www.ijiarec.com

ISSN:2348-2079

Volume-6 Issue-3

International Journal of Intellectual Advancements and Research in Engineering Computations

A machine learning approach to android malware detection

N.Sathya[1], Dr.P.Sumitra[2]

[1]M.Phil, Research Scholar (Full-Time), [2]Assistant Professor.

PG and Research Department of Computer Science and Applications Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam.

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.

Keywords: Search rank fraud, Malware detection

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, BestApp Promotion) 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 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 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

Author for correspondence:

M.Phil, Research Scholar (Full-Time), PG and Research Department of Computer Science and Applications Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam.

RELATED WORK

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

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.

Proposed system

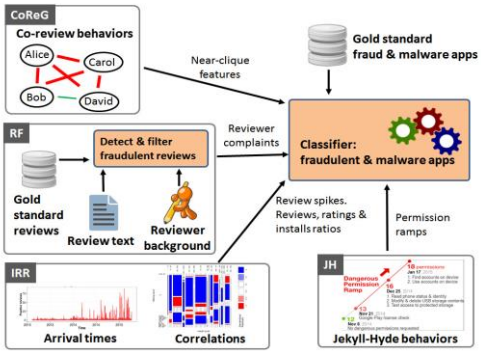
In this, we introduce FairPlay, a novel system that discovers and leverages traces left behind by fraudsters, to detect both malware and apps subjected to search rank fraud. FairPlay correlates review activities and uniquely combines detected review relations with syntactical and behavioral signals gleaned 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.

Architecture diagram



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.

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. SummerCon 2012, New York, 2012.
[12]. VirusTotal - Free Online Virus, Malware and URL Scanner. https: //www.virustotal.com/, Last accessed on 2015.
[13]. Iker Burguera, Urko Zurutuza, and Simin Nadjm-Tehrani. Crowdroid: Behavior-Based Malware Detection System for Android. In Proceedings of ACM SPSM, 2011, 15–26. ACM.
[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), 2012, 161–190.
[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, 2014.
[19]. Yajin Zhou and Xuxian Jiang. Dissecting Android Malware: Characterization and Evolution. In Proceedings of the IEEE S&P, 2012, 95–109. IEEE.
[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, 2013, 289–298. Springer.
[24]. Junting Ye and Leman Akoglu. Discovering opinion spammer groups by network footprints. In Machine Learning and Knowledge Discovery in Databases, 2015, 267–282. Springer.
[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), 2006, 28–42.
[28]. Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. 3111, 2004, 260–272.
[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), 1990, 179–191.
[36]. Leo Breiman. Random Forests. Machine Learning, 45, 2001, 5–32.
[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, 2014, 1524–1533, New York, NY, USA. ACM