Artificial & Computational Intelligence / Published Online : June 2020
https://acors.org/ijacoi/VOL1_ISSUE3_15.pdf

Artificial & Computational Intelligence

A Study on Android Malware Detection System using Static Analysis

BEENU P B
Department Of Computer Science And Engineering
beenupb@cet.ac.in
College Of Engineering Trivandrum, Trivandrum, India

Abstract

Android has gotten one of the most well known open-source working frameworks that have been generally utilized in cell phones. Due to its compatibility with various applications and services, it's been good to go to operating systems. Due to the increased popularity, attackers are more interested in it. Several attacks are occurring in Android smartphones nowadays. So the wide variety of malware samples lead to the failure of several malware detection systems. An efficient malware detection system is to be implemented to detect the malware to provide security to the Smartphone user. The various machine learning techniques are implemented to classify the malware samples. In this study, the comparisons between various detection methods using static features are analyzed.

Keywords: Machine Learning, Malware Detection, Security, Static analysis.

1. Introduction

In the recent past, there have been several technological innovations that have changed the way people live. One such innovation is the development of communication technology. The operations that would take several days are done in a snap of a second. So the entire industry has been completely changed its way that every single impossible task is now within a click away. Smartphones have been one of the boons as well as a bane to mankind. It has several benefits but the defects outline the gains. Smartphones have replaced everything and been to our personal life itself. So the most important aspect is that it should ensure that the user's privacy and security are not violated. But knowingly or unknowingly the security aspect has been breached. One of the main problems is that there are several apps(applications) that would ease our day to day tasks. Without much concern, each one of us installs tons of apps into our smart devices. This is a serious security flaw that is being made at the user end. Nothing is cent percent safe and secure. The smart users are not at all aware because there are millions of malicious apps that have been used by hackers to get into smart devices and get hold of personal data and property.

Android is an open-source and Linux-based Operating System for smartphones. Android was made by the Open Handset Alliance, drove by Google, and various associations. One of the fundamental highlights of android is that the engineer just needs to create for Android and afterward it will have the option to run on different contraptions powered by Android. Android is an unimaginable working structure fighting with Apple 4GS and supports unprecedented features.

Android applications are regularly made in the Java language using the Android Software Development Kit. When made, Android applications can be packaged viably and sold out either through a store, for instance, Google Play or the Amazon App store. Android controls countless mobile phones in more than 190 countries around the world. It's the greatest presented base of any adaptable stage and is growing rapidly. Reliably more than 1 million new Android contraptions are started far and wide.

Malware, or malignant programming, is any program or archive that is perilous. Sorts of malware can fuse PC infections, worms, Trojan horses, and spyware. These malevolent procedures can play out a wide extent of limits, for example, taking, encoding, or deleting fragile data, adjusting or holding onto focus handling limits, and watching customers' PC activity without their assent. Malware can be contracted on a cell phone if the client downloads an easygoing application or in case they click on a compromising link from an email or text. A mobile phone can in like manner get infected through a Bluetooth or Wi-Fi affiliation. Malware is discovered inside and out more for the most part on contraptions that run the Android OS going to iOS gadgets. Malware on Android gadgets is generally downloaded through applications.Signs that an Android gadget is ruined with malware review astonishing increases for data utilization, a rapidly lessening battery charge or calls, messages, and messages being sent to the gadget contacts without the client's assent. Also, if a client gets a message from a clear contact that has all the reserves of being dubious, it might be from a kind of versatile malware that spreads between gadgets. Apple iOS gadgets are just to a great extent ruined with malware considering the way that Apple watchfully vets the applications sold in the App Store. Regardless, it is up to this point helpful for an iOS gadget to be contaminated by opening a dim affiliation found in an email or content. iOS devices will wind up being dynamically frail if jail broken.

India is one of the most popular destinations for cell phone companies. Due to the open idea of free source working framework Android, has become well known among the market. Due to its expanded ubiquity, it has been the essential objective for attackers. The number of malware that is being delivered each year is expanding step by step. The Fig.1 shows the statistics of growth in malware samples produced each year.

