International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Mining Based Learning Framework for Android Malware Detection

[1]D.Sindhu, [2]V. Bakyalakshmi
M.Phil. Research Scholar[1], Associate Professor[2]
Department of Computer Applications, Sri Jayendra Saraswathy Maha Vidyalaya College of Arts & Science, , Coimbatore, Tamil Nadu, India

ABSTRACT

The Android malware threat has increased owing to the increase popularity of Android smartphones. The widespread adoption and contextually sensitive nature of smartphone devices has increased concerns over Android malware writers. Mining based learning framework is proposed for detecting malicious applications on Android devices. The system begins with analyzes only manifest files that are required to classify the Android applications into malware or benign applications. It realizes a lightweight approach for detection, and its effectiveness is experimentally confirmed by employing real samples of Android malware. The result shows that the new method can effectively detect Android malware, even when the sample is unknown.

Keywords : Android, malware, Manifest files, data mining.

I. INTRODUCTION

The smartphone has rapidly become an extremely prevalent computing platform, with just over 115 million devices sold in the third quarter of 2011, a 15% increase over the 100 million devices sold in the first quarter of 2011, and a 111% increase over the 54 million devices sold in the first quarter of 2010 [1,2]. Android's popularity among users has made the developers provide innovative applications (popularly called apps). Google Play, official Android app market hosts third-party developer apps with a nominal fee providing moderate control. Google Play hosts more than one million apps with large number of downloads each day. Unlike Apple market app store, Google Play does not verify uploaded apps manually. Instead, Google Play relies on Bouncer, a dynamic emulation environment to protect itself from malicious app threats. It would provide protection against threats, but cannot analyze the vulnerability of existing apps. Malicious apps may trick vulnerable apps to divulge user's private information that inadvertently harms the reputation of the latter. Moreover, Android does not recommend, but allows installation of third party apps on device, which has stirred up dozens of regional as well as international app-stores. However, protection and quality of apps available in third-party app stores is a matter of concern.

Android security solution providers report an alarming rise of malware from just three families and mere 100 samples in 2010, to more than hundred families with 0.12-0.6 million unique samples. The number of malicious apps uploaded on Virus Total is doubling every year. Malicious apps are using clever ways to bypass existing security mechanisms provided by Android OS as well as anti-malware products such as stealth techniques, dynamic execution, code obfuscation, repackaging and encryption. Existing malware propagate by employing above techniques to defeat signature-based approach used by anti-malware products. Thus, new mechanisms that adapt and provide timely response to such techniques are important. Proactive approaches are needed to detect unknown variants of known malware with less number of signature updates, in contrast to one signature for each known malware.

Malware app developers gain smartphone control by exploiting platform vulnerabilities, stealing sensitive user information, and getting monetary benefits by exploiting telephony services or creating botnet. Thus,

it is important to understand their operational activities, mode of working and usage pattern in recent past to devise proactive detection methods. Huge increase in malicious apps has forced anti-malware industry to carve out robust methods for efficient detection on device under existing constraints. Majority of anti-malware still employ retrospective signature based detection due to implementation simplicity and efficiency. Signature based methods can be easily circumvented through code obfuscation, necessitating a new signature for every malicious sample and that is why an anti-malware client has to regularly update its signature database. Due to limited processing capability and constrained battery power on a smartphone, cloud-based solutions for analysis and detection came into existence. Signature generation needs expertise and patience for each malware sample as it may incur false positives while detecting unknown variants of a known malware family. Due to increasing number of malware and their variants, there is a need to employ mining based detection incurring low false positive rate.

Some malicious behaviors of Android malware

Malware is usually motived by controlling mobile device without user intervention, such as:
(1) Privilege escalation to root,
(2) Leak private data or exfiltrate sensitive data,
(3) Dial premium numbers,
(4) Botnet activity, and
(5) Backdoor triggered via SMS.

The rest of the paper is organized as follows: Related works are described in section 2. Section 3 briefly describes the overview of the proposed detection framework. Section 4 shows the experimental results and performance evaluation. Section 5 concludes the paper.

