MOBILE MALWARE DETECTION USING CLASSFICATION TECHNIQUES

[1]ISIL KARABEY, [2]ONDER COBAN

[1,2]Computer Engineering Department, Erzurum, Turkey
[1]Erzurum Technical University, [2]Ataturk University
E-mail: [1]isil.karabey@erzurum.edu.tr, [2]onder.coban@atauni.edu.tr

Abstract- The number of applications for smart mobile devices is steadily growing with the continuous increase in the utilization of these devices. Security vulnerabilities such as seizure of personal information or the use of smart devices in accordance with different purposes by cyber criminals often arises through the installation of malicious applications on smart devices. Therefore, the number of studies in order to identify malware for mobile platforms has increased in recent years. In this study, permission-based model is used to detect the malicious applications on Android which is one of the most widely used mobile operating system. M0Droid data set has been analyzed using the Android application package files and permission-based features extracted from these files. In our work, permission-based model which applied previously across different data sets investigated for M0Droid data set and the experimental results has been expanded. While obtaining results, feature set analyzed using different classification techniques. The results shows that permission-based model is successful on M0Droid data set and Random Forests outperforms another methods. When compared to M0Droid system model, it isobtainedmuchbetter conclusions depend on success rate.

Keywords- Mobile Malware Detection, Permission data, Classification techniques, M0Droid.

I. INTRODUCTION

In recent years, smart mobile devices sucha s mobile phones, tablets, personal digital assistants (PDAs) have all become popularbecause of their processing capacity, smaller size and simpler design. Considering the fact that there are about 30 millions of smart phone users in Turkey, as soon as operating systems are concerned, the use of Android operating system in Turkey is 78.2% since May 2015, whereas it's the most commonly used mobile operating system worldwide with an average rate of 63.72% [1]. With an increased use of Android-based mobile devices, the number of malwares targeting these devices is also increasing day by day. According to the information obtained from the G Data Mobile Malware Report, a number of 440,267 new malwares were identified within the first quarter of 2015. When compared to those belonging to the fourth quarter of 2014, it was observed to increase by 6.4% [2]. Malwares which are embedded into the applications downloaded to mobile devices by means of internet may result in the device being captured by a cybercriminal, rendered dysfunctional or the privacy such as credit card information being stolen. The processes of detecting these malwares that normally cannot be recognized by users are usually performed by two different methods, namely static and dynamic methods. In static analyzing approach, the behavior of the application is tried to be perceived via resource codes form the applications, and then, malwares are detected using these behaviors. According to the dynamic analysis approach, application analyses are performed based on their operability on mobile devices [3]. In this analysis, it is possible to detect the behaviors that cannot normally be detected by static analysis; however, running of the malware in the device poses a security thread for the mobile device. Therefore, in this study, by using the static analysis approach, it is accessed to manifest files of Android applications present in M0Droid dataset which was previously analyzed and the relevant malwares the device are detected with permission data from these files. During the testing stage of system, different classification methods are used, and the permission-based attributes used are analyzed.

The remaining parts of the present paper are designed as follows: The relevant studies are discussed in Chapter 2. The methods related to pre-processing, feature extraction and classification are explained in Chapter 3. Information about datasets are included in Chapter 4. In Chapter 5, experimental results are provided with an assessment of the best classification method. In the final chapter, the results are summarized with referring suggestions.

II. RELATED WORKS

Detection of malicious application is achieved via attributes derived from the application file. In this context, the attributes are obtained by two different approaches in the literature, namely the static and dynamic analysis. In DroidRanger system using dynamic analysis approach, extraction of attributes is achieved by using the fingerprint-based detection engine [4]. These attributes include permission data extracted from the manifest file, and semantics in byte code. It also includes data obtained from heuristic based detection engine called root privileges which controls the running application. MAST, a mobile application security tool, based on static analyzing approach, helps to bring out the behaviors that are likely to be malicious using limited analyzing

resources. In this tool, Multiple Correspondence Analysis was used, which statically measure the relations among multi-categorical data during the feature extraction from application package. Code of practice and naturally presence of codes as well as permission data were used in developing this tool. The MAST tool was tested and compared by 5 different attribute groups and separate machine learning methods on 182 attributes [5]. In a method called DREBIN, the authors statically collected various attributes among Android applications, and they applied the support vector machines (SVM) on the resulting model. Along with permission data, they also used many attributes such as API calls, network addresses, hardware components, and gained a success rate of 93.9% with the SVM method [6]. Another study using the similar features is called DroidMat system, developed by Wu et al [7].

This system used permission data, API calls, constituents and instant messaging transitions as attributes in order to characterize the behavior of Android application by using static analyzing method. The modeling capacity of malware was attempted to be increased by applying k-means clustering method to these features. The number of clusters was determined by Singular Value Decomposition (SVD) method. Following the clustering process, 97% of success was achieved with k-NN classifier [7].

The permission data is of great importance in determining whether Android applications are malicious or not. In several studies, the permission data has been observed to significantly affect the success of classification as a common feature, even though different attributes were used. By only using permission data, Egemen et al. achieved a high performance in detection of malwares with a success rate of 98% using the Random Forest method [8]. In another study using permission data, a permission-based Android malware detection system, namely APK Auditor, was composed of three main constituents.