Rising strains of malware consolidate new shirking and indefinite quality frameworks that are expected to deceive customers just as security administrators and threaten malware things as well. A segment of these evasion techniques relies upon direct systems, for instance, using web mediators to cover dangerous traffic or source IP addresses. Progressively perplexing perils consolidate polymorphic malware, which can on and on change its fundamental code to keep up a vital good ways from ID from signature-based disclosure gadgets, antagonistic to sandbox strategies, which license the malware to distinguish when it is being separated and delay execution until after it leaves the sandbox, and fileless malware, which lives just in the structure's RAM in order to refrain from being found. So the location of malware is a genuine test in the current situation.

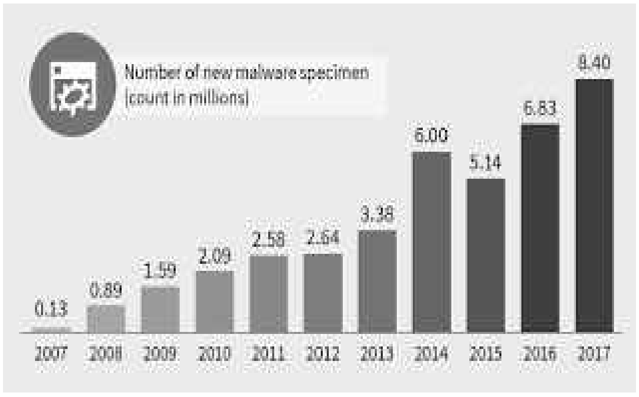

Fig. 1. Graph showing the number of Malware samples produced every year (From Internet)

2. MALWARE DETECTION SYSTEMS

Numerous recognition frameworks have been planned earlier, but it has certain constraints with regards to precision and efficiency. Broadly, the examination system has been separated into two classifications namely, Statical and Dynamical analysis. The statically investigation will remove highlights from the source code whereas dynamical examination breaks down the malignant conduct by setting up a runtime condition of the application. The paper"An Efficient Android Malware Detection System Based on Method-Level Behavioral Semantic Analysis" [1] based on the method level correlation relationship of app's abstracted API calls. The framework configuration depends on the presumption that the malevolent application will utilize a specific determined blend of API calls. So so as to recognize the malware contaminated applications, abstracted API calls are used. The connections between the preoccupied API calls are dissected utilizing association rule analysis. The certainty esteems which is the resultant of the examination is at that point taken care of into a classifier that has been created utilizing AI, for example, k-NN, SVM, Random

Forest. By applying abstraction in the framework, it prompts a significant decrease in the number of API calls and adapts up to the updates in the Android framework. Also it catches the semantic conduct of the application which will have an additional advantage while computing the certainty value. The fundamental confinement of this framework is that it won't distinguish the encoded assaults.

The paper titled ”MAMADROID: Detecting Android Malware by Building Markov Chains of Behavioral Models" [2] detects the malicious app using Markov Chains. The API calls are abstracted and afterward, Markov Chains are utilized to speak to the arrangement of API calls. The framework comprises four steps, namely, Call graph extraction, Sequence extraction, Markov-chain based modeling, Classification. Call graph extraction is the place the static examination on the APK record is done. The sequence extraction comprises recognizing the entry nodes from which the reachable way could be found out. Then by building the Markov-chain based model is made from which highlights are extricated and selected. The yield is then taken care of into the classifier which utilizes Random Forest,1-NN,3-NN, SVM to distinguish whether the application is malevolent or not. This model is adaptable to the API changes yet expends more memory.

In this paper titled ”DroidAPIMiner: Mining API-Level Features for Robust Malware Detection in Android” [3] defines a system which will identify the malicious activity by analyzing the byte-code of the application. The data in the API level, package level, and parameter level will give adequate data with respect to the semantic conduct of the application. The framework comprises three phases, Feature extraction, Feature refinement and Models learning & generation. The feature extraction and refinement will remove the crude and refined highlights respectively. The last advance comprises grouping dependent on the features. The classifiers utilized are ID5 DT,C4.5 DT,k-NN,SVM.A general information mining approach is utilized in the system. As a consequence it experiences a lot of misclassified results. But it is one of the lightweight classifiers created.

