International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 203-210 www.ijaem.net ISSN: 2395-5252

Securing Android App Markets via Modeling and Predicting Malware Spread between Markets

[1]kamal J &Mahesh J , [2]kaladevi P
[1]Students, K S Rangasamy College Of Technology,TiruchengodeTamilnadu.
[2]Assistant Professor, CSE Dept,K S Rangasamy College Of Technology,TiruchengodeTamilnadu.

Submitted: 01-05-2021 Revised: 09-05-2021 Accepted: 10-05-2021

ABSTRACT:
The absolute most hazardous web assaults, for example, Cross-Site Scripting and SQL infusion, abuse weaknesses in web applications that may acknowledge and handle information of unsure cause without legitimate approval or sifting, permitting the infusion and execution of dynamic or space explicit language code. These assaults have been continually besting the arrangements of different Security announcement suppliers notwithstanding the various countermeasures that have been proposed in the course of recent years. In this paper, we give an investigation on different guard systems against web code infusion assaults. We propose a model that features the key shortcomings empowering these assaults, and that gives a typical viewpoint to examining the accessible guards. We at that point arrange and dissect a set of 41 recently proposed guard's dependent on their exactness, execution, arrangement, security, and accessibility attributes.

I. DIGITAL MEDIA

Google Play (formerly Android Market) is a digital sharing service operated and developed by Google. It serve as the official app store for the Android operating system, allowing users to browse and download applications developed with the Android software development kit (SDK) and published through Google. Google Play also serve as a digital media store, presenting music, magazines, books, movies, and television programs. It formerly offered Google hardware devices for purchase until the beginning of a separate online hardware retailer, Google Store, on March 11, 2015.Applications are accessible through Google Play for free of charge or at a cost. They can be downloaded using Android device through the Play Store mobile app or by deploying the application to a gadget from the Google Play website. Applications exploiting hardware capabilities of a device can be targeted to users of devices with particular hardware components, such as a motion sensor (for motion-dependent games) or a front-facing camera (for online video calling). The Google Play store had 82 billion app downloads in 2016 and has reached over 2.7 million apps published in 2017. It has been the theme of multiple issues relating to security, in which malicious software has been agreed and uploaded to the store and downloaded by uers, with unreliable degrees of severity. Google Play was launched on March 6, 2012, bringing together the Android Market, Google Music, and the Google eBook store under one kind, marking a shift in Google's digital distribution strategy. The services working under the Google Play standard are: Google Play Books, Google Play Games, Google Play Movies & TV, Google Play Music, and Google Play Newsstand. 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 boguinstallation counts), while malicious developers use app markets as a launch pad for their malware.

II. ANDROID APP MARKETS

Commercial achievement of Android app market such as Google Play and the incentive model they offer to popular apps, create them interesting targets for fraudulent and malicious behaviors. Various fraudulent developers dishonestly increase the search rank and fame of their apps (e.g., through fake reviews and bogus installation counts), while malicious developers utilize app markets as a launch pad for their malware. The impulses for such behaviors are for: app popularity surges translate into economic benefits and expedited malware proliferation**.**On daily basis, an app leader board can be updated by

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 203-210 www.ijaem.net ISSN: 2395-5252

