ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 4, Special Issue 11, March 2017

Malware Detection through ICC based Detector Using Linear SVM Algorithm

Karthikeyan M, Akila T and Vinitha Sree L

Assistant Professor[1], Assistant Professor[2], PG Scholar[3]

Department of Information Technology[12] , Computer Science and Engineering[3],

Mahendra College of Engineering, Salem.

ABSTRACT: Many existing malware detection methods are designed to detect malware based on required resources such as permissions, system calls, and suspicious API calls. Now, a tag of Android applications is used to inform users about the risks of installing applications. World Wide users have begun to develop an increasingly large number of mobile apps in response to advancements in Android technologies and wireless networks. Increased number of applications results in a greater chance of installing malicious applications and malware. When a user installs an application, the user has the ability to check the Android application's permission requirements and cancel the installation if the permissions are unwanted or excessive.

As a result, the majority of existing methods are less effective in identifying many typical malware that require little or no suspicious resources, but leverage on inter-component communication (ICC) mechanism when rigid attacks start. In existing system ICC Detector gives a recognition model after training with a set of benign applications and a set of malware, and employs the trained model for malware detection. In our proposed method, a framework that can detect android malware applications is suggested to help the organization of Android Market. The proposed framework intends to develop a machine-based malware detection system on Android to detect malware applications and improve the security and privacy of smartphone users. This system monitors various authorization-based features and events received by the Android applications and analyzes these characteristics using machine learning classifiers to classify whether the application is good or malware.

1. INTRODUCTION:

In recent mobile phones have become popular in our lives since they offer almost the same functionality as personal computers. Recently in mobile technology, Android application based mobile devices gain more popularity, and they were now an ideal target for attackers. Android-based smartphone users can download free apps from Android Application Market. These applications have not been certified by authorized organizations and may contain malware applications that can use and steal the privacy information for users. The increasing number of security threats targeting mobile devices has arisen. In fact, malicious users and hackers use both the limited capabilities of mobile devices and the lack of standard security mechanisms to design mobile-specific malware that can access sensitive data, steal the user's phone credit, or deny access to some device functionality.

In recent years, mobile devices such as smartphones, tablets, and PDAs have become popular by increasing the number and complexity of their capabilities. Current mobile devices offer a large amount of services and applications than those offered by personal computers. At the same time, the increasing number of security threats directed to mobile devices has arisen. In fact, malicious users and hackers use both the limited capabilities of mobile devices and the lack of standard security mechanisms to design mobile-specific malware that access sensitive data, steal the user's phone credit, or deny access to some device functionality. In 2011, malware attacks rose by 155 percent over all platforms : In particular, Android is the platform with the highest malware growth rate by the end of 2011. To alleviate these security threats, various mobile specific intrusion detection systems (IDSes) have been recently Suggested. Most of these IDs are behavior-based, so you do not rely on a database of malicious code patterns, as is the case with signature-based IDSs. In this work, we describe a machine-based malware detection system for Android-based smartphone users.

Manual Detection Model (MDM) which is different from the traditional signature scanning methods. Firstly, we monitor the path and files, and touch the scan option. After decoding MDM model, abnormal process can be detected using the matching extension files with empty spaces, the experimental results demonstrate that the proposed method can effectively detect mobile malwares. In recent years, smart phone technology is becoming increasingly popular. The dangers of mobile phone malwares are becoming more and more serious.

Permissions are one of the keys to security on Android. Earlier work evaluates the detection of malware with permissions. They all recognize that a permission-based mechanism can be used as a fast filter to identify malicious applications and that it must be linked to a second element (such as dynamic analysis) to make complete analysis a reported malicious application. This result is justified by interesting performance indicators such as true positive, false positive, correct negative and false negative. For the authorization analysis work, the solutions provided help the user select the permissions, while either extending the installation system (without the user's involvement) or an interface with necessary information about authorizations (with the participation of the use). The process of extracting features from the Android .apk files. We create a record from extracted features of Android applications to develop android malware detection framework. We carry out an empirical validation of machine learning methods and show that they can achieve high accuracy. [9] proposed a system in which an automatic anatomy segmentation method is proposed which effectively combines the Active Appearance Model, Live Wire and Graph Cut (ALG) ideas to exploit their complementary strengths. It consists of three main parts: model building, initialization, and delineation. For the initialization (recognition) part, a pseudo strategy is employed and the organs are segmented slice by slice via the OAAM (Oriented Active Appearance method). The purpose of initialization is to provide rough object localization and shape constraints for a latter GC method, which will produce refined delineation. It is better to have a fast and robust method than a slow and more accurate technique for initialization.

