ANDROID MALWARE DETECTION USING MACHING LEARNING

Dr.K.P.Porkodi [1], G.Hemalatha [2], S.Manjula [3], K.Abarna [4], S.Deepika [5]

[1] Associate Professor, Department of Computer Science and Engineering.
[2.3.4.5] Under Graduate Students, Department of Computer Science and Engineering.
[12345]Vivekananda College of Technology for Women, Thiruchengode, Namakkal, Tamil Nadu, India.

Abstract:

Android mobiles are more popular than other software system like IoS. Android market share will reach 85.3%, more than 1 billion android device are vulnerable to malware. In android mobile are attack by malware to stealing the information, private message, hacking to manage the equipment. It causes series damage to consumer. Our proposed technique is used to identify the malware in android application by developing an algorithm namely Support Vector Machine (SVM), the system is light that is responsible for detecting the malware effectively.

Keywords: SVM, PCA-RELIEF, KNN, APK, ML

I. Introduction

Nowadays, malwares are growing rapidly, many organizations causing financially losses because of design to harm programming device, service and network. So the anti-malware companies have been actioning attacks from malware. Android has been always been much needed for free software and unpaid benefits in day to day lives. So malicious application and computer program with increase the degree of accuracy [1]. The method of signature based detection used both PC and Android devices from APK by extracting the signature with comparing a malicious signature on a virus website [17]; however our method is detecting anonymous malicious act database. Discovered the possible ways to analyze an unpredictable version of an Android computer: static analysis and dynamic analysis. In the advancement of machine learning technique or methods defined distinguish acquisitions are positive and negative collection of event and various signal features of malicious application and the system, immature collection are unimportant set causes a false conclusion, so the feature selection process is important[4]. Feature selection and detection are important process cause a accuracy level of separator. Then achieve easily the acquisition effect with classifier. Encouraged the observation, we suggest an Android platform acquisition system based the subdivision of machine learning called support vector machine [6]. This system focuses on feature selection and feature removal including static and dynamic analysis to detect features of smart phones. And a feature selection process called PCA-RELIEF to determine good feature clause. System shows the test results of android mobile to detect malware.

II. Literature Review

It is noteworthy that many magazines use multiple machines learning technique. We have analyzed in detail which technique researchers are working with or which technique is being discussed or developed an algorithm that has found the best result in another experiment as a basis for classification.

2.1 Logistic Regression:

Shukla and Tiwari [11] &Hanna [5] suggested a method to identify the malicious software in android apps by the permission of an API. The authors split the discovery of malicious programs into four stages: Review engineering, extract features, feature vector production and segmentation. They downloaded the APK file with the return engine tools and found the smali files and android.xml .Exposed to vectors of the combined feature. They gained 96.56% accuracy on integrated features using an alignment algorithm.

2.2 Naive Bayes and Bayesian Network

Tam k [9] proposes a Bayesian classification method to face a malicious android application. Three tools are developed by authors: call researchers, command and authorization detectors. They extract the functionalities in the API telephone, the resources, the assets, the libraries and the

authorizations respectively by analytical analysis of data, the principal attributes of discriminatory attributes are selected to create the functionalities. Finally, the Bayesian category is designed to make decisions of their test data base contains 1000 malicious software samples and 1000 malicious applications. Under the conditions uses of 20 attributes as elements of the division, the performances indicate the accuracy, the precision and the AUC.

2.3 K-Nearest Neighbor

Droid Mat is the system developed by Wu [7]. That consistently details, permissions, component sending ,Objective messages, and API calls, display behavior in android app. Droid Mat extracts information from Manifest file once and views the components as drop-down entries to track API calls. Next, the k-means algorithm is used to develop non-computer program modeling skills. Collection number is determined by the single value deviation (SVD) method at the lowest value [18]. Finally it classifies an application as clean or malicious by using the KNN algorithm. The model reached 97.87% perfection, 87.39% memory, 96.74% perfection, and 91.83% F1-score on a website from the "mobile Congtagio" site. Lee&Mao [7] investigated the operation of the simple machine learning separators. The author has made extended comparisons using well-known distance measurements over the Drebin website. The results show that the range of the appropriate selection range can provide significant improvements in class accuracy. Specifically, Hamming and City Block can improve performance on malware detection. For example, when compared to the Euclidean range, City Block could promote KNN's false algorithm up to 33%.