app store which display chart rankings of most admired apps, also it is an inspiring thing to make encouraged the growth of mobile apps. In fact, for promoting mobile phone Apps, leader board of apps is the mainly important way of up gradient in the market. An app should be ranked advanced depending upon how its chart of growth raise and progressively it can create number of downloads and ultimately high income. There were dissimilar ways to promote Apps in order to get peak position in App leader boards, the official one is white hat basis to promote their App to get famed and alternately more number of downloads. But there are also some illegal ways say black hat basis for bumping up the App by using some unreliable means used by fraudulent App developers to get famed in some short time period. This method usually implements "internet bots" or "human water armies" to increase the App downloads ratings and reviews in a very small time. Some are required points that are to restrict fraud, showed as given two constraints. The first restriction is that an app can be rated only one time from a user login and the second is implemented with the support of IP address that restricts the number of user logged per day. Finally, the proposed system will be evaluated with real world App data which is to be collected from the App Store for a long-time period called historical records.In the existing system, from the collected historical records, the top event and leading session of an app is recognized. There are two main steps for mining leading sessions. First, need to notice leading events from the App's historical ranking records. Second, need to combine adjacent leading events for constructing leading sessionCareful inspection shows that the mobile Apps are not constantly at top most places in leader board. But only in some time period called leading event which form different leading sessions means ranking fraud mostly arise in this leading session.Then from the user hypercritical feedback, three different types of evidences are collected namely ranking based evidence, rating based evidence and review based evidence. As they are based on evidences collected from app data; the one of the mostly judgment by people is rating based evidences which can be used to rate the app while downloading it or rate it later seeing its performance. It is significant proof to judge the app. But as discussed above there are some techniques with help of which the rating can get increases by doing fraud. So, another judged evidence based technique is review based evidence; which finds the exact specification of app whether it is good or bad app to download. In Review Based Evidences, in addition to ratings, most of the App stores also permit users to write some textual comments as App reviews. So, people may sure about downloading that particular app by reading comments specified in review part and also give their view about that app. Due to the huge number of apps, it is hard to search ranking fraud for every apps; so, it is main to have a scalable way to automatically notice ranking fraud without using any benchmark information. The efforts of Android markets to identify and remove malware are not always successful. Google Play uses the Bouncer system to remove malware. However, out of the 7,756 Google Play apps which are analyzed using VirusTotal, 12 percent (948) were flagged by at least one anti-virus tool and 2 percent(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 tool

III. OBJECTIVES

We propose FairPlay, a system that leverages the above observations to efficiently detect Google Play fraud and malware.Fraudulentbehaviors 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 fraudFairPlay correlates review activities and uniquely combines detected review relations with linguistic and behavioural signals gleaned from Google Play app data (87K apps, 2.9M reviews, and 2.4M reviewers, collected over half a year).Our major contributions are: A fraud and malware detection approach. To detect fraud and malware, we propose and generate 28 relational, behavioral and linguistic features that we use to train supervised learning algorithms.

• We formulate the notion of co-review graphs to model reviewing relations between users. We develop PCF, an efficient algorithm to identify temporally constrained, co-review pseudo-cliques — formed by reviewers with substantially overlapping co-reviewing activities across short time windows.

• We use temporal dimensions of review post times to identify suspicious review spikes received by apps; we show that to compensate for a negative review, for an app that has rat

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 203-210 www.ijaem.net ISSN: 2395-5252

FAKE REVIEWS IN GOOGLE PLAY AND GOOGLE APP STORE

Crowd-sourcing systems create a real task to existing security mechanisms deployed to guard Internet services. Many of these protection strategies rely on the assumption that malicious activity is generated robotically by using computerized programs. Thus they would operate badly or be easily bypassed when attacks are generated by actual users working in a crowd-sourcing system. Through measurements, it indicates shocking evidence showing that not only do malicious crowd-sourcing systems exist, but they are rapidly growing in both user base and total income. Use entire crawls to extract facts about the size and operational structure of these crowdturfing systems. Examine details of campaigns offered and performed in these sites, and assess their end-to-end efficiency by running active, benign campaigns of own. Finally, study and evaluate the source of personnel on crowdturfing websites in different countries. Results reveals that campaigns on these systems are effective at reaching users, and their ongoing growth poses a concrete threat to online communities. Unexpected evidences were determined displaying that now not only malicious crowd-sourcing systems exist, but they are quickly increasing in both user base and profits. Because of their similarity with both traditional crowd-sourcing systems and as torturing behavior, they are called as crowdturfingsystems.