2. EXISTING SYSTEM:

ICC Detector is an effective and accurate malware detection method that detects malwares based on not their required resources, but their ICC patterns. The ICC patterns of an app represent how they use the ICC mechanism and can be extracted from the app's APK file. The ICC detector is trained with the ICC patterns extracted from some benign applications and from certain malware before issuing a detection model. The detection model is used to detect malware based on its ICC patterns. Looking at the ICC patterns, ICC Detector not only examines the communication between applications and Android system, but also interactions between applications. For this reason, the ICC detector is particularly useful for detecting those "advanced malware" that invalidate most existing malware detection methods by exploiting the ICC mechanism rather than requiring suspicious resources. ICC Detector is evaluated with 5,264 malwares and 12,026 benign apps. We argue that existing malware defenses, without considering the special characteristics of Smartphone malware and that of Smartphone’s themselves, might not be sufficient to detect sophisticated malware. First, mobile malware is continuing advancing and becomes more sophisticated and stealthy. To avoid crafting detection patterns manually, we make use machine learning for generating detection models. While learning techniques provide a powerful tool for automatically inferring models, they require a representative basis of data for training.

Disadvantages:

- Designed to detect malwares based on required resources.
- Detected applications as standalone entities in Android platforms.
- Android allows for installing applications from unverified sources, such as third-party markets, which makes bundling and distributing applications with malware easy for attackers.

3. PROPOSED SYSTEM:

In our proposed method, a framework that can detect android malware applications is suggested to help the organization of Android Market. The proposed framework intends to develop a machine-based malware detection system on Android to detect malware applications and improve the security and privacy of smartphone users. This system monitors various authorization-based features and actions that are obtained from the Android applications and analyze these functions using machine learning classifiers to classify whether the application is good or malware. To eliminate all of the aforementioned problems, a new model is proposed based on the feature selection as the first phase, the K-mean cluster model generation as the second phase, the classification of this new data set generated by the second phase as the third phase, and finally Assess the performance of this proposed model in terms of accuracy, precision and recall.

We propose TWR, a novel approach for malware prevention with an exclusive focus on the smartphone platform based on intuitive gesture recognition. As part of this system, we propose two novel light-weight gesture recognition schemes that can be used in different contexts with little to no additional user involvement. It implicit phone tapping detection based on phone acceleration data, is geared for NFC applications, which usually require the user to tap her phone with another device. Our results suggest the proposed approach could be very effective for malware prevention, while imposing little to no additional burden on the users. The false negatives are expected to further reduce significantly as users become more familiar with the underlying gestures, especially since they are quite intuitive.

Our detection algorithm consists of two phases: training phase and recognition phase. In the training phase, a user performs the target action (tapping) multiple times, and accelerometer data of the action is recorded and processed to generate a tapping template. when the user requests access to a particular resource, the sensor is turned on and the gesture detection algorithm is executed. Once the kernel captures the required gesture, the permission will be granted to the active application.If kernel fails to capture the gesture within certain duration, application will be denied to use the resource. Android malware is a new yet fast growing threat.Classic defenses, such as anti-virus scanners, increasingly fail to cope with the amount and diversity of malware in application markets. The previous evaluation demonstrates the efficacy of our method in detecting recent malware on the Android platform. Another limitation which follows from t use of machine learning is possibility of mimicry and poisoning attacks.In addition, the false positives can also be carefully avoided in most cases, for example, by detecting the orientation of the device.

