Volume: 09 Issue: 04 Pages: 3506-3509(2018) ISSN: 0975-0290

Android Malware Detection in Official and Third Party Application Stores

Sangeeta Rani
Reseacrch Scholar, I.K Gujral Punjab Technical University, Punjab, India
Email: parthj6@gmail.com
Kanwalvir Singh Dhindsa
Baba Banda Singh Bahadur.Engg.College, Fatehgarh Sahib,Punjab,India
Email: kanwalvir.singh@bbsbec.ac.in

ABSTRACT

Android is one of the most popular operating system for mobile devices and tablets. The growing number of Android users and open source nature of this platform has also attracted attackers to target Android devices. This paper presents the static and dynamic analysis of the Android applications in order to detect malware. In this work, we have performed permission based and behavioural based filtering of Android applications with the help of malware analysis tools. Our results revel that 80% of the applications request for dangerous permissions. 13% applications consist of malicious activities. Most of the applications are interested in the device data like contact lists, IMEI, IMSI, SMS etc. These results clearly indicate the need for better security measures for Android apps.

Keywords - **Android Malware, Static analysis, Dynamic analysis, Permissions, Applications**

Date of Submission: Jan 10, 2018 Date of Acceptance: Jan 23, 2018

I. INTRODUCTION

There has been an enormous growth in the number of Android applications available both on official and unofficial stores in the past few years. Due to this attackers are also targeting Android to steal and misuse private information of the device users and to gain financial benefits. Android provides security at different levels. One of the important security measures is its permission based framework that provides access controls for various applications. At the time of installation users grant a set of permissions for every application, which control its access to certain resources. However most of the users mostly ignore these permissions while installing applications, which may cause malware attacks. Therefore, this work identifies a set of dangerous permissions and detects applications that may cause risky behaviour on the basis of these permissions. Further these risky applications are analyzed to detect malware using dynamic analysis.

The rest of the paper is as follows: Section II presents related work. Section III provides methodology of work. Results are presented in section IV. Lastly, conclusion is in section V.

II. RELATED WORK

Many researchers have analyzed and detect malware based on permission system. Felt et al. examined the effectiveness of installation time permission grant methods [1]. Barrera et al. performed an experimental analysis on the effectiveness of Android's permission sets. They also discussed some possible upgradation for Android's permission model [2]. Felt et al. detected overprivileged permissions in Android applications available at official and third party stores [3]. Zhou et al. and Felt et al. presented the characteristics of Android malware after performing analysis of the malware samples [4] [5]. Zhou et al. and Grace et al. developed malware detection tools DroidRanger and RiskRanker respectively [6] [7]. These tools can detect both known and unknown malware. Wei et al. proposed different methods to secure device data after analyzing the permission model of third party applications [8]. Holavanalli et al. proposed a tool Flow Permissions that examines the flow of permission grant methods within the applications [9]. Zhang et al. developed VetDroid which is a dynamic analysis tool that helps to monitor the permission usage behaviour at run time [10]. Faruki et al. detected malware using static analysis method which was based on application's code and structure. Besides static analysis, dynamic analysis techniques are also popular to detect malware [11]. Rashidi et al. and Beresford et al. used dynamic analysis to understand the behaviour of an application at runtime [12] [13]. Burguera et al. developed Crowdroid for behavioural based malware detection [14]. They analyzed Android applications behaviour to differentiate between applications with same names but different behaviour.

III. METHODOLOGY

Present work focuses on malware detection using both static and dynamic analysis techniques. Static analysis refers to the analysis of the source code of the application without executing the code [15] [16]. Permission based analysis is an important static analysis technique that analyze AndroidManifest.XML file included in every application. As a first step every application is disassembled and analyzed using open source static analysis tool Androguard and ApkInspector. These tools generate AndroidManifest.XML file, from where the permissions are extracted. It also provides information about other components like ContentProviders, Services,

Volume: 09 Issue: 04 Pages: 3506-3509(2018) ISSN: 0975-0290

Broadcast receivers and Intents. The extracted permissions are then compared with a set of dangerous permissions mostly requested by the malware (a total of 20 permissions which is considered as the feature set [17]). If the application is requesting all dangerous permissions then it is marked as Riskware application. These Riskware applications are then further analyzed with the help of dynamic analysis tools. Dynamic analysis refers to detect the malicious activity performed by the application at runtime [18]. Dynamic analysis reports generated by online tools like VirusTotal, ScanDroid and NVISO ApkScan are analyzed. Run time activities like dynamic loading of code to install backdoors, connecting to Command and Control servers, stealing personal information of the users and sending it outside is considered as malicious activities and the application performing it is marked as malware.