SCALABLE AND ACCURATE ZERO-DAY ANDROID MALWARE DETECTION

Malicious apps hide with-in other normal apps, which makes their detection difficult. Existing mobile anti-virus software are not sufficient in their reactive nature by relying on identified malware samples for signature mining. It describes a proactive technique to spot zero-day Android malware. Without relying on malware samples and their signatures, this scheme is stimulated to evaluate possible security risks posed by means of these untrusted apps. Specifically, an automated system called RiskRanker to scalably have a look at a particular app that exhibits risky behavior (e.g., launching a root exploit or sending background SMS messages). The output is then used to make a prioritized list of reduced apps that merit further investigation

PROBABILISTIC GENERATIVE MODELS FOR RANKING RISKS OF ANDROID APPS

One of Android's core protection methods against malicious apps is a risk communication method which, before a user installs an app, warns the user about the permissions the app requires, trusting that the user will make the correct judgment. This approach has been shown to be useless as it presents the risk information of every app in a "stand-alone" fashion and in a way that needs too much practical awareness and some time to extort valuable information. Begin the notion of risk scoring and risk ranking for Android apps, to build up risk communication for Android apps, and recognize three desired data for an efficient risk scoring scheme. Probabilistic generative models are used for risk scoring schemes, and recognize several models, ranging from the simple Naive Bayes, to advanced hierarchical mixture models. Experimental results carried out using real-world datasets reveal that probabilistic general models considerably do better than other existing approaches, and that Naive Bayes models give a promising risk scoring approach.Probabilistic generative models have been used widely in a range of applications in machine learning, computer vision, and computational biology, to model complex facts. The foremost strength is to model functions in a large quantity of unlabeled information. Using these models, it is believed that some parameterized random procedure generates the app data and learn the model parameter based on the information. Then, calculate the probability of each app generated by the model. The risk score can be any function that is inversely related to the probability, so that lower probability interprets into a higher score.

MACHINE LEARNING APPROACHES

Android OS is one of the broadly used mobile Operating Systems. The range of malicious applications and malwares are growing continuously with the wide variety of mobile devices. A great number of commercial signature based tools are presented in the market which avoids the penetration and distribution of malicious applications. Various researches have been performed which claims that traditional signature based detection system work well up to certain degree and malware authors use numerous strategies to evade these tools. So given this state of affairs, there is a growing need for an alternative, really tough malware detection system to complement and resolve the signature based system. Latest substantial research centered on machine learning algorithms that examine features from malicious application and hire those features to categorize and spot strange malicious applications. This study summarizes the evolution of malware detection strategies based on machine

learning algorithms focused on the Android OS. Malware authors utilize many techniques to evade the detection such as (i) code obfuscation technique, (ii) encryption, (iii) including permissions which are not needed by the application, (iv) requesting for unwanted hardware's, (v) download or update attack in which a benign application updates itself or update another application with malicious payload, which is tough to detect. This also encourages the need for new studies on other detection techniques, including machine learning techniques. Many studies have shown that machine learning algorithms to identify the malicious activities are successful in detecting them with very high accuracy.

PERMISSION USAGE TO DETECT MALWARE IN ANDROID

Android devices have appeared these days and the number of programs available for this operating system has advanced exponentially. Google already has its Android Marketplace where applications are structured and it is far at risk to misuse. In fact, malware writers put in malicious applications into this market, but additionally among other different markets. Therefore, PUMA, a new method for detecting malicious Android applications via machine learning strategies by analyzing the extracted permissions from the application itself. In the last decade, users of these devices have experienced problems when installing mobile applications. There was not a centralized place where users could obtain applications, and they had to browse the Internet searching for them. When they found the application they wanted to install, the problems begin. In order to guard the device and avoid piracy, numerous operating systems, such as Symbian, employed an authentication system based on certificates that brought some inconveniences for the users (e.g., they could not install applications regardless of having bought them).The platforms have used special approaches to protect against this type of software. Machine learning techniques were substantially carried out for classifying applications that are specifically targeted on generic malware detection. Besides, several strategies have been proposed to categorize applications specifying the malware class; e.g., Trojan, worms, virus; and, even the malware family.applications which are used by many users, from recognized publishers, have other attributes that contains their legitimacy and good reputation. Bad applications, on the other hand, come from mysterious publishers, have appeared on few computers, and have other attributes that indicate poor reputation. The application status is computed by leveraging tens of terabytes of information anonymously contributed by the millions of users participating in the worldwide Norton Community Watch program. These nameless data contain significant characteristics of the applications running on their systems.

