IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 01-03

Android Application Malware Detection Using Machine Learning.

Gavhane Sachin[1*] Hariyani Vinay[2] Bagora Sanjay[3] Bari Charul[4]
[1](Dept. of Information Technology, Atharva College of Engg, Mumbai University, India)
[2](Dept. of Information Technology, Atharva College of Engg, Mumbai University, India)
[3](Dept. of Information Technology, Atharva College of Engg, Mumbai University, India)
[4](Dept. of Information Technology, Atharva College of Engg, Mumbai University, India)

Abstract: *Mobile devices have seen an exponential increase in the demand and its usage in recent years and this has made information access easy as well as vulnerable. Sensitive and critical information can be accessed by malicious applications using voluntary permission controls. It is necessary to develop an efficient and adaptable solution as signature-based antivirus solutions are ineffective due to high false detection rates. Android targeted malware has increased dramatically in recent years, and the focus of malware attackers towards the Android system has been much greater than other mobile operating systems. To address the problem of malware detection, we have proposed a machine learning-based malware detection system for Android platform. The system utilizes the features of collected random samples of benign and malware apps to train the classifiers*

I. Introduction

The major advantage of the Android mobile operating system is its open source code, which allows developers to build varied novel and unique systems. Android enables program developers to put their apps on the Google Play for users to download. Besides Google Play, there are quite a few unofficial program markets, such as Apkpure. Neither the official or unofficial markets have established an effective method of preventing the spread of malware; thus, smart phone security is under significant threat. Numerous free apps make mobile phones more likely to be attacked. Unlike personal computers, the defense mechanism of smart phones is limited by hardware constraints, and it is thus easier to access a user's personal data, possibly even leading to financial loss as a result. Nearly 1.5 billion unit of smartphones had been sold to end users in 2017 and Google's Android extends its leads in smartphone OS market by occupying approx. 82% of total market in 2017.

The swift adoption and modification of Android's OS, applications, and real world implementation have in many cases resulted in the widely used application with little or no malware protection. The notoriety of Android and execution imperfections because of fast change has not gone unnoticed by malware creators. Avast [6] reported that cyber-attacks against android operating system are increasing by 40% year-over-year since 2016. There is an earnest requirement for powerful and precise malware recognition framework to stop spread of malware in Android platform.

Section I contains the introduction of malware in android devices, Section II contain the related work of existing systems to detect malware in Android devices, Section III explain the proposedandroid malware detection system methodology with flow chart and Section IV concludes research work with future directions

II. Related Work

Malware detection in mobile devices is one of the hot topics in cyber security. Notable work has been done on the issue of malware detection in android mobile devices. A few methodologies screen the power utilization of applications, and report strange consumption. In literature, researchers extensively used machine learning techniques to model Android malwares patterns based on their static features and dynamic behavior to avoid difficulty of manually craft and update detection pattern for android malware and successfully discriminate Android malware samples form benign application. The techniques used to recognize malware are static and dynamic analysis. Dynamic analysis-based malware detection solutions experience the negative effects of manual overhead and require overwhelming instrumentation. Utilization of dynamic analysis to effectively identify malware though behavioral monitoring and traffic analysis of application at runtime is shown in DroidRanger [1] and CopperDroid [2] but this requires heavy instrumentation.

Drebin [3], Li-Dai [4] and DroidMat [5] used static analysis method to detect malware but suffer from manual craft and update problem and only used permission and API calls. Moreover, most of the mentioned approaches does not consider presence of key features like dynamic code, reflection code, native code, database and cryptographic code as potential features. Malware writers do uses dynamic and reflection code to make

application statically undetectable also uses crypto code for code obfuscation. Native code allow developer to access some of processor features and run directly on operating system hence making static and dynamic analysis approaches for mobile apps unusable. These techniques usually inspect only data flow in Dalvik bytecode and miss the data in native code components, which are becoming increasingly predominant. The proposed system uses Support Vector Machine (SVM) to perform malware classification. It makes use of comprehensive static analysis approach of application. The system uses permissions, API calls along with presence of key app's information which were not considered in most of previous proposed approaches, such as: crypto code, dynamic code, native code, reflection code, and database as a features set to generate binary vector from Android application samples of identified malware and goodware applications and adopt machine learning to perform malware classification.