II. RELATED WORKS

Analysis and detection of Android malware is a major research topic in recent years. Several concepts and techniques have been proposed to counter the increasing amount and sophistication of this Android malware. Most of the current detection mechanisms are mainly focusing Android permissions system.

The permissions may be required when an application is interacting with system resources, including calling system API functions and reading from and writing to file systems. The Android platform employs the permission system to restrict applications privileges to secure the users privacy-relevant information. In the current Android permission framework, accesses to critical resources on smartphones are controlled according to permissions given to applications at install-time. That is, each application must request for certain permissions for it to access system resources on a smartphone at install-time and the user of the smartphone should make a decision on whether or not to grant the permissions requested. Analysis of Android's permission security model has been done by various authors.

Frank et al [3] proposed a data mining method to analyze Android application permission requests. They have identified over 30 common patterns of permission requests by using matrix factorization techniques. Barrera et al [4] proposed a methodology to analyze the trends of permission requests from a dataset of 1,100 Android applications. They have used self-organizing maps to visualize which permissions are used in application with similar characteristics. Sanz et al [5] employs strings contained in disassembled Android applications, constructing a bag of words model in order to train machine learning algorithms to provide detection of malicious applications. In their work, Random Forest configured with 100 trees, obtained an accuracy of 92.04%. Advantage of this work is minimal dataset. The limitations of this work are that the operational overhead to scan string is high. Moreover the technique is not robust. It can be evaded by encrypting the malware.

Rassameeroj & Tanahashi [6] proposed an Android application analysis based on a permission security model using visualization techniques and clustering algorithms to reveal benign and malicious permission request combinations. They have evaluated their methodology on a dataset consisting of 999 samples. Peiravian & Zhu [7] developed a machine learning framework to analyze benign and malicious applications. They have extracted requested permissions and Application Programming Interface (API) calls from each application with similar characteristics and achieved highest accuracy of 96.88%. Our work differs from the above mentioned works in the sense, that we enrich the dataset by including requested permissions and used features

describing why these features are requested and used.

In this work, we conducted a thorough frequency analysis of permissions within benign and malware applications to extract the most significant patterns and employ feature selection technique to get the most relevant ones.

III. METHODOLOGY

In this section, we propose a permission analysis based learning framework for Android malware detection. The building block of the framework is given in fig. 1. It consists of three major parts. The first part decompresses Android Application Package (APK) file of an application and extracts AndroidManifest.xml files which are necessary for characterizing applications [8]. The second part carries out feature extraction, which include permissions extracted from AndroidManifest.xml. After the feature set generation, wrapper feature selection measure is used to select the most significant features. The last part includes machine learning techniques which are used to classify the collected data into benign and malicious.

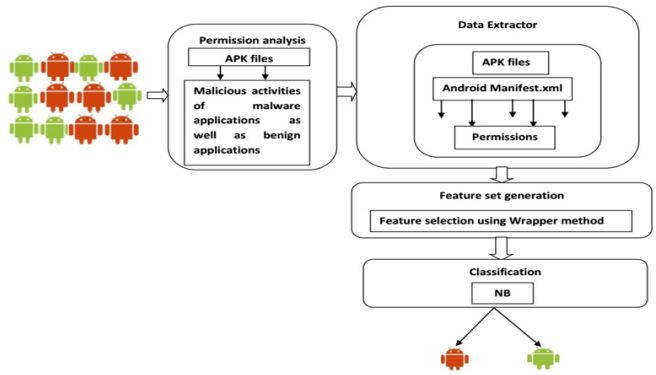

Figure 1. Learning framework for Android malware detection system

A.) Applications Permission Analysis

In order to identify the significant characteristics of applications, various categories of malicious and benign applications are analysed in terms of permissions. These features are normally defined in AndroidManifest.xml file. Each Android application must have a Manifest file in the root directory, which presents essential information about the application [9]. There are currently 147 Android permissions as of August 2014 that are classified into four different types such as Normal, Dangerous, Signature and Signature or System. Normal permissions do not require the user's approval and will be granted automatically by the Android platform. They can be viewed after the application has been installed. It doesn't possess any harmful consequences for the Android users.