LARGE SCALE MALWARE DETECTION BY MINING FILE-RELATION GRAPHS

The increasing complexity of malicious software calls for new defensive techniques are harder to avoid and are able to guard users from novel threats. Aesop is an algorithm that identifies malicious executable files. Aesop leverages locality-sensitive hashing to calculate the strength of those inter-file associations to create a graph, on which it performs large scale inference by propagating information from the labeled files (as benign or malicious) to the preponderance of unlabeled files. Computer security providers identify the need to respond with better security against novel threats. The role of this 0-day threat protection is to limit the efficiency of malware's window, so that malicious files are detected soon after their first appearance. A new dangerous measure of achievement is a vanishingly small false positive rate, as labelinga original file as malicious can have vast consequences, mainly if it is a popular file or one that is essential to the constancy of the system, as in the case of operating system and driver files.

PUMA: Permission Usage to Detect Malware in Android

The presence of mobile devices has increased in our lives offering almost the same functionality as a personal computer. Android devices have appeared lately and, since then, the number of applications available for this operating system has increased exponentially. Google already has its Android Market where applications are offered and, as happens with every popular media, is prone to misuse. In fact, malware writers insert malicious applications into this market, but also among other alternative markets. Therefore, in this paper, we present PUMA, a new method for detecting malicious Android applications through machine-learning techniques by analysing the extracted permissions from the application itself.methodology for the security evaluation within third-party Android MarketplacesThis paper aims to evaluate possible threats with unofficial Android marketplaces, and geo-localize the malware distribution over three main regions:

China; Europe; and Russia. It provides a comprehensive review of existing academic literature about security in Android focusing especially on malware detection systems and existing malware databases. Through the implementation of a methodology for identification of malicious applications it has been collected data revealing a 5% of them as malicious in an overall analysis. Furthermore, the analysis shown that Russia and Europe have a preponderance of generic detections and adware, while China is found to be targeted mainly by riskware and malware.

IMASHUP:MASHUP-BASED FRAMEWORK FOR SERVICE COMPOSITION

The Web has undergone a tremendous change from a primarily publication platform towards a participatory and "programmable" platform, where a large number of heterogeneous Web-delivered services (including SOAP and RESTful Web services, RSS and Atom feeds) are emerging. It results in the creation of Web mashup applications with rich user experiences. However, the integration of Web-delivered services is still a challenging issue. It not only requires the developers' tedious efforts in understanding and coordinating heterogeneous service types, but also results in the time-consuming development of user interfaces. In this paper, we propose the iMashup composition framework to facilitate mashup development and deployment. We provide a unified mashup component model for the common representation of heterogeneous Web-delivered service interfaces. The component model specifies necessary properties and behaviors at both business and user interface level. We associate the component model with semantically meaningful tags, so that mashup developers can fast understand the service capabilities. The mashup developers can search and put the proper mashup components into the Web browser based composition environment, and connect them by data flows based on the tag-based semantics. Such an integration manner might prevent some low-level programming efforts and improve the composition efficiency. A series of experimental study are conducted to evaluate our framework.

OPERATING SYSTEMS FOR INTERNETWARE: CHALLENGES AND FUTURE DIRECTIONS

An operating system is an essential layer of system software that is responsible for resource management and application support on a computer system. As the evolvement of computer systems, the concept of OSs has also been evolved into many new forms beyond the traditional OSs such as Linux and Windows. We call this new generation of OSs as ubiquitous operating systems (UOSs). Among many new types of UOSs, we are particularly interested in the operating systems for Internetware, i.e., Internetware Operating Systems. Internetware is a paradigm for new types of Internet applications that are autonomous, cooperative, situational, evolvable, and trustworthy. An Internetware OS represents our perspective on the OS for future Internet-based applications. This paper discusses the examples, technical challenges and our recent effort on Internetware OSs, as well as our vision on the future of Internetware OSs. We believe that, in the foreseeable future, Internetware OSs will become ubiquitous and could be built for many different types of computer systems and beyond.

