ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 9, September 2018

Detection of Malware Using Machine Learning Algorithms

Vrushal R Patil, Shital Jadhav

M. E Student, Department of Computer Engineering, GHRIEM Jalgaon, India

Assistant Professor, Department of Computer Engineering, GHRIEM Jalgaon, India

ABSTRACT: In the recent years smart phones are widely used and with that malwares also come with mobile phones like with the Android phones and that malware cause harm to user information. And the big problem is how to detect new malware and malicious. And by observing that develop the feature extraction method by using binary problem, this paper represent the method for feature extraction of java source code. This method uses the Keyword Correlation Distance for computing the correlation between key codes such as API calls, Android permissions, the common parameters, and also the common keywords in Android malware source code. After that use the SVM (Support Vector Machine) for the system gain to assists the function of new malware software sample, so as to detect new malicious software and existing malwares. This method is different from the conventional methods which are based on the context of the text. This method combines the characteristics of the malicious software categories and operating environment to record the behavior of the malicious software. Experiments show that the method is efficient and effective in detecting malwares on Android platform.

KEYWORDS: Malware; Keywords Correlation Distance; SVM

I. INTRODUCTION

Android has become the most popular mobile OS worldwide. Meanwhile, the number of Android applications (apps) grows exponentially, which has significantly benefited the daily lives for mobile users. According to Symantec Internet Security Threat Report there were more than three times as many Android apps classified as malware in 2015 than in 2014 [1]. The private data of the users, such as contacts list, and other user specific data, are the primary target of the Android malware, which has imposed a serious threat for the security and privacy of mobile users [6]. Unfortunately, most malware detection methods are based on traditional content signatures, such as a list of malware signature definitions, and compare each application against the database of known malware signatures. The disadvantage of this detection method is that users are only protected from malware that are detected by most recently updated signatures, but not protected from new malware. Some researches dynamically run the App On the sandbox to capture runtime activities of the App. But analyzing Apps' runtime dynamic behaviors requires sophisticated skills and platforms which is time consuming process and will cause high cost overhead [8]. Motivated by the above observations, propose feature extracted method based on the keywords vector. A malware identification strategy through SVM is used in view of the element vector set, which can distinguish new malwares

II. REVIEW OF LITERATURE

Rudi L, Paul M.B. [1]. The Google similarity distance, of words and phrases from the WWW using Google page counts. Author gave applications in hierarchical clustering, classification, and language translation. Author gave examples to distinguish between colors and numbers, cluster names of paintings by 17th century Dutch masters and names of books by English novelists, the ability to understand emergencies and primes, and we demonstrate the ability to do a simple automatic English-Spanish translation. Finally, author use the WordNet database as an objective baseline against which to judge the performance of our method.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 9, September 2018

A.P. Fuchs, A. Chaudhuri, and J.S. Foster [2]. SCANDROID's analysis is modular to allow incremental checking of applications as they are installed on an Android device. It extracts security specifications from manifests that accompany such applications, and checks whether data flows through those applications are consistent with those specifications. To our knowledge, SCANDROID is the first program analysis tool for Android, and we expect it to be useful for automated security certification of Android applications.

Y. Zhou, Z. Wang, W. Zhou, and X. Jiang [3]. A permission based behavioral foot printing scheme to detect new samples of known Android malware families. Among those malicious apps, proposed system also uncovered two zero-day malware (in 40 apps): one from the official Android Market and the other from alternative marketplaces. The results show that current marketplaces are functional and relatively healthy.

Peiravian N, Zhu X. [4]. By using permissions and API calls as features to characterize each Apps, one can learn a classifier to identify whether an App is potentially malicious or not. An inherent advantage of proposed method is that it does not need to involve any dynamical tracing of the system calls but only uses simple static analysis to find system functions involved in each App. In addition, because permission settings and APIs are always available for each App, proposed method can be generalized to all mobile applications.

