International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 12 | Dec 2019 www.irjet.net p-ISSN: 2395-0072

Effective Technique Used for Malware Detection using Machine Learning

Ms. Pooja Kote[1], Ms. Gauri Vinayak Sonawane[2]

[1]Assistant Professor, Sandip University, Maharashtra, Nashik
[2]Research Scholar, Sandip University, Maharashtra, Nashik

Abstract - *Todays Android platform allows developers to take full advantage of the mobile OS, but also raises significant issues related to malicious applications. The main aim on the Android platform growing is malicious apps. Among the various approaches in detecting malware, machine learning-based algorithms have achieved a high accuracy in detecting malware. The increased numbers of applications, on the other hand prepares a suitable prone for some users to develop different kinds of malware and insert them in Google Android market or other third-party markets as safe applications. This paper states the different machine learning techniques used for malware detection in Android.*

Key Words: Machine Learning, Android OS, Android Security, Malware Detection

1. INTRODUCTION

Even if designing a malware is nowadays considered quite common the most advanced programmers try to hide malicious behaviours by using different techniques, such as the repackaging of legitimate applications or the obfuscation/ciphering of code. Malware, short for malicious software, is a general term used to refer to a variety of forms of hostile or intrusive software such as viruses, worms, spyware, Trojan horses, rootkits, and backdoors. Malware can does infect any computing machines running the user programs (or applications), and the propagation and prevention of the malware have been well studied for personal computers.

With the huge malware production now-days, on-machine learning-based Android malware detection approaches are time consuming to detect it as well as its inability to detect unseen malware. Thus, Android devices must implement techniques with the ability to detect malware using machine learning algorithms. Machine learning algorithms have been increasingly applied in security, and perform to improve the effectiveness in android platform. In this perspective, the system aims to spot malware using various combinations of algorithms and detect the malware. The algorithms that we are going to used are Call Graph Based Classification, NN based Classification, Navie-Bayes Based Classification. Most malware detection methods are based on traditional signatures-based, such as a malware definition, and compare each application against the database of known malware signatures.

Being motivated by the increasing number of Apps and the lack of effective approaches and tools, some research tries to detect malware by observing the statistic and/or dynamic behaviour and characters of applications. This paper aims to detect malware using various combination of Machine learning technique and detect the malware. To study various classification techniques.

From various existing approaches in detecting malware, Machine Learning algorithms and techniques achieved a high accuracy in detecting malware. In Today's world, Modern malware uses advanced techniques to hide from static and dynamic analysis tools. To achieve stealthiest when attacking a mobile device, an effective approach is required for the diagnosis of the application.

2. RELATED WORK

2.1Analyzing Log Files for Postmortem Intrusion Detection [1]:

To build a model for ordinary behavior, in the first step, we factor out repetitive behavior in a collection of ordinary (attack-free) log files. As a result, we obtain, first, a compressed version of all log files, and, second, a relation of the sequences of most frequent occurrence across all those logs, which we call across repetitive sequences.

In the second step, we follow a 100-size, 100-step sliding-window approach to analyze every reduced log: starting at the first position of the log, we retrieve a window of size 100, then characterize each window by means of an attribute vector and then slide the window a step of 100 elements to continue with the same procedure

In a third step, we build a model that captures the commonality in the sequence of attribute vectors, representing the original log.

2.2 Evaluating Android Anti-Malware against Transformation Attacks

Methodology describe through the series of transformations applied to different samples and the detection results on various anti-malware tools. Empty cells in the tables indicate positive detection while cells with 'x' indicate that the corresponding anti-malware tool failed to detect the malware sample after the given transformations were applied to the sample.

Each transformation is applied to a malware sample (of course, some like exploit encryption apply only in certain cases) and the transformed sample is passed through anti-malware. If detection breaks with trivial transformations, we stop.

Next, we apply all the DSA transformations. In general there is no well-defined order in which transformations should be applied (in some cases a heuristic works; for example, malware that include native exploits are likely to be detected based on those exploits).