TOWARD UBIQUITOUS OPERATING SYSTEMS: A SOFTWARE-DEFINED PERSPECTIVE

In recent years, operating systems have expanded beyond traditional computing systems into the cloud, IoT devices, and other emerging technologies and will soon become ubiquitous. Despite the apparent differences among existing OSs, they all have in common so-called "software-defined" capabilities-namely, resource virtualization and function programmability. Resource virtualization and function programmability also lie at the heart of so-called "software-defined" systems including software-defined networks (SDNs),2 software-defined storage (SDS), and software-defined datacenters (SDDCs). Just as a traditional OS manages a hardware system with software abstractions and provides runtime support for applications, we believe that future OSs will provide all of the software-defined capabilities for emerging technologies. Thus, an SDN is an OS for networking hardware, while a software-defined cloud is an OS for the cloud. We refer to these OSs as ubiquitous operating systems (UOSs).

INTERNETWARE: SOFTWARE PARADIGM FOR INTERNET COMPUTING

Due to its open, dynamic, constantly changing nature, the Internet computing environment exhibits characteristics that call for new software development technologies, reflecting the pattern of ever-evolving paradigms throughout computing history. A software paradigm (also called a programming paradigm) describes a software model and its construction from the perspective of software engineers or

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 203-210 www.ijaem.net ISSN: 2395-5252

programmers.1 A software model specifies the forms, structures, and behaviors of software entities as well as their collaborations. As Figure 1 shows, software collaborations typically occur via predefined or dynamic interactions among software entities. In a structured paradigm, these entities are procedures, and their interactions occur through predefined procedural calls. In an object-oriented paradigm, object interactions occur through message passing. In a component-based paradigm, component interactions occur through connectors. However, in the Internet computing environment, requirements are often unclear or undetermined before collaboration occurs. For example, handling a swine flu epidemic requires cooperation among numerous organizations and individual software and services, such as airlines, hotels, hospitals, and mobile phone service providers. Similar scenarios arise in both emergency situations (such as earthquakes, typhoons, or snow disasters) and during major international or national events (such as the Olympic Games or the National Day celebration). The requirements involved in handling these types of situations call for new technologies that can support on demand collaborations among software entities. Chinese researchers have proposed Internetware,1-5 a software paradigm that provides a set of technologies for developing applications to meet computing requirements in the Internet environment. The "China's National Research and Development Framework" sidebar offers more information on the current status of projects related to the development of Internet ware.

END-TO-END PERFORMANCE ISOLATION THROUGH VIRTUAL DATACENTERS

The lack of performance isolation in multi-tenant data enters at appliances like middleboxes and storage servers results in volatile application performance. To insulate tenants, we propose giving them the abstraction of a dedicated virtual datacenter (VDC). VDCs encapsulate end-to-end throughput guarantees—specified in a new metric based on virtual request cost—that hold across distributed appliances and the intervening network. We present Pulsar, a system that offers tenants their own VDCs. Pulsar comprises a logically centralized controller that uses new mechanisms to estimate tenants' demands and appliance capacities, and allocates datacenter resources based on flexible policies. These allocations are enforced at end-host hypervisors through multi-resource token buckets that ensure tenants with changing workloads cannot affect others. Pulsar's design does not require changes to applications, guest OSes, or appliances. Through a prototype deployed across 113 VMs, three appliances, and a 40 Gbps network, we show that Pulsar enforces tenants' VDCs while imposing overheads of less than 2% at the data and control plane.

OPERATING SYSTEMS FOR INTERNETWARE: CHALLENGES AND FUTURE DIRECTIONS

—This paper presents our vision on present and future operating systems for Internetware, i.e., Internetware Operating Systems. Internetware is a paradigm for new types of Internet applications that are autonomous, cooperative, situational, evolvable, and trustworthy. An Internetware system consists of a set of autonomous software entities distributed over the Internet, together with a set of connectors to enable various collaborations among these entities. An Internetware OS represents our perspective on the OS for future Internet-based application. This paper discusses the technical challenges and our recent effort on Internetware OSs, as well as our vision on the future of Internetware OSs. We believe that, in the foreseeable future, Internetware OSs will become ubiquitous and could be built for different scales of computer systems and beyond.