In”DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket” [4] provides a lightweight method that performs a broad static analysis. The method consists of Broad static analysis, embedding in vector space, Learning based detection and Explanation. The first step is to extract features as a set of strings. The feature sets regarding each component is as follows:

• Feature sets from the manifest file:The features corresponding to :

– Hardware components

– Request permissions

– App components

– Filtered intents

• Feature sets from disassembled code: The features from :

– Restricted API calls

– Used Permissions

– Suspicious API calls

– Network addresses

The highlights removed are made into a vector space utilizing hash tables. The discovery depends on AI which utilizes the algorithm, SVM. The included bit of leeway of the framework is that it gives the clarification dependent on the suspicious vindictive conduct that has been distinguished by the system. This framework is unequipped for recognizing the change assaults.

In the paper ”DroidMat: Android malware detection through manifest and API calls tracing” [5] defines a system which extract the static information regarding the permissions, components deployed, intent message passing and API calls from the manifest file.At that point, the functional behavior is perceived by grouping by k-means and EM algorithm. The last stage comprises distinguishing the malware which utilizes the AI procedures, for example, k-NN,Naive Bayes algorithm. This model underlines the consents that have been mentioned by the app, only in light of the fact that the Android security model essentially relies upon the authorization based mechanism. One of the bit of leeway is that it doesn't require sending in the run time which spares a lot of cost just as time. But this framework couldn't recognize the malware tests that have a place with the families, Base Bridge, DroidKungFu.

The main highlight of the system, Crowdroid [6] is that it distinguishes between the applications having the same name and version. It gets the sign of the application conduct utilizing a publicly supporting system. The remote server plays out the examination procedure whereas the lightweight customer is on the cell phone itself. The server performs information control which comprises parsing of the information and makes a framework call vector. Then every datum set is grouped by utilizing a partition clustering calculation, for example, k-means. The general procedure could be condensed into three phases, Data acquisition, Data manipulation, and Malware investigation & detection. This framework is viable if the information given by the various clients correctly if the information given is off base or ruined the framework flops in the end.

MADAM [7] is one of the state of art technique which uses the dynamic analysis technique.Actually it is Multi-Level Anomaly Detector for Android Malware. It is a host based malware detection system.This system extracts five group of features from four different levels.The different levels are:

• Level I - Kernel Level
• Level II - Application Level
• Level III - User Level
• Level IV- Package Level

The architecture of the system consists of four blocks:

• App Risk Assessment
• Global Monitor
• Per-App Monitor
• User Interface and Prevention

The principal module comprises an App evaluator which computes the hazard score and makes an App suspicious list. The activity is totally straightforward to the user. The worldwide screen is a fundamental part of the system. It makes a highlight vector to reproduce the conduct of the application. This module comprises three modules namely, System call monitor, User Activity & Message monitor and Action logger & Classifier. These modules ceaselessly screen the application during run-time. The next module comprises a mark based plan where the noxious application could be distinguished easily. The last module gives the outcome of whether the application is favorable or not. The framework has high precision however a lot of vitality utilization because of its dynamic nature.

HinDroid [8] implements a high level semantics that is very difficult for the attackers to evade the detection.The API call relationships are spoken to through an organized heterogeneous data network. From the system, a meta way is made so as to show the similarity measure. The Multi-kernel learning approach is utilized to arrange this similarity measure by preparing the network. The framework comprises the accompanying modules such as Unzipper &Decompiler, Feature Extractor, HIN Constructor, Multi-part learner, Malware Detector.