A. *Dataset*

The following datasets have been used:

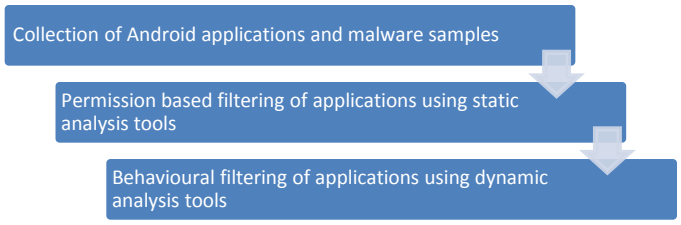
a) Android Applications

A total of 1900 benign applications have been collected with the help of a crawler and also manually. These applications are collected from Google Play store and third party application stores for Android.

b) Android Malware

This dataset consists of 100 Android applications detected as malware samples that cover common Android malware families available in the year 2016. These samples are collected from two online malware dumps malmr.com and AndroMalShare.

Figure 1. Implementation Process



IV. RESULTS

A. *Permission Based Filtering*

Using static analysis tools ApkInspecter and Androgaurd, AndroidManifest.XML file of the collected applications is extracted to perform permission based filtering. The behaviour of an application is strictly controlled by the permissions. Analysis results revel that 80% of the applications request dangerous permissions. Applications that request dangerous permission combinations are considered as risky applications. The frequently asked permissions by applications that can pose risky behaviour are as follows:

Table 1: Frequently requested permissions

INTERNET
READ_PHONE_STATE
WRITE_EXTERNAL_STORAGE
ACCESS_NETWORK_STATE
SEND_SMS
RECEIVE_SMS
RECEIVE_BOOT_COMPLETED
ACCESS_WIFI_STATE
ACCESS_FINE_LOCATION
INSTALL_SHORTCUT

Table 2: Riskware Applications

AppStore	Total Apps	Riskware
Play Store and Third Party Store	1900	1520

Out of total 1900 applications collected from official Play store and third party store 1520 applications request dangerous permissions. Thus these applications are marked as Riskware applications. INTERNET and READ_PHONE_STATE permission is requested by most of the applications. This permission is required by embedded ad libraries to work properly. The WRITE_EXTERNAL_STORAGE permission allows the application to read or write external storage. Malware use this permission to update or delete device data. SMS related permissions are also requested commonly. The RECEIVE_BOOT_COMPLETED permission assists the malware to run immediately after booting. The malware request ACCESS_NETWORK_STATE permission to access information like network availability, roaming or local networks etc. ACCESS_WIFI_STATE permission allows application to access wifi networks and the malware may use this information to hack the wifi network

B. *Behavioural Filtering*

Permission based filtering can result in false positives. So the next phase examines all the applications that ask dangerous permissions with dynamic analysis tools. Dynamic analysis works during the execution of the application [19]. It detects and maintains log about the file activities, short messages, network activities, loading of code performed by the application.

Table 3: Malware applications

AppStore	Total Apps	Riskware	Malware
Play Store and Third Party Store	1300	1520	350

The results of this analysis disclose the presence of malware in third party stores and even in official Google Play store applications. 12% of the collected applications from official store enclose malware. 28 % of third party store applications enclose malware.

The behaviour of these malware applications can be arranged in the following categories:

a) Invasively Collecting Personal Information

The first category includes applications that collect personal information of the user. 70% of the detected malware applications access personal information of

Volume: 09 Issue: 04 Pages: 3506-3509(2018) ISSN: 0975-0290

device such as IMEI, phone no, contact list, location and SMS.

b) Unsafely Fetching and Loading Dynamic Code

The second category include loading of dynamic from the internet. Dynamic loading of code may pose potential security threat due to two reasons. One reason is that static analysis tools cannot analyze dynamically loading of code reliably. Hence it can easily bypassing static analysis techniques. The second reason is that the downloaded code can be easily changed at any time. It is difficult to detect dynamic fetching of code. But it can be caught by monitoring the loading of classes by java.lang.reflect package. DexClassLoader class also permits applications to load random code which is not a part of applications package file. 36% of malware perform functionality with the help of dynamic loading of code.

c) Connecting Command and control server

Connection with C&C servers boosts the functionality of the malware. 58 % of the malware connect with C&C servers. These servers monitor the device continuously. Malware sends device information to these servers in different ways. For example InMobi malware detected in collected applications sends text messages to these servers. SMSSpy malware encodes data into JSON format and sends it to a remote server.

V. CONCLUSION

In this paper, we have analytically examined the security and privacy issues raised by Android malware. We analyzed 1900 applications collected from the official Android Market and third party Android stores. The results discovered threats to security and privacy exist on both official and third party stores. Most of the Android application request for dangerous permissions. Run time analysis of the applications revel threats range from collecting device information, connecting with command and control servers and dynamically loading of code without user's awareness. Android's permissions system cannot differentiate between actions performed by dynamic loaded code and those performed by the hosting application. Thus the current Android security system provides little indication of the existence of these threats within any given application. The results surly necessitate a more secure and robust Android security architecture.

REFERENCES

[1] A. P. Felt, K. Greenwood, and D. Wagner, The effectiveness of install-time permission systems for third-party applications, *Technical report*, University of California at Berkeley, UCB/EECS-2010-143, Dec 2010.

