Digital Investigation 13 (2015) 1–14

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

APK Auditor: Permission-based Android malware detection system

Kabakus Abdullah Talha [a, *], Dogru Ibrahim Alper [b], Cetin Aydin [b]

[a] *IT Center, Abant Izzet Baysal University, Bolu 14280, Turkey*
[b] *Department of Computer Engineering, Gazi University, Ankara 06500, Turkey*

ARTICLE INFO

Article history:
Received 24 July 2014
Received in revised form 11 January 2015
Accepted 15 January 2015
Available online 6 March 2015

Keywords:
Android
Smartphone
Mobile security
Malware
Static analysis

ABSTRACT

Android operating system has the highest market share in 2014; making it the most widely used mobile operating system in the world. This fact makes Android users the biggest target group for malware developers. Trend analyses show large increase in mobile malware targeting the Android platform. Android's security mechanism is based on an instrument that informs users about which permissions the application needs to be granted before installing them. This permission system provides an overview of the application and may help gain awareness about the risks. However, we do not have enough information to conclude that standard users read or digital investigators understand these permissions and their implications. Digital investigators need to be on the alert for the presence of malware when examining Android devices, and can benefit from supporting tools that help them understand the capabilities of such malicious code. This paper presents a permission-based Android malware detection system, *APK Auditor* that uses static analysis to characterize and classify Android applications as benign or malicious. *APK Auditor* consists of three components: (1) A signature database to store extracted information about applications and analysis results, (2) an Android client which is used by end-users to grant application analysis requests, and (3) a central server responsible for communicating with both signature database and smartphone client and managing whole analysis process. To test system performance, 8762 applications in total, 1853 benign applications from Google's Play Store and 6909 malicious applications from different sources were collected and analyzed by the system developed. The results show that *APK Auditor* is able to detect most well-known malwares and highlights the ones with a potential in approximately 88% accuracy with a 0.925 specificity.

Introduction

Today, Android is the most widely used mobile operating system in the world. The reasons for being favored by users can be listed as: (1) Being an open source software, (2) being supported by Google, (3) being an advanced programmable software framework used to develop mobile applications in popular programming language Java (4) being open to customization. According to the International Data Corporation's latest report, Android dominates the market with 84.7% share with more than 300 million shipments in Q2 2014.[1] Android's senior vice president Sundar Pichai recently revealed that Android has reached over 1 billion devices.[2] Hugo Barra, Android product development vice president, declares that Google Play Store has more than 1 million applications.[3] Lookout

* Corresponding author. Tel.: +90 5353787341.
E-mail address: talha.kabakus@ibu.edu.tr (K.A. Talha).

[1] http://www.idc.com/getdoc.jsp?containerId=prUS25037214.
[2] https://twitter.com/sundarpichai/status/374933465998708736.
[3] http://readwrite.com/2013/07/24/google-play-hits-one-million-android-apps.

http://dx.doi.org/10.1016/j.diin.2015.01.001
1742-2876/© 2015 Elsevier Ltd. All rights reserved.

reports that Play Store grows three times faster than its biggest rival Apple Store (Lookout, 2011). Parallel to this popularity, the number of malicious applications is increasing continually (Felt et al., 2011a). Sophos Mobile Security Threat Report reveals that number of Android malware has increased by 600% in last 12 months (Mobile Security Threat Report, 2014). Kaspersky Labs reports that Android is the main target of malicious software, attracting a whopping 98.05% of known malware.[4] Similarly, Symantec's latest Internet Security Threat Report reviews that “the number of mobile threats that track users doubled in 2013 and mobile malware seemed almost solely focused on the Android platform” (Internet Security Threat Report, 2014). Hence, mobile security applications can only protect systems from certain kinds of malicious programs. Most of these security applications compare newly installed applications against a repository that stores malware signatures (Guido et al., 2013). This “blacklisting” technique has known weaknesses that can be exploited by malware distributors (Vidas et al., 2011b). This blacklisting approach never suffices to ascertain whether an application is malicious or not since applications get updated over time and their purposes may change during these updates. A piece of Android malware can surreptitiously escalate privileges by modifying system files, send short messaging service (SMS) messages, make calls, share location information through global positioning system (GPS) or internet and collect excessive amounts of personal information. The Android security mechanism does not set a limit on how much of the system resources an application can use such as central processing unit (CPU), memory and local drive, etc. This is a critical vulnerability point for malicious applications.