This is one of the least complex models which will recognize the pernicious movement by contrasting it and the predefined model of malware samples. First of the considerable number of highlights are extricated dependent on permissions, API calls.The model matching is done in three steps :

• Permission model
• Android Framework API calls model
• API calls chains model

Therefore there will be a rundown of coordinating applications which is like the past predefined models. This framework is utilized as a partner instrument on the off chance that manual investigation is done. It will offer a second input to the analyzer. The primary restriction of the framework is that the exactness of the framework, for the most part, relied on the determination of the predefined tests.

In ”Dalvik opcode graph based Android malware variants detection using global topology features” [10], a light weight method is illustrated based on graph and information theory.Actually the system works both online and offline.The technique comprises of Dalvik opcode diagram construction, Dalvik opcode graph pruning, Global topology feature extraction, and Similarity searching. Dalvik grouping is the portrayal of the source code and could be deciphered utilizing DalvikVM.From the succession, a diagram is made where every node compares to the activity and edge compares to the relationship among them. The weight that is in the vertex is the probability of co-occurrence of the relation. The unpredictability of the diagram is then diminished utilizing data hypothesis yet this pruning will hold the malware sensitive data. Then topology highlights are separated which speak to the conduct globally. Then the Manhattan distance between the predefined and target test is calculated. This framework isn’t so compelling as some malware tests could without much of a stretch devour the framework.

3. Conclusion

Every system has its own limitations and advantages.So in order to reach up with the challenging attacks more enhanced mechanisms should be implemented.By constantly updating the database as well as the extraction techniques we could achieve a considerable success rate.

Acknowledgement

I might want to communicate my extremely incredible thankfulness to Dr. Ajeesh Ramanujan for his significant and valuable recommendations during the arranging and improvement of this examination work. His ability to give his time so liberally has been especially appreciated. I might likewise want to thank my companions and guardians who have given steady consolation over the span of this work.

References

[1] Hanqing Zhang , Senlin Luo , Yifei Zhang, and Limin Pan, “An Efficient Android Malware Detection System Based on Method-Level Behavioral Semantic Analysis”, IEEE Access,vol. 7,2019 ,pp. 69246 - 69256.

[2] L. Onwuzurike, E. Mariconti, P. Andriotis, E. De Cristofaro, G. Ross, and G. Stringhini, “MAMADROID: Detecting Android Malware by Building Markov Chains of Behavioral Models”,ACM Trans. Privacy Secur., vol. 22, no. 2, 2019, p. 14.

[3] Y. Aafer, W. Du, and H. Yin,” DroidAPIMiner: Mining API-Level Features for Robust Malware Detection in Android”,in Proc. Int. Conf. Secur. Privacy Commun. Syst. Cham, Switzerland: Springer, 2013, pp. 86-103.

[4] 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. Annu. Symp. Netw. Distrib. Syst. Secur. (NDSS), vol. 14, 2014, pp. 23-26.

[5] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu,” DroidMat: Android malware detection through manifest and API calls tracing”,in Proc. 7th Asia Joint Conf. Inf. Secur., Aug. 2012, pp. 62-69.

[6] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, “Crowdroid: Behavior Based malware detection system for Android”,in Proc. 1st ACM Workshop Secur. Privacy Smartphones Mobile Devices, 2011, pp. 15-26.

[7] A. Saracino, D. Sgandurra, G. Dini, and F. Martinelli,”MADAM: Effective and efficient behavior-based Android malware detection and prevention”, IEEETrans. Depend. Sec. Comput., vol. 15, no. 1,Jan./Feb. 2018, pp. 83-97.

[8] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu,”HinDroid: An intelligent Android malware detection system based on structured heterogeneous information network”,in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2017, pp. 1507-1515.

[9] Anastasia Skovoroda,AnastasiaSkovoroda,”Automated static analysis and classification of Android malware using permission and API calls”,in 15th Annual Conference on Privacy, Security and Trust (PST), 2017,pp. 243-252.

[10] J. Zhang, Z. Qin, K. Zhang, H. Yin, and J. Zou, “Dalvik opcode graph based Android malware variants detection using global topology features“,IEEE Access, vol. 6,2018, pp. 51964-51974..