The system consists of a signature database for storing the attributes, a client for users requesting analyses, a database and a server for communicating to the server. By using this system, a total of 8762 applications were tested with a success rate of 88% [9].

III. METHOD

The file "AndroidManifest.xml" created by the developer of Android applications not only includes some application-related basic information such as the name of package and version, but also includes user permissions. Each application requests the related permissions in the manifest file from user

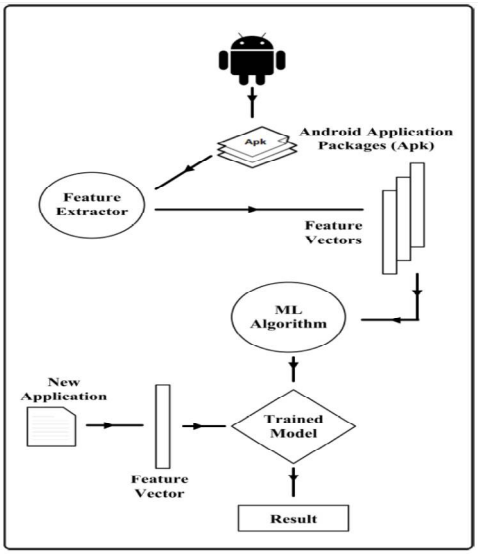

Fig.1. System Model

during loading. Serving as malicious software, these permissions are able to access the identity and the location of device rendering user privacy and security under thread. However, the majority of the users install the application to their devices without checking the permission information as they do not have sufficient knowledge about application permissions, and they are mostly unaware of these malicious applications. Thanks to the system model used in our study, such malicious softwares are automatically detected by static approach. Figure 1 illustrates the system model used in detecting malwares for Android operating system. During the first stage of the system, Apk files are granted to the attribute extractor in order to extract permission data. For each application, the permission data were obtained by Iteedee.ApkReader open code source library. This library lists the permission data required for each apk file. During the attribute extraction stage, all samples were browsed to gather unique permission data at first, and then they were acknowledged as attributes. In the encoding of attribute-value, the permission data requested from user by the application were assigned to the value 1, while others were assigned to value 0. Later, each sample in the data set was transformed into an attribute vector using the attributes obtained. The data set comprised of attribute vectors was delivered to the machine learning algorithm, and thus the system was trained. In the testing stage, each new sample was either classified as benign or malware with this trained model. The classification methods were conducted by Weka, a machine learning library with open source code.

Table1: # of sample for M0Droid and our data set

Data Set	# of sample		
	Bening	Malware	Total
M0Droid	200	200	400
Created Data Set	173	153	326

IV. DATASET

Various data sets used for malware detection are available in the literature. In this study, we used M0Droid data sets, developed by Damshenas et al.[10], which had previously been analyzed by a different method. As the attribute extractor could not extract the attributes from all samples present in this data set, however, experimental data set was obtained by convenience sampling. M0Droid data set includes a total of 400 samples, with each benign or malign category having equally 200 samples. On the other hand, the data set that we used in our experiments is composed of totally 326 samples, 173 of which are benign and 153 are maligns. A summary from the data set used in M0Droid and experiments is given in **Table 1**.

Change Wifi State	Internet	Bluetooth	Send Sms	Sample Category
0.0	0.0	0.0	0.0	Bening
1.0	1.0	1.0	0.0	Bening
1.0	1.0	0.0	0.0	Bening
0.0	1.0	0.0	0.0	Malware
0.0	1.0	0.0	1.0	Malware
0.0	1.0	0.0	1.0	Malware
0.0	0.0	0.0	1.0	Bening
1.0	1.0	1.0	1.0	Malware

Fig.2. A Small Part of Generated Dataset

Out of 195 attributes used in experimental data set, samples with only 4 attributes are partially shown in **Figure 2**.

V. EXPERIMENTAL RESULTS

In this section, testing is performed in order to detect any malware on the resulting data set. A total of 195 permission-based unique attributes are extracted from the data set. The main purpose of our study is to detect any malware using permission-based static approach on M0Droid data set. To separate training and testing, data, 10-fold cross validation method is used. The data set is divided into 10 pieces by cross validation method, and each piece was respectively used as a test set. By repeating of this process 10 times, classification results are obtained. Among the machine learning methods widely used during classification, k-Nearest Neighbor (k-NN), Naive Bayes (NB), Support Vector Machines (DVM), Backpropagation (BP) and Random Forest (RO) algorithms are used. In order to determine the optimum k value, the results for any k values between 1 and 5 are obtained in k-NN method. The default parameter values of experimental classifiers are given in **Table 2**. Accuracy metrics is used in the assessment of performance and the resulting classification success is shown in **Table 3**.