2.4 Random Forest and Decision Tree

A method developed by Coronado [3] & shung [11] removes a few features from the manifest file to create machine-readable separators. These features are set by the required application permissions and subgroups. Produce a vector of input for all possible Permits, and use Naïve Bayes, J48, Random Forest and other dividers to conduct experiments, as wellget very good results in a random forest consist of 100 trees with 98% AUC and 94.83% accuracy. Canfora employed the probability n consecutive opcodes in the code section features such as 2 opcodes sequences [(move, Ask), (ask, add) …], and individual brackets are vector part [19]. If (submit, request) chances, (ask, add) 0.003 chances. Then, the value of this vector element is [0.001, 0.003 …].The authors trained dividers into two categories, SVM and Random Forest, to make binary divisions .The outcome show 97%accuracy can be achieved on average. If 2 codes are used Kaang did the same job. They also used n-opcodes to test features for Naïve Bayes, SVM, Part Decision Tree, and Algorithms for Random Forest Classification. In N3, N4 SVM shows a high F1 score of 98%, once The Random Forest shows excellent performance in terms of both speed training and prediction.

2.5 Convolutional Neural Network

Zhang et al [2] &Yang [8] proposed Deep Classify Droid, which takes a three-stage approach as follows: feature Extraction and feature integration then detection, to evaluate malware program on android apps based on Convolution Neural Network. From the codes to the combined-vector joint Space then, they coached the CNN model by using of two complexity layers, a pooling layers and a fully connection layer to learn these vectors. Experiments shows that the approach achieves an accuracy of 97.4% with few false approach achieve the dataset of 5546 malware and 5224 good apps.

III.ANDROIDAPP FEATURES

3.1 Android App's Static Features

Zhu [10] & Kira [16] suggested that the loosely coupled components were bound together by manifest file of Android Apps. It describes manifest files of each external libraries, components, application metadata, platform requirements, external libraries, required permits, etc. Work, service, content provider and broadcast receiver are basic components that make up the android app respectively perform different tasks. An APK of android application is downloaded via APK package, zip archive. Mostly composed of benefits, lib, res, apparent, Dalvik Bytecode these are the files were used as source material. There are types of following features

.

3.1.1 Permission Features:

From AndroidManifest.xml uses various permission in the app during runtime. Extracting the permission by checking malicious app. The permission of the android system is about 250 kind of permission in the feature of binary vector taking 250 bits.

3.1.2 Component Features:

It needs four basic compounds for registered AndroidManifest.xml .They will be initialized and created in relation to the system calls in classes.dex file. Fored feature vector is included the types and the quantities of component.

3.1.3 Intent Features:

A.Desnos [15] show that class.dex and AndroidManifest.xml folders are used the pass messages in between components**.** When intent passed to component call back function is predefined to execute the process and often intents are used with components to analyze association through the two component.

3.1.4 Constant String Features:

Munoz A [12] proved that the small codes stored by dex file and resources, strings.xml files are stored in developer defined strings. Extracting frequency, content of string from files can reflect app characteristics. Consider type of strings

and few of them are long, it carries a hash operation before forming vector processing.

3.1.5 Opcode and API Features:

Peiravian implemented and proved that [13] the dex files the frequency of Dalvik opcode and calls from API shows developer programming habits that are best suited to generate acquisition features. The number of opcodes and APIs reveals significant differences between malicious and malicious applications. It can be generated from feature vectors by measuring N opcode frequency and -APIs.

3.1.6 Native Code Frequency Features:

Sahib [14] experienced that .so files use an instruction to performing malware operation, by complied codes and more difficult for decompile, it brings more obstacles to the detection process. Extracting the frequencies of the system call and the arm opcode from the .so file can greatly to help the detection function.

