IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Volume-4, Issue-4 E-ISSN: 2347-2693

Android Malicious apps and Malware Security: A Review

Vishal Kumar Gujare[1*] and Praveen Malviya[2]

[1*,2]*Department of Computer Science and Engineering, RGPV, India*

www.ijcseonline.org

Received: Mar/25/2016 Revised: Apr/04/2016 Accepted: Apr/17/2016 Published: Apr/30/2016

Abstract- Smartphones are becoming the important devices for us in the post-PC era, which aid in our daily tasks with the useful functionalities such as Internet, GPS, cameras, NFC (Near Field Communication) and accelerometers. Millions of users are using android phones and android application is becoming more and more popular. Among all the mobile platforms available, Android has become the more targeted one. The exponential growth of the Android platform in the recent years has made it a main target of cyber-criminals. As a result, the amount of malware for Android is constant and rapidly growing. This exponential growth of malware given, there is a need for new detection models designed to specifically target Android malware in order to better protect the end-users and, eventually, to counter the rise of Android malware itself. This is probably due to the fact that Android is the most widespread platform and to some technical particularities, such as the fact that Android applications are really easy to reverse engineer and to modify/repackage. This paper focus on the survey of security threats in Android platform. We gives the survey on malware and malicious applications in Android.

Keywords— Smartphone, Android, Malicious applications, Security, Malware

I. INTRODUCTION

In recent years, we have witnessed an exponential growth in mobile devices adoption. According to CNN [1] (CNN, 2011), from 2008 to 2011 the number of smartphone shipments has tripled, and this number is still increasing. A smartphone is actually a small computer built on a mobile operating system with more advanced computing capability and connectivity than a feature phone. Smartphones are becoming the important devices for us in the post-PC era, which aid in our daily tasks with the useful functionalities such as Internet, GPS, cameras, NFC (Near Field Communication) and accelerometers. In addition, smartphone applications are available in multiple application stores or markets (e.g., Google Play [2], Amazon Android App Store [3] and iOS App Store [4]) that further spur the popularity of smartphones. The smartphone ecosystem encompasses smartphones hardware and software platform, applications (apps) running on top of the platform, as well the infrastructural components (e.g., networks and the cloud).

Android is the world's most popular mobile platform. It's the largest installed base of any mobile platform and growing fast. Millions of users are using android phones and android application [5] is becoming more and more popular. The exponential growth of the Android platform in the recent years has made it a main target of cyber-criminals. As a result, the amount of malware for Android is constant and rapidly growing. This exponential growth of malware given, there is a need for new detection models designed to specifically target Android malware in order to better protect the end-users and, eventually, to counter the rise of Android malware itself. All these mobile devices, together with the wealth of sensitive information carried with them (e.g. personal and bank information, GPS location, SMS messages and emails), have inevitably driven to the evolution of malware targeting mobile platforms. The fact that malware can be successful even without exploiting system vulnerabilities [6] (especially Rogue-AVs and Trojans) and the drastic change of malware landscape from a for-fun activity to a profit-driven criminal business. Furthermore, many companies still do not provide protection for their employees' mobile devices.

Android phones users can download useful applications and install them on smartphones to accomplish tasks such as checking emails, browsing the Internet, and etc.. Unfortunately, not all applications from those markets are "clean". Some applications, known as "malware", are hostile or intrusive as they may disrupt computer operation, gather sensitive information, or gain access to private systems. According to the survey more than 80% of smartphones remained unprotected from the malware attacks. Android is a particular focus since its ecosystem isn't controlled as tightly as iOS or Windows Phone. Hence, how to effectively detect malware from millions of applications has been a hot research topic in the recent years.

The paper is organized as follows. Section II represents background related to Android platform and ecosystem. Section III provides literature survey of Android privacy and

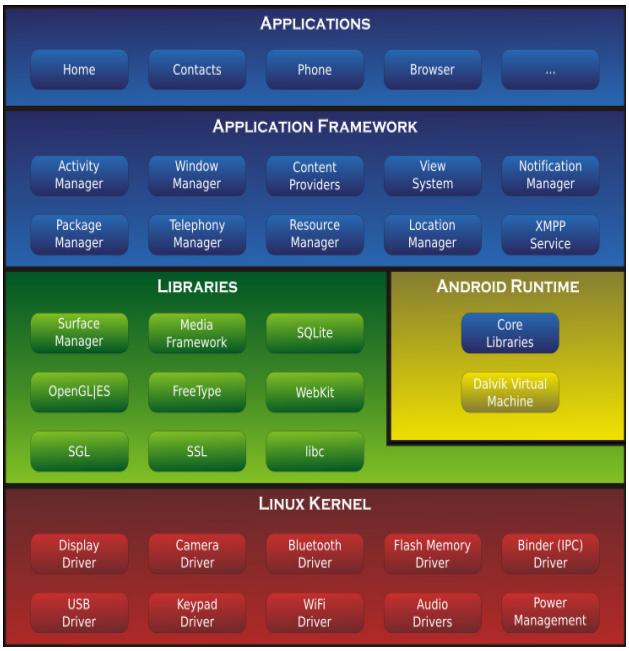
security and malware. Section IV provides the related work. Section V concludes the paper.