Our future effort will be focused on realizing this approach in practice and further evaluate it with a wide range of smartphones and smartphone users. Linear SMV algorithm. The linear SVM shows high performance among machine learning algorithms in order to effectively detect malware in the Android platform with monitored resources during application runtime.

SVM is one of the machine learning classifiers receiving the most attention currently, and its various applications are being introduced because of its high performance. The SVM could also solve the problem of classifying nonlinear data.It shows higher performance.he SVM technique could accurately detect the malware function.

Advantages

- Designed to reduce the developers' burden and promote functionality reuse.
- We create a dataset from extracted features of Android applications in order to develop android malware detection framework.
- We perform an empirical validation of machine learning approaches and show that they can achieve high accuracy rates.

4. MODULES

- Features
- Feature Extraction
- Feature Selection
- Machine Learning and Malware Detection

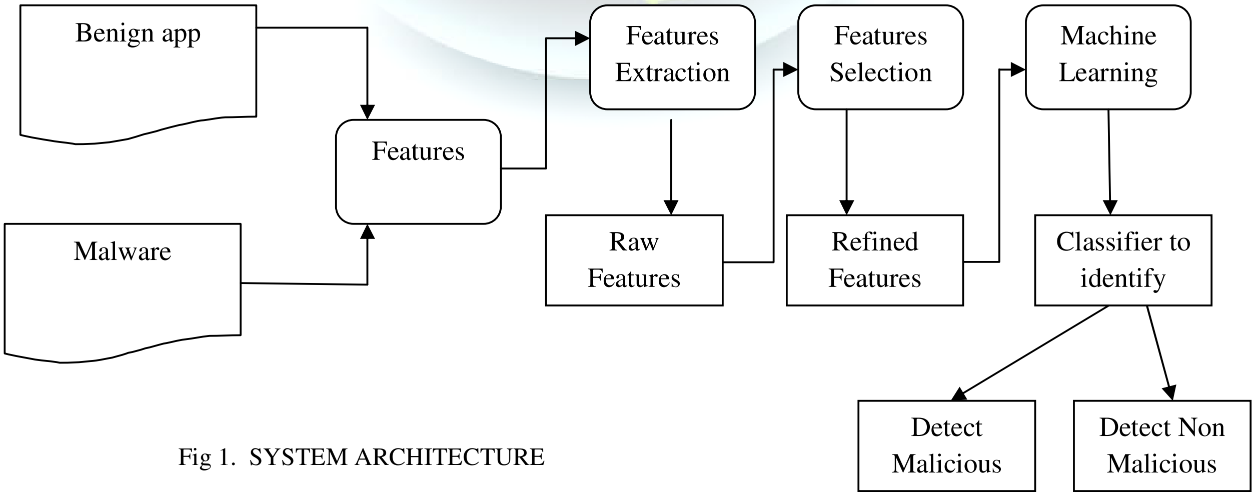

Fig 1. SYSTEM ARCHITECTURE

4.1 Features

For each Android application, we have retrieved several selected features from the corresponding application package (APK). In addition, we identified the actual permissions required by the application, and approved the features for malware detection. The values of selected features are stored as a binary number (0 or 1), which is displayed as a sequence of comma separated values. We list all selected functions in the following points. Each element contains the name of a function, the data type of the function and data of the function. The few examples are described below:

- **android.permission.WRITE_SMS-** can write the application SMS message without user response by allowing this permission request.
- **android.permission.SEND_SMS-** the application can send SMS message, so the money can be lost by installing similar applications with this permission request.
- **android.permission.CALL_PHONE-** some applications can request CALL_PHONE permission without needing them. If the user permits this permission request, the application calls the phone itself without user notification. Users do not know why their money was lost without their use.

4.2 Feature Extraction

We describe the process we followed to get data from the Android application file. The general steps we have followed for each application are:

1. We downloaded and collected malware and good ware applications from appliance market.
2. We decompress applications to extract the content.
3. We extract the authorization request functions from each application.
4. We create a record in an ARFF [4] file format with the extracted data.