[2] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, A methodology for empirical analysis of permission-based security models and its application to android, *Proc. 17th ACM conference on Computer and communications security*, ACM, New York, 2010, 73–84.

[3] A.P. Felt , E. Chin., S. Hanna, D. Song , D. Wagner, Android permissions demystified, *Proc. 8th ACM conference on Computer and communications security*, New York, USA. 2011b. 627-638.

[4] Y. Zhou and X. Jiang, Dissecting android malware: Characterization and evolution, *Proc. IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2012, 95–109.

[5] A. P. Felt, M. Finifter, E. Chin, D. Wagner, A Survey of Mobile Malware in the Wild, *Proc. 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '11)*, ACM, New York, USA, 2011, 3-14

[6] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. Proceedings of the 19th Network and Distributed System Security Symposium, San Diego, CA, 317-326.

[7] M. Grace, Y. Zhou, Q. Zhang, S. Zou and X. Jiang , Riskranker: scalable and accurate zero-day android malware detection, *Proc. International Conference on Mobile Systems, Applications, and Services*, London, UK, 2012, 281–294.

[8] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, Malicious android applications in the enterprise: What do they do and how do we fix it?1, *Proc. IEEE 28th International Conference on Data Engineering Workshops (ICDEW),* 251–254, 2012.

[9] S. Holavanalli, D. Manuel, V. Nanjundaswamy, B. Rosenberg, F. Shen, S. Y. Ko, and L. Ziarek, Flow permissions for android, *Proc. 28th IEEE/ACM International Conference on Automated Software Engineering (ASE),* 2013, 652–657,2013.

[10] Y. Zhang, M. Yang, B. Xu, Z. Yang, G. Gu, P. Ning, X. S. Wang, and B. Zang, Vetting undesirable behaviors in android apps with permission use analysis, *Proc. 2013 ACM SIGSAC Conference on Computer & Communications Security*, 611–622, 2013.

[11]P. Faruki, V. Ganmoor, V. Laxmi, M. S. Gaur, and A. Bharmal, Androsimilar: Robust statistical feature signature for android malware detection, *Proc. International Conference on Security of Information and Networks (SIN'13)*, Aksaray, Turkey. ACM, 2013, 152–159

[12] B. Rashidi, C. Fung, and T. Vu, Recdroid: A resource access permission control portal and recommendation service for smartphone users, Proc. *2014 ACM MobiCom Workshop on Security and Privacy in Mobile Environments (SPME'14)*, Maui, Hawaii, USA, 2014, 13–18.

[13] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, Mockdroid: Trading privacy for application functionality

Volume: 09 Issue: 04 Pages: 3506-3509(2018) ISSN: 0975-0290

on smartphones, *Proc. 12th Workshop on Mobile Computing Systems and Applications (HotMobile'11*), Phoenix, Arizona, USA. ACM, 2011, 49–54.

[14] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, Crowdroid: Behavior-based malware detection system for android, *Proc. 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM'11)*, Chicago, Illinois, USA, 2011, 15–26.

[15] Z. Aung, and W. Zaw, Permission-based Android malware detection, *International Journal of Scientific & Technology Research, 2*(3), 2013

[16] S. Ryo, D. Chiba, and S. Goto, Detecting Android Malware by Analyzing Manifest Files, *Proc. Asia-Pacific Advanced Network* , 2013, 23-31

[17] S. Rani, and K. Dhindsa, Behavioural Characterization of Android Malware to Detect Similar Malware, *International Journal of Research in Electronics and Computer Engineering*, *5*(4), 2017, pp. 509-514

[18] A. Kapratwar, Static and Dynamic Analysis of Android Malware, *Proc. 1st International Workshop on FORmal methods for Security Engineering In conjunction with the 3rd International Conference on Information Systems Security and Privacy*, Porto, Portugal, 2017

[19] F. Yuhui, and X. Ning, The Analysis of Android Malware Behaviors, *International Journal of Security and Its Applications*, *9*(3), 2015, 335-346

Biographies and Photographs

Sangeeta Rani is a Research Scholar at the I.K Gujral Punjab Technical University, Jalandhar, Punjab. Her research interests include social network analysis, mobile phone security and network security. She is currently working as an Assistant Professor at the Mata Gujri College, Fatehgarh Sahib, Punjab, India.

Kanwalvir Singh Dhindsa is currently working as a Professor (CSE) and Incharge, ERP at the Baba Banda Singh Bahadur Engg., College, Fatehgarh Sahib (Punjab), India, with an overall experience of 17 years in the academia and research fields of computer science and IT, He is an avid researcher having guided dissertations of many MTech and PG students and he is currently guiding seven PhD scholars. He is also on the reviewer panel of many esteemed refereed and peer-reviewed journals. His current research interests include cloud computing, web engineering, big data, internet of things, database and security, and mobile computing. He is also life member-CSI, Fellow IETE, member IEI, and life member-ISCA