II. BACKGROUND

Android is a Linux-based operating system developed for smart phones or tablet computers. It is a stack of software that includes operating system, middleware and libraries and APIs written in C [7]. It was developed by Google and Open Handset Alliance in July, 2005. Android is an open source and Google releases the source code under Apache license. This open source and free license allow the manufacturers and the enthusiastic developers to freely develop and modify their applications in Java-like language that utilizes Google-developed Java libraries. The Android applications are developed using the Android software development kit (SDK). The SDK includes a comprehensive set of development tools which includes a debugger, software libraries, a handset based emulator which is based on QEMU (Quick Emulator) and tutorials. The integrated development environment (IDE) which is officially supported for Android apps development is Eclipse which uses the Android Development Tools (ADT) plugin. The software stack consists of a custom Linux system, the Dalvik Virtual Machine (VM), and apps running on top of the VM. Each app runs in its own copy of the VM with a different user id, hence apps are protected from each other.

Android is an Operating System (OS) primarily designed for touchscreen mobile devices (i.e. smartphones and tablets). It was initially designed and developed by the namesake Android Inc., a startup bought by Google in 2005. Android was officially unveiled in 2007, and its source code was released by Google under the Apache License. (Google, 2013). The Android OS is built on top of a modified version of the Linux kernel. In particular, Android is a multi-user Linux system where each application is a different user and runs in its own Linux process. (The Android Open Source Project, 2013). On top of the Linux kernel there are the native libraries, such as OpenGL and WebKit, and the Dalvik Virtual Machine (VM), an open source Virtual Machine originally written by Dan Bornstein (who named it after the Icelandic village of Dalvík) and optimized to run Java applications in mobile devices.

The Dalvik VM uses a register-based architecture (with its own 16-bit instruction set) and Just-In-Time (JIT) compilation to run Dalvik Executable (dex) files. From the version 4.4, Android also supports ART Virtual Machine, a new experimental Virtual Machine that uses Ahead-Of-Time (AOT) compilation to run oat executable files. (The Android Open Source Project, 2013m).



On top of the Android system architecture there are the applications (commonly referred as apps), which run on an application framework made of Java-compatible libraries based on Apache Harmony.

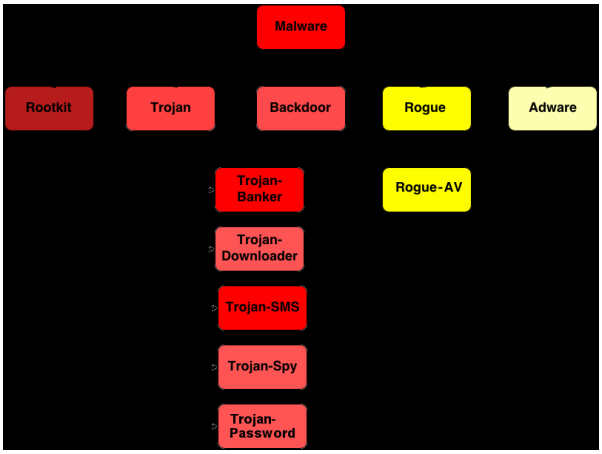
User applications are installed in the /data/app directory and their data are saved into the /data/data directory, while the system applications are installed in the /system/app and, starting from Android 4.4 KitKat, in the /system/priv-app ones. Without the root privileges, a third-party application cannot access to the other applications' /data/data directory. Furthermore, each application in Android runs in its own process, with its own instance of the Dalvik VM. Therefore, an application's code runs in isolation from other applications.

A permission model, explained shortly, protects sensitive resources, e.g., the hardware and stored data. In this model, resources are protected by permissions, and only apps holding the permission (which is granted
when the app is installed) are given access to the permission-protected resource.

API Levels. To facilitate app construction, the Android platform provides a rich framework to app developers. The framework consists of Android packages and classes, attributes for declaring and accessing resources, a set of Intents, and a set of permissions that applications can request. This framework is accessible to apps via the Android application programming interface (API). The Android platform has undergone many changes since its inception in 2008, and each major release forms a new API level.