Dangerous permissions are in need of users' authorization before starting the installation process. These are the only permissions displayed to the user upon installation. It provides access to users' personal sensitive data. Signature permissions have the highest privileges which are granted without the users knowledge provided the application is signed with the device manufacturer's certificate. SignatureOrSystem permissions are granted to those applications that are present in the Android system image or being signed with device manufacturer's certificate [10]. We are mainly focusing on normal and dangerous permissions categories since the Android malware comes with a repackaged version of legitimate applications.

B). Data Extractor and Permission frequency analysis

In this section, we review the different permissions that distinguish between malicious and benign applications. The various categories of Android malware applications and benign applications are collected from different malware applications repositories such as Contagio, Offensive, Vxheavens and VirusShare and google app store for benign applications. The collected Android malware applications perform all range of attacks such as phishing, banking-Trojan. We analyzed the permissions by the applications in order to measure their relevance in the Android malware detection process. We used the *Android Asset Packaging Tool* (aapt) to extract and decrypt the data from the AndroidManifest.xml file, provided by the Android SDK [11, 12]. Also we performed a thorough frequency study of different permissions of both applications in order to determine the significant permissions for malicious application detection. We compared top 11 permissions from 3833 samples of both normal and malicious applications to select the most significant permissions. The result of frequency analysis of normal and malicious applications is shown in fig.2.

C). Feature set generation and selection

In order to extract these features from each application we utilized aapt tool available within the set of tools provided by the Android SDK. Extracted features are passed through the wrapper feature selection measure to ensure the selection of the most discriminant features [13]. In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from our subset.

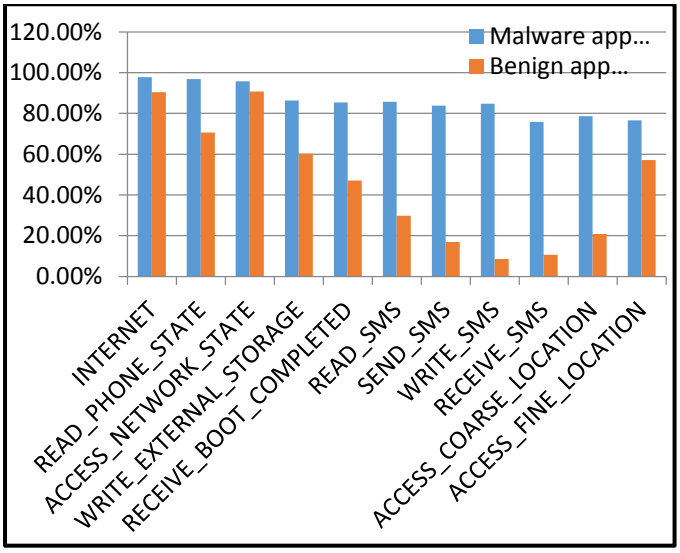

Figure 2. Frequency comparison of top 11 permissions

D). Naive Bayesian Classification

A Naive Bayesian classifier is a simple probabilistic classifier based on Bayes theorem with strong independence assumptions between the features [14, 15]. It assumes that the presence or absence of a particular feature of a class in unrelated to the presence or absence of any other feature. This classifier is highly scalable and it can work with small amount of data and can also accommodate a large number of samples make them a choice for Android botnet detection. Also, the selected features satisfy the conditional independence property of the Naive Bayesian classifier.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

We have evaluated our classifier with evaluation measures, such as accuracy, F-measure and false positive rate. The accuracy is percentage of correctly identified Android malware applications.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN}$$

Where,

True Positive (TP) =Number of applications correctly classified as malware.

False Positive (FP) = Number of applications incorrectly classified as malware.

True Negative (TN) = Number of applications correctly classified as normal.

False Negative (FN) = Number of applications incorrectly classified as normal.

F-Measure is a measure of test's accuracy, which measures the balance between precision and recall.

$$\text{F-Measure} = \frac{2 * Precision * Recall}{Precision + Recall}$$

False Positive Rate (FPR) is percentage of wrongly identified normal classes.

$$\text{Positive Rate (FPR)} = \frac{FP}{FP+TN}$$

The experimental results are given in table 2.