First, we uncompress the Android application package file to extract the content. In the first three steps, the information from this source is retrieved. We process the "AndroidManifest.xml" file to extract this data.

4.3 Feature Selection

In machine learning applications, a large number of extracted features, some of which are redundant or irrelevant, present some problems, such as misleading the learning algorithm, overshooting, reducing the generality, and increasing model complexity and runtime. These undesirable effects are even more important in the application of machine learning methods on mobile terminals since these are often restricted by processing and storage capacities as well as battery power. Applying a fine feature selection in a preparation phase makes it possible to use our malware detector more efficiently, with a faster detection cycle. Nevertheless, the reduction in the amount of features should be carried out while maintaining a high degree of accuracy. In this section, we select the best features from the extracted features of Android application packages by using the feature selection method: Information Gain.

4.4 Machine Learning and Malware Detection

The selected functions are collected into the signature database and subdivided into training data and test data, and used by the usual machine learning techniques to detect the Android malware applications. We choose K-clustering (i) data-driven methods that have relatively few assumptions about the distribution of the underlying data, and (ii) guarantees at least one local minimum of the criterion function, thereby accelerating the convergence of clusters to large amounts of data. First stage: Clustering is performed at training instances to obtain k disjoint clusters. Each cluster represents a region of similar instances in the form of Euclidean distances between the instances and their cluster centroids. Second level: The K-mean method is cascaded with the decision tree learning by using the instances in each K-center cluster.

5. CONCLUSION

In this paper, we implement a framework for classify Android applications using machine learning techniques, be it malware or normal applications. To generate the models, we have extracted several authorization functions from several downloaded applications from the Android markets. Some of the malware applications are used in malware sample databases, and both malware and normal applications are classified using machine learning techniques. To validate our methods, we collected 200 samples of Android applications, and we extracted the above features for each application, and we trained the models that were evaluated with the Area Under ROC Curve (AUC).

REFERENCES:

[1] A. Liaw and M. Wiener, “Classification and regression by randomforest,” R news, vol. 2, no. 3, pp. 18–22, 2002.

[2] M. Grace, Y. Zhou, Q. Zhang, S. Zou, and X. Jiang, “Riskranker: scalable and accurate zero-day Android malware detection,” in Proceedings of the 10th international conference on Mobile Systems, Applications, and Services. ACM, 2012, pp. 281–294.

[3] Appchina, http://www.appchina.com/.

[4] F. Wei, S. Roy, X. Ou, and Robby, “Amandroid: A precise and general inter-component data flow analysis framework for security vetting of Android apps,” in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014, pp. 1329–1341.

[5] R. Schlegel, K. Zhang, X.-y. Zhou, M. Intwala, A. Kapadia, and X. Wang, “Soundcomber: A stealthy and context-aware sound Trojan for smartphones.” in NDSS, vol. 11, 2011, pp. 17–33.

[6] D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. Siemens, “Drebin: Effective and explainable detection of Android malware in your pocket,” in Proc. of NDSS, 2014.

[7] M. A. Hall, “Correlation-based feature selection for machine learning,” Ph.D. dissertation, The University of Waikato, 1999.

[8] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” 1990. D. Sounthiraraj, J. Sahs, G. Greenwood, Z. Lin, and L. Khan. Smvhunter: Large scale, automated detection of ssl/tls man-in-the-middle vulnerabilities in android apps. In *Proceedings of the 2014 Annual Network & Distributed System Security Symposium (NDSS)*, 2014.

[9] Christo Ananth, G.Gayathri, I.Uma Sankari, A.Vidhya, P.Karthiga, “Automatic Image Segmentation method based on ALG”, International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 2, Issue 4, April 2014,pp-3716-3721

[10] VirusTotal, https://www.virustotal.com/.

[11] C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras, “Droidminer: Automated mining and characterization of fine-grained malicious behaviors in Android applications,” in Computer Security-ESORICS 2014.Springer, 2014, pp. 163–182.

[12] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin, “Permission re-delegation: Attacks and defenses.” in USENIX Security Symposium, 2011.