III LITERATURE SURVEY

Malware[8] is an umbrella-term for various types of unwanted piece of software and/or executable codes that are used to perform unauthorized, often harmful, actions on computing devices. Malware can be firstly classified according to their behaviour, i.e. their malicious activities.



In Android most of the existing malware types are directly inherited from the desktop space (e.g. Adware or Rogue-AV), even if some of them have additional capabilities due to the mobile space (e.g. all the contacts information stored in one place, the ability to send SMS messages and to perform phone calls). However, there are also others malware types that are unique to mobile space (e.g. Trojan-SMS). Most of the Android malware are actually "Trojanised" versions of legitimate applications.

Backdoor: it is a particular application - also known as Remote Administration Tool (RAT)[9] - that allows an attacker to take control of the device (without the user consent or knowledge) and perform various malicious activities from a remote location, such as: intercept phone calls and SMS messages, make phone calls, send SMS messages, open websites, download and install other applications, delete files, collect and send information back to the attacker.

Rogue (also known as FraudTool): it is a deceptive application that pretends to be a well-known or trusted software in order to steal money and/or confidential data. An example of a Rogue is the Rogue://Android/FakeFlash.A, which tries to lure its victims to pay 5 Euro to download the Adobe Flash Player app. Adobe does not offer the "standalone" app on Google Play Store anymore.

The Android platform contains two main app categories: third-party and pre-installed. Third-party apps are available for download from Google Play (previously known as Android Market [10]) and other app stores, such as Amazon. These Android apps are developed by individual third-party developers, which can include software companies or individuals around the world. Malicious apps, designed for nefarious purposes, form a special class of third-party apps. Pre-installed apps come along with the devices from the vendors; they are developed and loaded in the devices before the devices ever reach the user in the market. These apps can be designed and configured exclusively per device model depending on the needs of particular manufacturers and phone service carriers by the vendor developers.

IV RELATED WORK

[11]Android has been a major target of malicious applications (malapps). How to detect and keep the malapps out of the app markets is an ongoing challenge. One of the central design points of Android security mechanism is permission control that restricts the access of apps to core facilities of devices. However, it imparts a significant responsibility to the app developers with regard to accurately specifying the requested permissions and to the users with regard to fully understanding the risk of granting certain combinations of permissions. Android permissions requested by an app depict the app's behavioral patterns. In order to help understanding Android permissions, in this paper, we explore the permission-induced risk in Android apps on three levels in a systematic manner. First, we thoroughly analyze the risk of an individual permission and the risk of a group of collaborative permissions. We employ three feature ranking methods, namely, mutual information, correlation coefficient, and T-test to rank Android individual permissions with respect to their risk. [11] use sequential forward selection as well as principal component analysis to identify risky permission subsets. Second, we evaluate the usefulness of risky permissions for malapp detection with support vector machine, decision trees, as well as random forest. Third, we in depth analyze the detection results and discuss the feasibility as well as the limitations of malapp detection based on permission requests. We evaluate our methods on a very large official app set consisting of 310 926 benign apps and 4868 real-world malapps and on a third-party app sets. The empirical results show that our malapp detectors built on risky permissions give satisfiedperformance (a detection rate as 94.62% with a false positive rate as 0.6%), catch the malapps' essential patterns on violating permission access regulations, and are universally applicable to unknown malapps (detection rate as 74.03%).

In [12] As the risk of malware is sharply increasing in Android platform, Android malware detection has become an important research topic. Existing works have demonstrated that required permissions of Android applications are valuable for malware analysis, but how to exploit those permission patterns for malware detection remains an open issue. In this paper, we introduce the contrasting permission patterns to characterize the essential differences between malwares and clean applications from the permission aspect. Then a framework based on contrasting permission patterns is presented for Android malware detection. According to the proposed framework, an ensemble classifier, *Enclamald*, is further developed to detect whether an application is potentially malicious. Every

contrasting permission pattern is acting as a weak classifier in *Enclamald*, and the weighted predictions of involved weak classifiers are aggregated to the final result. Experiments on real-world applications validate that the proposed *Enclamald* classifier outperforms commonly used classifiers for Android Malware Detection.

Malware detection of Android applications based on permissions is a rather new field of research in recent years. [12] pointed out that applications can be granted more permissions than they actually need, and the unused permissions may be leveraged for malicious goals. They presented a static analysis tool for identifying the essential permissions of an application to avoid granting those unnecessary ones [13] presented a dynamic analysis platform for reconstructing sensitive behaviors in Android applications, which can be used to examine the internal sensitive behaviors of the application. [14] proposed a feature-based learning framework for Android malware detection.