Measures	Values
Accuracy	98.10
F-Measure	0.974
False Positive Rate	0.065

Table 2. Experimental Results

From the table, one can observe that the naïve Bayesian classifier achieves highest accuracy of 98.10% and false positive rate of 0.065 when detecting the Android malware applications.

V. CONCLUSION

In this research paper we presented a learning based data mining framework to detect Android malware applications. The proposed framework begins with analyzing the malicious activities of malware applications and extracts significant permissions from every. By applying machine learning to classify applications as benign or malware. We showed 98.10% detection rate with a 0.065% false positive rate.

VI. REFERENCES

[1]. Christy Pettey and Holly Stevens. Gartner says 428 million mobile communication devices sold worldwide in first quarter 2011, a 19 percent

[2]. increase year-on-year. http://www.gartner.com/it/page.jsp?id=1689814.

[3]. Christy Pettey and Holly Stevens. Gartner says sales of mobile devices grew 5.6 percent in third quarter of 2011; smartphone sales increased 42 percent. http://www.gartner.com/it/page.jsp?id=1848514.

[4]. Frank, M., Dong, B., Felt, A. P., & Song, D. (2012, December). Mining permission request patterns from android and face book applications. In Data Mining (ICDM), 2012 IEEE 12th International Conference on (pp. 870-875). IEEE.

[5]. Barrera, D., Kayacik, H. G., van Oorschot, P. C., & Somayaji, A. (2010, October). A methodology for empirical analysis of permission-based security models and its application to android. In Proceedings of the 17th ACM conference on Computer and communications security (pp. 73-84). ACM.

[6]. Sanz, B., Santos, I., Laorden, C., Ugarte-Pedrero, X., Bringas, P. G., & Álvarez, G. (2013). Puma: Permission usage to detect malware in android. In International Joint Conference CISIS'12-ICEUTE´ 12-SOCO´ 12 Special Sessions (pp. 289-298). Springer Berlin Heidelberg.

[7]. Rassameeroj, I., & Tanahashi, Y. (2011, May). Various approaches in analyzing android applications with its permission-based security models. In Electro/Information Technology (EIT), 2011 IEEE International Conference on (pp. 1-6). IEEE.

[8]. Peiravian, N., & Zhu, X. (2013, November). Machine learning for android malware detection using permission and api calls. In Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on (pp. 300-305). IEEE.

[9]. Arp, D., Spreitzenbarth, M., Hubner, M., Gascon, H., Rieck, K., & Siemens, C. E. R. T. (2014, February). DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket. In NDSS.

[10]. Aafer, Y., Du, W., & Yin, H. (2013, September). Droidapiminer: Mining api-level features for robust malware detection in android. In International Conference on Security and Privacy in Communication Systems (pp. 86-103). Springer International Publishing.

[11]. Dini, G., Martinelli, F., Saracino, A., & Sgandurra, D. (2012, October). MADAM: a multi-level anomaly detector for android malware. In International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security (pp. 240-253). Springer Berlin Heidelberg.

[12]. Sahs, J., & Khan, L. (2012, August). A machine learning approach to android malware detection. In Intelligence and security informatics conference (eisic), 2012 european (pp. 141-147). IEEE.

[13]. Shabtai, A., Kanonov, U., Elovici, Y., Glezer, C., & Weiss, Y. (2012). "Andromaly": a behavioral malware detection framework for android devices. Journal of Intelligent Information Systems, 38(1), 161-190.

[14]. Yan, G., Brown, N., & Kong, D. (2013, July). Exploring discriminatory features for automated malware classification. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (pp. 41-61). Springer, Berlin, Heidelberg.

[15]. Lindorfer, M., Neugschwandtner, M., Weichselbaum, L., Fratantonio, Y., Van Der Veen, V., & Platzer, C. (2014, September). Andrubis--1,000,000 apps later: A view on current Android malware behaviors. In Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2014 Third International Workshop on (pp. 3-17). IEEE.

[16]. Huang, C. Y., Tsai, Y. T., & Hsu, C. H. (2013). Performance evaluation on permission-based detection for android malware. In Advances in Intelligent Systems and Applications-Volume 2 (pp. 111-120). Springer Berlin Heidelberg.