Android's security model significantly relies on permission-based mechanisms (Barrera et al., 2010; Felt et al., 2011b). Permissions are requested by applications and they are required to access application's interfaces/data that are defined in its manifest file (Enck et al., 2011). They are used to inform users about application's capabilities (“Android Security Overview,” n.d.). Android security framework does not block or review applications through their permissions. Instead of that, Android informs users about which permissions an application wants to be granted when user initiates installation process. Evaluation process is handled by user and s/he can continue on the installation or cancel it if s/he is not comfortable with the permissions. These permissions are not shown to users at any time other than installation (Felt et al., 2012). According to the research by Felt et al., only 17% of participants are interested in these permissions; 42% of them are unaware about them (Felt et al., 2012). Some researchers on the other hand speculated that most of the time these permissions are ignored and not understood by users (Enck et al., 2009; Felt et al., 2011b; Kelley et al., 2012; King et al., 2011). Past studies on smartphone users' privacy concerns have primarily focused on location tracking and sharing (Barkuus et al., 2003; Consolvo et al., 2005; Kelley et al., 2011; Lindqvist et al., 2011; Sadeh et al., 2009). They are just 2 of 145 permissions that Android 4.4 (KitKat) defines. At this point, users need to be informed and guided by the application about these permissions before installation.

Android applications are packaged as Android application (APK) files which contain manifest file (`AndroidManifest.xml`), compiled Java classes and application resources. We have developed an Android malware detection system based on permission analysis through APK files, named *APK Auditor*. The system developed uses static analysis techniques based on permissions in order to characterize and extract profiles for Android applications. *APK Auditor* contains three main components: (1) An Android client, (2) a signature database that contains extracted analysis, (3) a central server that communicates with both the signature database server and the clients.

This paper is structured as follows: Related works section presents the related works. APK Auditor section discusses the materials and methods, *APK Auditor*'s architecture with its components. Results and discussion section presents the experiments and Conclusions section presents the conclusions and the future works.

Related works

DroidMat (Wu et al., 2012) detects Android malware through static analyst paradigm. It takes into consideration some static information including permissions, deployment of components, intent message passing and Application Programming Interface (API) calls for characterizing Android applications' behaviors. After application characteristics are extracted, K-means algorithm is used to enhance malware modeling capability. Numbers of clusters are decided by Singular Value Decomposition (SVD) method on the low rank approximation. At last, the system classifies applications as benign or malicious using k-Nearest Neighbors (kNN) algorithm.

MADAM (a Multi-level Anomaly Detector for Android Malware) (Dini et al., 2012) is a multi-level Android malware detector that concurrently monitors Android at the kernel-level and user-level to detect real malware infections using machine learning techniques to distinguish between standard behaviors and malicious ones. At kernel-level, *MADAM* evaluates system calls, running processes, free random access memory (RAM) and CPU usages. At user/application level, it evaluates idle/active status, keystrokes, dialed numbers, sent/received SMS and Bluetooth/Wi-Fi analysis.

Guido et al. (2013) presents a real time malware detection system called *Periodic Mobile Forensics (PMF)* that contains a smartphone client, a central server, a database and an analysis framework. The smartphone client sends changed file system data to central server in order to allow expensive forensic processing and offline application of traditional tools and techniques rarely applied to mobile environment. The analysis framework identifies changes to important system partitions, recognizes the changes in the file system, including file deletions and finds persistent and triggering mechanisms in newly installed applications. Unlike *APK Auditor*, the analysis performed by PMF does not focus on application permissions.

[4] http://www.kaspersky.com/about/news/virus/2013/number-of-the-year.