2.3 Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection

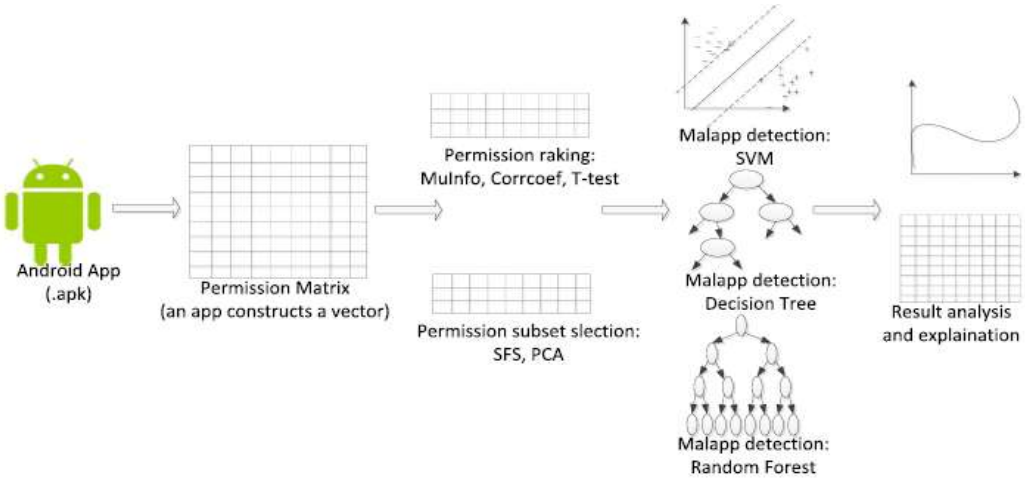


Fig. 1. The methods and process for exploring permission-reduced risks and the detection of malicious applications in three levels.

3. METHODOLOGY

The proposed methodology for exploring permission-induced risk in Android apps. First, we employ three feature ranking techniques to evaluate the risk of granting each permission, based on which the permissions are ranked from most to least risky.

The Next one, instead of individual permission, are evaluated by feature subset selection methods for investigating the risk introduced by the collaboration of several permissions.

The last step, the detection of malapps based on risky permissions is formulated as a classification problem and executed by building classifiers.

Machine Learning Techniques for malware detection in Android platform

In this section we are going to review the different machine learning techniques used for malware detection: SVM, Naïve Bayes, Behavioral based. Following states different techniques with detail explanation:

3.1 Support Vector Machine Based Technique:

SVM is a technique for data classification; it can generate a nonlinear decision plane and classifies data which has non-regular distribution. SVM consist of two phases training phase and testing phase. During the training phase an SVM takes a set of input points in the form of Attribute Relation File Format (ARFF), each of which is marked as belonging to one of two categories, and builds a model representing the input points in such way that the points of different categories are divided by a clear gap that is as wide as possible.

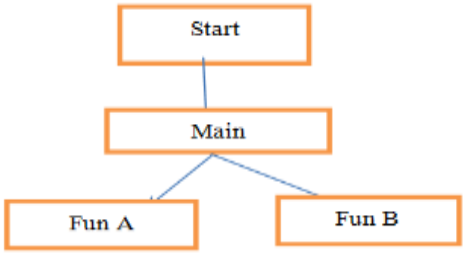
Thereafter, a new data point is mapped into the same space and predicted to belong to a category based on which side of the gap it falls on. The SVM is configured and trained to traverse through two types of features. At first SVM highlights those files whose system calls are having deviating behavior than normal behavior of benign files and other one is SVM pinpoints those files having opcodes that are having positive impact on the classification of benign and malicious software. Finally during testing phase SVM validates the dataset discriminating the files into sets of benign and malicious files.

B.Naive Bayes Based Technique:

Naive Bayes is a classic machine learning algorithm in which we can use all our feature to detect whether they become malicious file or not and used it for the purpose of classification.

A naïve bayes is Bayesians classification which represents statistical learning methods. It is simple technique for constructing classifiers: in this class labels are assign to PI, represent feature vector set. It is based on principle all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.