DATADRIVEN COMPOSITION OF SERVICE-ORIENTED SITUATIONAL WEB APPLICATIONS

The convergence of Services Computing and Web 2.0 gains a large space of opportunities to compose "situational" web applications from web-delivered services. However, the large number of services and the complexity of composition constraints make manual composition difficult to application developers, who might be non-professional programmers or even end-users. This paper presents a systematic data-driven approach to assisting situational application development. We first propose a technique to extract useful information from multiple sources to abstract service capabilities with a set tags. This supports intuitive expression of user's desired composition goals by simple queries, without having to know underlying technical details. A planning technique then exploits composition solutions which can constitute the desired goals, even with some potential new interesting composition opportunities. A browser-based tool facilitates visual and iterative refinement of composition solutions, to finally come up with the satisfying outputs. A series of experiments demonstrate the efficiency and effectiveness of our approach.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 5 May 2021, pp: 203-210 www.ijaem.net ISSN: 2395-5252

PROGRAMMING SITUATIONAL MOBILE WEB APPLICATIONS WITH CLOUD-MOBILE CONVERGENCE: AN INTERNETWARE-ORIENTED APPROACH

Mobile Web applications (a.k.a., Web apps) stand for an important trend for next-generation Internet-based software. Currently popular mobile Web apps need to be adapted to various and ever-changing contexts and personalized user requirements. Based on our over-decade research experiences and practice on the Internetware paradigm, this position article describes an Internetware-oriented approach to designing, developing, and deploying situational mobile Web apps, by synthesizing the resources and services of mobile and cloud. Guided by a novel Service-Model-View-Controller (SMVC) software model, a mobile Web app is organized into a well-defined structure that facilitates adaptation including online/offline data access, computation offloading, user interface optimization, hybrid composition, etc. We provide efficient runtime support spanning mobile and cloud to make mobile Web apps more flexibly adaptive. The proof-of-concept evaluation demonstrates that our approach can benefit end-users with optimized user experience of mobile Web apps.

IV. METHODOLOGY

In case of the existing system the fraud is detected after the fraud is done that is, the fraud is detected after the complaint of the Mobile apps holder. And also now a days lot of online purchase are made so we don't know the person how is using the Mobile apps online, we just capture the IP address for verification purpose. So there need a help from the cyber crime to investigate the fraud. In the experiments, we validate the effectiveness of the proposed system, and show the scalability of the detection algorithm as well as some regularity of ranking fraud activities. Due to the huge number of mobile Apps, it is difficult to manually label ranking fraud for each App, so it is important to have a scalable way to automatically detect ranking fraud without using any benchmark information. Therefore, detecting ranking fraud of mobile Apps is actually to detect ranking fraud within leading sessions of mobile Apps. Specifically, we first propose a simple yet effective algorithm to identify the leading sessions of each App based on its historical ranking records..

V. PROPOSED SYSTEM

In proposed system, we present A Novel Ensemble Learning using PCA. Which does not require fraud signatures and yet is able to detect frauds by considering a Mobile apps holder's spending habit. The details of items purchased in Individual transactions are usually not known to any Fraud Detection System (FDS) running at the bank that issues detecting Mobile apps to the Mobile apps holders. The types of goods that are bought in that transaction are not known to the FDS. It tries to find any anomaly in the transaction based on the spending profile of the Mobile apps holder, activity monitoring.A Java web crawler was developed to download 970 positive reviews and 710 negative reviews randomly.

DATA PRE-PROCESSING

Previous studies revealed that pre-processing of text messages can improve the performance of text classification.

APP FEATURE IDENTIFICATION

As product reviews are about product features the product features are good indicators in classifying the sentiment of product reviews for product review based sentiment classification.