III. Proposed System

In this paper we have proposed, a malware detection system for Android that extracts appropriate features that can be used in the machine learning classification algorithms to classify the benign and malicious Android applications. This system takes an Android application as input, examines it and returns two possible outputs: Malware or Goodware. This system consists of two parts: Training and Classification. Training consists of extracting features from each .apk file in training dataset and creating binary vector to train training classifier model using supervised machine learning algorithm. Classification consists of same steps as Training except the binary vector of Android app is passed through classifier module to classify the app as malware or goodware.

The system consists of following steps:

1. *Reverse Engineering:* In this step, the .apk files of Android applications are decompiled into their source code in the forms of AndroidManifest.xml and java classes by using Androguard [7]. Androguard [7] is a static analysis tool for third-party Android applications which disassembles apps and accesses their components using its API.
2. *Feature Extraction*: In this step, the source code received from the previous step are further examined and required features are parsed from the source code using python module, and stored in MongoDB database. The extracted features include requested permissions, API calls, along with presence of key information.
3. *Binary Vector Generation*: In this step, features extracted from each app are transformed into binary vector which is usable for machine learning algorithms. Classifier Modeling In this step, supervised machine learning algorithms are trained using binary vectors of the apps with label information to create classification model. Label information can be represented as goodware or malware. The algorithms used are: SVM, Decision Tree and Naive Bayes.
4. *Prediction:* In this step, same set of features are extracted from apps stored for classification to create binary vector. The binary vector is then passed through classifier module which uses classification model to classify apps into "goodware" or "malware".

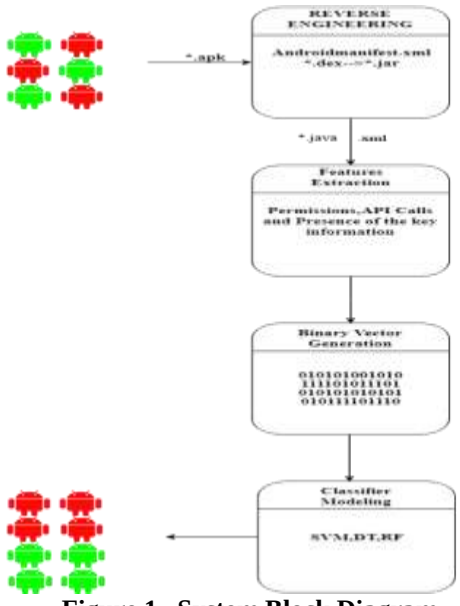

Figure 1. System Block Diagram

IV. Conclusion

The proposed Android malware detection system uses permission, APIs, and presence of others key apps information such as, dynamic code, reflection code, native code, cryptographic code, database etc. as features to train and build classification model by using various machine learning techniques which can automatically distinguish malicious Android apps (malware) from legitimate ones. In the presented system we missed out many features which can be useful for deciding behavior of any given application as malicious or benign. Broadcast receivers, Filtered Intend, deep native code analysis, and dynamic analysis are main topic of concern for future work which will help us to achieve better accuracy. Another area which require focus in future is in-depth understating of machine learning algorithms and feature engineering so that an efficient classification model can be built.

Acknowledgements

This manuscript would not have been possible without our respected principal Prof. Dr.Shrikant Kallurkar and the management of Atharva College of Engineering for providing such an ideal atmosphere to build up this project with well equipped library with all the utmost necessary reference materials and up to date IT Laboratories.We are extremely thankful to all staff and the management of the Atharva College of Engineering for providing us all the facilities and resources required.

References

[1]. Zhou, Y., Wang, Z., Zhou, W., & Jiang, X. (2012, February). Hey, you, get off of my market: detecting malicious apps in official and alternative android markets. In NDSS (Vol. 25, No. 4, pp. 50-52).
[2]. Reina, A., Fattori, A., & Cavallaro, L. (2013). A system call-centric analysis and stimulation technique to automatically reconstruct android malware behaviors.EuroSec, April.
[3]. Arp, Daniel, et al. "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket." NDSS. 2014.
[4]. Li, Wenjia, JigangGe, and Guqian Dai. "Detecting malware for android platform: Ansvm-based approach." Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on.IEEE, 2015.
[5]. Wu, Dong-Jie, et al. "Droidmat: Android malware detection through manifest and api calls tracing." Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on.IEEE, 2012.
[6]. Avast Report (September, 2017), https://press.avast.com/ avast-reports-40-increase-in-mobile-cyberattacks.
[7]. Desnos, Anthony. "Androguard." https://github.com/androguard/androguard.