They represented each application as a single instance with binary permission and API call features with a class label indicates whether the application is a clean or a malware. Then classification models can be constructed from the data using machine learning methods such as SVM, decision tree and Bagging. [15] proposed *DroidRisk*, a framework for quantitative security risk assessment of both Android permissions and applications based on permission request patterns from clean applications and malwares. The assessment can help users easily understand the risk of an application to be malicious and pay more attention to those permissions with high risk levels. [15] defined contrasting permission pattern to analyze the difference between permissions required by malwares and that by clean applications, their research is the prior work of [15].

V. CONCLUSION

A smartphone is actually a small computer built on a mobile operating system with more advanced computing capability and connectivity than a feature phone. Android is the world's most popular mobile platform. It's the largest installed base of any mobile platform and growing fast. Millions of users are using android phones and android application is becoming more and more popular. The exponential growth of the Android platform in the recent years has made it a main target of cyber-criminals. As a result, the amount of malware for Android is constant and rapidly growing. This exponential growth of malware given, there is a need for new detection models designed to specifically target Android malware in order to better protect the end-users and, eventually, to counter the rise of Android malware itself. This paper focused on the survey of security threats in Android platform. We reviewed on malware and malicious applications in Android.

REFERENCES

[1] BHAS N. Press Release: More Than 80% of Smartphones Remain Unprotected from Malware and Attacks, Juniper Research Finds [EB/OL].[2014-02-23].http://www.juniperresearch.com/viewpressrelease.php?pr=404.

[2] D. Research. (2013). Global Smartphone Shipments to Reach 1.24 Billion in 2014. [Online]. Available: http://www.digitimes.com/news/a20131125PD218.html

[3] I. D. Corporation. (2013). Android Pushes Past 80% Market Share While Windows Phone Shipments Leap 156.0% Year Over Year in the Third Quarter, According to IDC. [Online]. Available:http://www.idc.com/getdoc.jsp?containerId=prUS24442013

[4] H. Peng et al., "Using probabilistic generative models for ranking risks of Android apps," in Proc. ACM Conf. CCS, 2012, pp. 241–252.

[5] BARRERA D, KAYACIK H G, OORSCHOT P C V, et al. A Methodology for Empirical Analysis of Permission-Based Security Models and Its Application to Android[C]//Proceedings of the 17th ACM Conference on Computer and Communications Security: Oct 4-8, 2010, Chicago, Illinois, USA: ACM, 2010: 73-84.

[6] FELT A P, GREENWOOD K, WAGNER D. TheEffectiveness of Application Permissions[C]//Proceedings of the 2nd USENIX Conference on Web Application Development: Jun 15-16, 2011, Portland, Oregon, USA: USENIX Association, 2011: 7-7.

[7] FELT A P, HA E, EGELMAN S, et al. Android Permissions: User Attention, Comprehension, and Behavior[C]//Proceedings of the 8th Symposium on Usable Privacy and Security: Jul 11-13, 2012, Washington, D.C., USA: ACM, 2012: 1-14.

[8] CHEN Y, XU H, ZHOU Y, et al. Is This App Safefor Children?: A Comparison Study of Maturity Ratings on Android and Ios Applications[C]//Proceedings of the 22nd International Conference on World Wide Web: May 13-17, 2013, Rio de Janeiro, Brazil: International World Wide Web Conferences Steering Committee, 2013: 201-212.

[9] ZHOU Y, JIANG X. Dissecting Android Malware: Characterization and Evolution[C]//Proceedings of the 2012 IEEE Symposium on Security and Privacy: May 21-23, 2012, San Francisco, California, USA: IEEE Computer Society, 2012: 95-109.

[10] B. Uscilowski. (2013). Symantac White Paper: Mobile Adware and Malware Analysis. [Online]. Available: symantec.com/content/en/us/enterprise/media/security_response/whitepapers/madware_and_malware_analysis.pdf

[11] Wei Wang, Xing Wang, Dawei Feng, Jiqiang Liu, Zhen Han, and Xiangliang Zhang, Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 11, NOVEMBER 2014, pp-1869-1883

[12] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android permissions: A perspective combining risks and benefits," in Proc. 17th ACM SACMAT, 2012, pp. 13–22.

[13] XIONG Ping, WANG Xiaofeng, NIU Wenjia, ZHU Tianqing, LI Gang, Android Malware Detection with Contrasting Permission Patterns, PROTECTING COMMUNICATIONS INFRASTRUCTURE AGAINST CYBER ATTACKS, IEEE 2014, pp-1-14

[14] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie, "WHYPER: Towards automating risk assessment of mobile applications," in Proc. 22nd USENIX Secur. Symp., 2013, pp. 527–542.

[15] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A study of Android application security," in Proc. 20th USENIX Secur. Symp., 2011,p. 21.