C.Behavioural Analysis based Technique: In behavioural analysis technique call graph based method is used .A call graphis a directed-graph that represents calling relationship between subroutines in a computer program. Specifically, each node represents a procedure and each edge (f, g) indicates that procedure f calls procedure g. Thus, a cycle in the graph indicates recursive procedure calls. Call graphs are a basic program analysis result that can be used for human understanding of programs, or as a basis for further analyses, such as an analysis that tracks the flow of values between procedures



4. CONCLUSIONS

In this, an effective approach where different machine learning techniques such as Call-graph, Naïve-byes and SVM is used to spot malware in android application. All these are known for better performance, by this it helps to spot complex malware in application. It is generalized approach can be applied to any android application .This approach extended with more additional feature or information to increase accuracy..

In this perspective, the paper aims to detect malware using various machine learning techniques. The different techniques that we described are Call Graph Based Technique, Naïve bayes based Technique, and SVM Based Classification .malware

5. FUTURE WORK

In future work, there will be on growing complex malware to be detected with more accuracy and effectively. It would be desirable to propose new frameworks and algorithms which is light, fast and strong enough to detect the malware. It will be more desirable if fused technique to be use by combining above one or two for more effective result. This approach extended with more additional feature or information to increase accuracy

REFERENCES

[1]Karen A. Garc´ıa, Ra´ulMonroy, Luis A. Trejo, Carlos Mex-Perera, and Eduardo Aguirre, “Analyzing Log Files for Post mortem Intrusion Detection,” IEEE transactions on systems, man, and cybernetics—part c: APPLICATIONS and reviews, vol. 42, no. 6, November 2012.

[2]Wei Wang, Xing Wang, “Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection,” information forensics and security, vol. 9, no. 11, November 2014..

[3]Suleiman Y. Yerima, SakirSezer, Gavin McWilliams, “A New Android Malware Detection Approach Using Bayesian Classification” IEEE 27th International Conference on Advanced Information Networking and Applications,2013.

[4]R. Andriatsimandefitra and V. V. T. Tong, "Detection and identification of Android malware based on information flow monitoring," in Int. Conf.on Cyber Security and Cloud Computing, 2015, pp. 1–4.

[5]V.Rastogi, Y. Chen, and X. Jiang, "Catch me if you can: Evaluating Android anti- malware against transformation attacks," IEEE Trans. On Information Forensics and Security, vol. 9, no. 1, pp. 99–108, Jan. 2014.

[6]Shabtai, U. Kanonov, Y. Elovici, C. Glezer, and Y. Weiss, "Andromaly: behavioral malware detection framework for Android devices,"Journal of Intelligent Information Systems, vol. 38, no. 1, pp. 161–190, 2012.

[7]Bose, X. Hu, K. G. Shin, and T. Park, "Behavioral detection of malware on mobile handsets," in Proc. Int. Conf. on Mobile Systems,Applications, and Services, 2008, pp. 225–238.

[8]Ebtesam J. Alqahtani, RachidZagrouba_and Abdullah Almuhaideb,"A Survey on Android Malware Detection Techniques",Sixth International Conference on Software Defined Systems (SDS),2019

[9]jaweizhu, Zhongchen,," API Sequences based malware detection for android",inProc.int. Conf. On Computer Securit, 2015, pp.673-676.

[10]MdLiakat Ali, Charles C. Tappert, "Keystroke biometric user verification using HMM," in Proc.Int. Conf on Cyber security and Cloud computing, 2016, pp. 204-209.

[11]P. Faruki, V. Ganmoor, V.Laxmi, M. S.Gaur, and A. Bharmal,"AndroSimilar: robust statistical feature signature for Android malware detection," in Proc. Int. Conf. on Security of Information and Networks,2013, pp. 152–159.

[12]W. Mazurczyk and L. Caviglione, "Information hiding as a challenge for malware detection," Security & Privacy, vol. 13, no. 2, pp. 89–93, 2015.

[13]Annachhatre, Chinmayee, "Hidden Markov Models for Malware Classification" (2013). http://scholarworks.sjsu.edu/etd_projects.

[14]Naive Bayes text classification. https://nlp.stanford.edu/IRbook/html/htmledition/naive-bayes- text-classification-1.html

BIOGRAPHIES

Ms. Pooja Kote
Assistant Professor, Sandip University, Nashik.

Ms. Gauri V. Sonawane
Research Scholar in Computer Science and Engineering Department, Sandip University Nasik, Working on AI and ML Platform. Pursuing PhD in Computer Engineering Department from Sandip University Nasik.