≥ θ))
          candClique.addNode(candNode); fi
      while (candNode != null);
      if (candClique.density() > maxRho)
          maxRho := candClique.density();
          PC := candClique; fi endfor
  else if (PC.size() > 0)
      do candNode := getMaxDensityGain(revs);
        if (density(candClique ∪ candNode) ≥ θ))
          PC.addNode(candNode); fi
      while (candNode != null);
return
```

The Pseudo Clique Finder (PCF) algorithm:

We propose PCF (Pseudo Clique Finder), an algorithm that exploits the observation that fraudsters hired to review an app are likely to post those reviews within relatively short time intervals (e.g., days). PCF (see Algorithm 1), takes as input the set of the reviews of an app, organized by days, and a threshold value θ. PCF outputs a set of identified pseudo-cliques with $\rho \geq \theta$, that were formed during contiguous time frames. For each day when the app has received a review (line 1), PCF finds the day's most promising pseudo-clique (lines3 and 12 – 22): start with each review, and then greedily add other reviews to a candidate pseudo-clique; keep the pseudo clique (of the day) with the highest density. With that "working- progress" pseudo-clique, move on to the next day (line 5): greedily add other reviews while the weighted density of the new pseudo-clique equals or exceeds _ (lines 6 and 23 – 27). When no new nodes have been added to the work-in-progress pseudo-clique (line 8), we add the pseudo clique to the output (line 9), then move to the next day (line 1). The greedy choice (getMaxDensityGain, not depicted in Algorithm 1) picks the review not yet in the work-in progress pseudo-clique, whose writer has written the most apps in common with reviewers already in the pseudo clique.

IV. CONCLUSION

We have introduced FairPlay, a system to detect both fraudulent and malware Google Play apps. Our experiments on a newly contributed longitudinal app dataset, 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.

V. 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]. Stephanie Mlot. 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, Manish Mohanlal, Haitao Zheng, and Ben Y. Zhao. Serf and Turf: Crowdturfing 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]. VirusTotal - 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 Yael Weiss. Andromaly: a Behavioral Malware Detection Framework for Android Devices. Intelligent Information Systems, 38(1):161-190, 2012.

[15]. Michael Grace, Yajin Zhou, Qiang Zhang, Shihong Zou, and Xuxian Jiang. 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: a Perspective Combining Risks and Benefits. In Proceedings of ACM SACMAT, 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 detectmalware in android. In International Joint Conference CISIS12-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 worstcase time complexity for generating all maximal cliques and computational experiments. Theor. Comput. Sci., 363(1):28-42, October 2006.

[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 UsingMachine 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 DataMining, KDD '14, pages 1524-1533, New York, NY, USA, 2014. ACM