IV.METHODOLOGY

Our system can detect malicious software directly through android application. The limitated resources on smart phones, in this system we proposed a client server model.

4.1 SYSTEM DESCRIPTION

There are two categories namely, client -server model. In client side, user interface (UI) alert you to outcome of system predicts. We collected the number of MD5 values from the limitated resources we put a simple check on customers when the new system is installed it releases MD5 value and it makes a comparison with malicious sql lite MD5 .This system will extract malware information and remainder the user to delete it, if new MD5 values available but new MD5 values is not in sql lite the application will be send to the server. In the server, extracted feature from static and dynamic analysis from output module. We release usage, permission, purpose, feature, application, API , then choose CPU consumption, battery consumption, number of short message and the number of running processes as the dynamic features. The raw feature will be sent to the feature selection module and select few key features then reducing the redundant, features based on PCA-RELIEF. Finally, we build a classification model by using SVM and evaluate the unknown android application by classifying it into malware or benign.

4.2 FEATURE EXTRACTION

4.2.1 Static Analysis Module

Static analysis is used to extract features. When analysis the static module, we implement a decoder based on Androguard tool, it is one of the largest open source projects for android static analysis. After decompiling, we collect various features from AndroidManifest.xml, classes.Dex. We choose permission, API, uses-feature, application, intent as the static features. Research finds that permission system in android is one of the most important security mechanisms; malicious software tends to request sensitive permissions more than benign software, such as android. Permission, SEND_SMS, etc. Similarly, uses-feature defines the access to the hardware, as is shown in Table 1, this malicious application applies access to the touch screen and the camera, requesting access to specific hardware often reflect harmful behavior. Application consists off our different types of component sin an application, the names of these components may help identify the famous component so malware. Intent can be used to trigger malicious activities, to collect the datas listed in manifest and find attributes.

Feature Name	Ranked Weight
Android. Permission. Read _Sms	0.915
Android. Intent. Action. Boot_ Completed	0.905
Android. Permission. Sent_ Sms	0.873
Android/Telephony/Telephony manager; Getdeviceid	0.840
Android. Permission. Read_ Phone_ State	0.792
Android/Telephony/Telephony manager; Getsubscriberid	0.532
Android. Permission. Call_ Phone	0.521
Android. Permission. System_ Alert_ Window	0.321
Android. Permission. Access_ Wifi_ state	0.242
Android. Net. Wifi. Pick_ Wifi_ Work	0.232