Table2: Default Parameters for Classification Techniques

Classifier	Value
k-NN	*k=1*
	Distance function= Euclidean
SVM	*c=1*
	Kernel=PolyKernel
	Tolerance=0.001
	ε=1.0E-12
BP	*# of Neuron in Hidden Layer=a ((attributes + classes) / 2)*
	Learning rate=0.3
	Momentum=0.2
RF	*# of tree=100*

Table3: The Success Rate of Classification Techniques (%)

Classifier / Success Rate					
NB	SVM	BP	RF	*k*-NN	
87.7	92.9	93.5	94.5	*k*=1	92.6
				k=2	92.9
				k=3	92.6
				k=4	93.2
				k=5	92.9

When examined the experimental results, permission data are found to be successful for M0Droid data set as well as detecting malicious applications. Among the classifiers, Random Forest method proved to be the most successful method with a accuracy rate of 94.5%. Analyzing the permission data which distinguish any mobile software from others is important for providing information to a user who will install the application. Therefore, it was determined which permission data were more often observed in benign and malware applications. The top 5 permission data (for each target) observed most often in benign and malware applications among data set were given in **Table 4**.

Table4: The Most Observed Five Features in Each Category*(Red ones illustrate that the number of most observed feature in target)*

Permission-Based Feature	Frequency	
	Bening	Malware
Internet	131	142
Access_Network_State	113	119
Read_Phone_State	75	132
Write_External_Storage	118	118
Vibrate	75	22
Send_Sms	9	101

When examined the values in Table 4, it is obvious that to which extent the observed frequency of attributes are efficient in distinguishing among samples.The most important point to detect malware is the frequency difference of features used in this target. It is much easier to distinguish between benign and malware applications, if there are many features, which has too much frequency difference, observed

in targets, In this context, if only malware samples has "Send_SMS" feature, it can be said that a sample containing this feature is malicious. The five features which has biggest frequency difference in benign and malware samples are given in **Table 5**.

Table5: The Frequencies of Five Features which has Biggest Difference Among Categories

Permission-Based Feature	Benign	Malware
Vibrate	75	22
Read_Phone_State	75	132
Instal_Packages	1	80
Delete_Packages	0	80
Send_Sms	9	101

According to Table 5, the five biggest difference valuesof featuresin benign and malware apps are determinedas the number of 53, 57, 79, 80 and 92 respectively.

CONCLUSION AND FUTURE WORK

In this study, the detection of malwares for Android platform using static approach is performed on M0Droid data set. In the experiments, permission-based data from the manifest files are used as attributes. The results are obtained using different classification methods, and the most common attributes between the categories are examined. In the study of creating M0Droid data set, a success rate of 69.5% is achieved using API system calls [10]. In our study, however, malicious softwares are detected using permission data from the same data set. When examined the experimental results, it is observed that success above 90% could be achieved by means of other classification methods except Naive Bayes. In addition, malicious softwares were automatically detected with the highest success rate of 94.5% in our study. In this context, the result increases the success rate of reference by 25%. When examined the attributes and taking the results into account, it can be said that the permission-based data are more efficient than attributes obtained by system calls. When examined the highest differential attribute frequencies in Table 5, the result is obviously expected. Applications in the data set can even be classified just for considering the attribute "*Delete Packages*", because the mentioned attribute has never been observed among benign samples. Therefore, the application including this attribute can be classified as malware. In all pattern and object recognition problems, the method used has a direct influence on the results throughout the stages of system. For this reason, any efforts to detect malicious software can be tackled as a whole pattern recognition problem. Besides, the effect of feature selection and term weighting methods on the results can also be studied in detecting mobile malwares. In future studies, the size of data set is intended to increase. It is also intended to propose a new attribute approach that can distinguish among samples.

REFERENCES

[1] http://grifikirler.com/turkiyede-akilli-telefon-istatistikleri.html,Access Date: 26/12/2015

[2] G Data Security Mobile Malware Report, Threat Report: Q1/2015, pp. 1-7, 2015

[3] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani,, "Crowdroid: Behavior-Based Malware Detection System for Android", SPSM '11 Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, pp. 15-26, 2011

[4] 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", 19th Network and Distrubted System Security Symposium (NDSS 2012), San Diego, CA, USA, 2012

[5] S. Chakradeo, B. Reaves, P. Traynor, and W. Enck, "MAST: Triage for Market-scale Mobil Malware Analysis", 6th WiSec, Budapest, Hungary, 2013.

[6] D. Arp, M. Spreitzenbarth, Hübner, H. Gascon,, and K. Rieck, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket", Network and Distributed System Secuirty Symposium, 2014

[7] D-J. Wu, C-H. Mao, T-E. Wei, H-M. Lee, K-P. Wu, "DroidMat: Android malware detection through manifest and API calls tracing", Information Security (Asia JCIS), Tokyo, pp. 62-69, 2012

[8] E. Egemen, E. İnal, and A. Levi, "Mobile malware classification based on permission data", Signal Processing and Communication Conference (SIU), pp. 1529-1532, Malatya

[9] T. Kabakus, A. Dogru, and A. Cetin, "APK Auditor: Permission-based Android malware detection system",Digital Investigation, Elsevier 2015

[10] M. Damshenas, A. Dehghantanha, and K-K. Raymond Choo, and R. Mahmud, "M0Droid: An Android Behavioral-Based Malware Detection Model", Journal of Information Privacy and Security, vol. 11, pp. 141-157, 2015

★★★