GUARANTEED RATING BASED EVIDENCES

The ranking based evidences are useful for ranking fraud detection. However, sometimes, it is not sufficient to only use ranking based evidences.

REVIEW BASED EVIDENCES:

Besides ratings, most of the App stores also allow users to write some textual comments as App reviews. Such reviews can reflect the personal perceptions and usage experiences of existing users for particular mobile Apps.

FRAUD APPS BLOCKING

One of the most important perspectives of App ranking fraud. Specifically, before downloadingmobile App, users often firstly 5, read its historical reviews to ease their decision making,

REFERENCES

[1]. Google Play. [Online]. Available: https://play.google.com/

[2]. E. Siegel, "Fake reviews in Google Play and Google App Store," Appentive, Seattle, WA, USA, 2014.

[3]. Z. Miners. (2014, Feb. 19). "Report: Malware-infected Android apps spike in the Google Play store," PC World. Available: http:// www.pcworld.com/article/2099421/report-malwareinfectedandroid- apps-spike-in-the-google-play-store.html

[4]. S. Mlot. (2014, Apr. 8). "Top Android App a Scam, Pulled From Google Play," PCMag. Available: http://www.pcmag.com/ article2/0,2817,2456165,00.asp
[5]. D. Roberts. (2015, Jul. 8). "How to spot fake apps on the Google Play store," Fortune. Available: http://fortune.com/2015/07/08/ google-play-fake-app/
[6]. A. Greenberg (2012, May 23). "Researchers say they snuck malware app past Google's 'Bouncer' Android market scanner,"Forbes Security, [Online]. Available: http://www.forbes.com/sites/andygreenberg/ 2012/05/23/
[7]. J. Oberheide and C. Miller, "Dissecting the Android Bouncer,"presented at the SummerCon2012, New York, NY, USA, 2012.
[8]. X. Liu, G. Huang, Q. Zhao, H. Mei, and M. B. Blake, "imashup: a mashup-based framework for service composition," Science China Information Sciences, vol. 57, no. 1, pp. 1–20, 2014.
[9]. January 2013DOI: 10.1007/978-3-642-33018-6_30In book: International Joint Conference CISIS'12-ICEUTE' 12-SOCO' 12 Special Sessions (pp.289-298)Publisher: Springer Berlin HeidelbergAuthors:Borja Sanz
[10]. A methodology for the security evaluation within third-party Android Marketplaces November 2017Digital Investigation DOI: 10.1016/j.diin.2017.10.002 Project: Distributed and Robust Sharing of Data within Cloud-based Architecture. Authors: William J Buchanan
[11]. Burguera, I., Zurutuza, U., Nadjm-Tehrani, S.: Crowdroid: behavior-based malware detection system for android. In: Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, ACM (2011) 15–26
[12]. Operating Systems for Internetware: Challenges and Future Directions 2018 Hong Mei; Yao Guo et.al
[13]. H. Mei and Y. Guo, "Toward ubiquitous operating systems: A software-defined perspective," Computer, vol. 51, no. 1, pp. 50–56, January 2018.
[14]. H. Mei and J. Lv, Internetware: A New Software Paradigm for Internet Computing. Springer, 2016.
[15]. S. Angel, H. Ballani, T. Karagiannis, G. O'Shea, and E. Thereska, "End-to-end performance isolation through virtual datacenters," in Usenix Conference on Operating Systems Design and Implementation, 2014, pp. 233–248
[16]. X. Liu, Y. Ma, G. Huang, J. Zhao, H. Mei, and Y. Liu, "Datadriven composition of service-oriented situational web applications," IEEE Transactions on Services Computing, vol. 8, no. 1, pp. 2–16, 2015.
[17]. H. Mei, "Understanding "software-defined" from an os perspective: technical challenges and research issues," Science China Information Sciences, vol. 60, no. 12, p. 126101, 2017.
[18]. H. Mei and Y. Guo, "Development and present situation of Internetware operating systems," Science & Technology Review, vol. 34, no. 14, pp. 33–41, 2016, (in Chinese).