Table 1. The highest 10 attribute by PCA-RELIEF

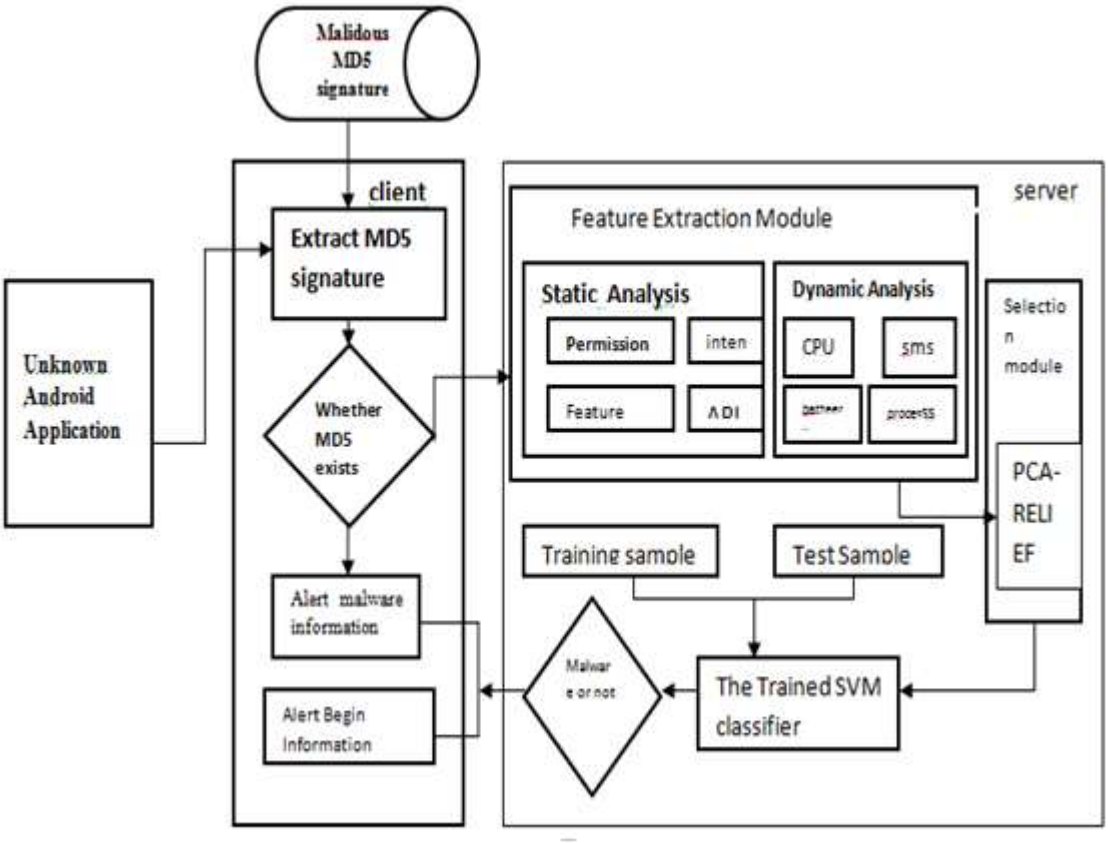

Figure 1. Architecture of Android Malware Detection

4.2.2 Dynamic Analysis Module

We implement a dynamic analysis module by using Droid Box dynamic analysis will incur a large number of computational overhead and consume a lot of time, and thus our system will mainly use the static analysis. The dynamic analysis module is motivated when the static analysis failing to decompile the APK because some of the applications are obfuscated and encrypted. First, the dynamic module will launch the android virtual machine to load the APK, and then invoke the Droid Box tool to monitor the application behavior and the system state, called dynamic analysis module. After that, information collection module will gather the dynamic features

4.3 Feature Selections
Description of Relief and PCA

Relief is one of the filter types in feature selection algorithm based on the relevance features and classification which is known to solve the problem of two classifications by giving the weight score. However, relief has a shortcoming for its disability in eliminating redundant features; with the high relevant features the correlations is a dimension reduction algorithm which helps transform original features linearly into a low-dimensional subspace to reduce the dimensionality of the raw data set. PCA is usually used to eliminate redundant features.

Order to remedy the defect of relief, a new feature selection PCA-RELIEF is a proposed to find the most discriminating android feature subset

4.4 Experiment and Discussion

We collect 2000 Android applications including 1000 benign applications and 1000 malware, the benign samples are collected in the Google Play by the crawler technology algorithm is build by SVM classifier, 20% of samples and the test data set, 80% of samples in trained As shown in the Table 1, READ_SMS can be defined as the most signal feature indistinguishing the malware good, some original features are removed because of its low rank, such as the internet permission. By using PCA-RELIEF we gathered the highest 10 attributes of each feature for building the classification model. True Positive Rate (TPR), False Positive Rate (FPR) and accuracy. TPR is the rate of correctly detection a sample, however FPR is defined as the false detection of good

application as malware. Accuracy shows the precise of the classifier in classifying the samples in the right class. SVM, a linear classifier, determines a hyper plane that separates both classes with maximal margin; we consider it for our system.

V. CONCLUSION

In this paper, we implement an Android malware detection system based on SVM, different from the traditional detection method, it can detect the unknown Android application based on the machine learning. We extract various features with the method of static analysis and dynamic analysis. A new feature selection algorithm PCA-RELIEF is also proposed to dispose the raw features and our experimental result shows that the new method performs better with higher detection rate and lower error detection rate compared with the traditional detection approaches such as the detection method based on signature.

VI. REFERENCES

1. Long Wen and Haiyang Yu .the Android data statistics in the second quarter of 2016[EB/OL].http://www.baijingapp.com/article/7842.
2. X. Li, J. Liu, Y. Huo, R. Zhang and Y. Yao, "An Android malware detection method based on Android Manifest file" 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, 2016, pp. 239-243.R. T. Wang, "Title of Chapter," in Classic Physiques, edited by R.B. Hamil (Publisher Name, Publisher City, 1999), pp. 212–213.
3. L. D. Coronado-De-Alba, A. Rodríguez-Mota and P. J. E. Ambrosio, "Feature selection and ensemble of classifiers for Android malware detection," 2016 8th IEEE Latin-American Conference on Communications (LATINCOM), Medellin, 2016, pp. 1-6.
4. Qin Z, Xu Y, Liang B, et al. An Android malware static detection method [J]. Journal of Southeast University, 2013, 43(6):1162-1167.

5.Felt A P, Chin E, Hanna S, et al. Android permissions demystified[C]// ACM Conference on Computer and Communications Security. ACM, 2011:627-638.

5. Yang Z, Yang M. LeakMiner: Detect Information Leakage on Android with Static Taint Analysis[C]// Software Engineering. IEEE, 2012:101-104.
6. Wu D J, Mao C H, Lee H M, et al. DroidMat: Android Malware Detection through Manifest and API Calls Tracing[C]// Information Security. 2012:62-69.
7. Qiao Y, Yang Y, He J, et al. CBM: Free, Automatic Malware Analysis Framework Using API Call Sequences [M]// Knowledge Engineering and Management. Springer Berlin Heidelberg, 2014:225-236.
8. Tam K, Khan S J, Fattori A, et al. CopperDroid: Automatic Reconstruction of Android Malware Behaviors[C]// Network and Distributed System Security Symposium. 2015.
9. Zheng C, Zhu S, Dai S, et al. SmartDroid: an automatic system for revealing UI-based trigger conditions in android applications [J]. 2012
10. Qiao M, Sung A H, Liu Q. Merging Permission and API Features for Android Malware

Detection[C]// Iiai International Congress on Advanced Applied Informatics. 2016:566-571.

11. Munoz A, Martin I, Guzman A, et al. Android malware detection from Google Play meta-data: Selection of important features[C]// Communications and Network Security. IEEE, 2015:701-702.
12. Peiravian N, Zhu X. Machine Learning for Android Malware Detection Using Permission and API Calls[C]// IEEE, International Conference on TOOLS with Artificial Intelligence. IEEE, 2013:300-305.
13. Mas'Ud M Z, Sahib S, Abdollah M F, et al. Analysis of Features Selection and Machine Learning Classifier in Android Malware Detection[C]// International Conference on Information Science & Applications. IEEE, 2014:1-5.
14. A. Desnos and G. Gueguen, et al.https://github.com/androguard/androguard, visited August 2015.

15.Droidboxproject.https://github.com/pjlantz/droidbox.Kira K, Rendell L A. The feature selection problem: traditional methods and a new algorithm[C]// National Conference on Artificial Intelligence. San Jose, Ca, July. DBLP, 1992:129-134.

16.Computational engine performance and emission analysis using Ceiba Pentandra biodiesel (Elservier).Panneerselvam,NMurugesan,A,Porkodi,K P,Jima,Terefe,Vijayakumar,CSubramanian,Biofuels,Volume.7,Issue.3,2015,PP.201-206.

17. Efficient Classification of Heart Disease using machine learning Algorthim(Scopus).,Dr K.P.Porkodi., Journal of Xi'an Shiyou University,Natural scienceEdition.,Volume 17,Issue 07,PP-120-122.

18. A Survey of Underwater Wireless Sensor Networks and its Challenges.,K.P.PORKODI.,and A.M.J.MD.ZUBAIRRAHMAN.,Asian Journal of Research in Social Sciences Vol.6 (2016).,pp.594-604.

.