Chang C C, Lin C J. [5]. LIBSVM is a library for Support Vector Machines (SVMs). The goal is to help users to easily apply SVM to their applications. LIBSVM has gained wide popularity in machine learning and many other areas. In this article, author present all implementation details of LIBSVM.

Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and CERT Siemens. [6]. Propose DREBIN, a lightweight method for detection of Android malware that enables identifying malicious applications directly on the smartphone. As the limited resources impede monitoring applications at run-time, DREBIN performs a broad static analysis, gathering as many features of an application as possible. These features are embedded in a joint vector space, such that typical patterns indicative for malware can be automatically identified and used for explaining the decisions of our method.

Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, and Kuo- Ping Wu. [7]. Propose a static feature-based mechanism to provide a static analyst paradigm for detecting the Android malware. The mechanism considers the static information including permissions, deployment of components, Intent messages passing and API calls for characterizing the Android applications behavior.

Yousra Aafer, Wenliang Du, and Heng Yin. [8]. Mitigate Android malware installation through providing robust and lightweight classifiers. We have conducted a thorough analysis to extract relevant features to malware behavior captured at API level, and evaluated different classifiers using the generated feature set.

William Enck, Machigar Ongtang, and Patrick McDaniel. 2009. [9]. Propose the Kirin security service for Android, which performs lightweight certification of applications to mitigate malware at install time. Kirin certification uses security rules, which are templates designed to conservatively match undesirable properties in security configuration bundled with applications.

Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. 2011. [10]. Android applications to determine whether Android developers follow least privilege with their permission requests. Author built Stowaway, a tool that detects over privilege in compiled Android applications. Stowaway determines the set of API calls that an application uses and then maps those API calls to permissions. Author used automated testing tools on the Android API in order to build the permission map that is necessary for detecting over privilege.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 9, September 2018

III. SYSTEM ANALYSIS

The proposed model extracts method based on the keywords vector. Every keywords vector is a set of keywords which can common complete a malicious attack. We know only some request may be no harm to users. Harm is often done by a series of malicious operations. Our system is mainly divided into two modules. One is feature extraction module; the other is machine learning module. Every module includes several parts. The feature extraction module is responsible for the feature extraction and the machine learning module is responsible for classification and decision making. In this model we use the NGD idea to computer the keywords similarity distance. With the same or similar meanings in a natural language sense tend to be "close" in units of Normalized Google Distance, while words with dissimilar meanings tend to be farther apart. So we use Keywords Correlation Distance (KCD) to represent the keywords correlation in software. We present a classification method based on SVM (Support Vector Machine). SVM is a supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis.

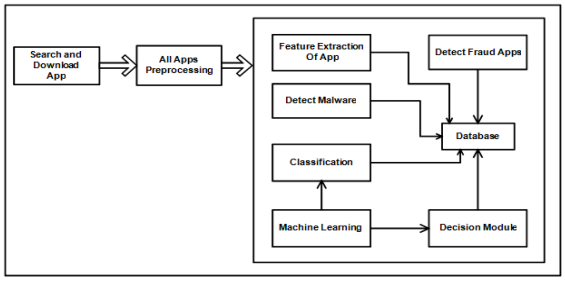

Fig 1: Proposed System Architecture

IV. EXPERIMENT RESULTS

Android uses several partitions (including boot, system, recovery, data etc.) to organize files and folders in the device with each partition having its own functionality. Due to our research mainly focusing on the identification of malware instead of studying Android system, we only consider data partition which might be objectionable for the user. For experimental analysis of the system, permission dataset was created that consist of different dangerous permissions. Dataset for experimental analysis is created that consist of app ID, apk file of the app, category of app, dangerous permissions that are necessary for proper functioning of the app and images related to app. Sample dataset is as shown below

ID	APK Name	Necessary Permission
1	Gmail_v8.4.8.194949231.release_apkpure.com.apk	BLUETOOTH_ADMIN,INTERNET, ACCESS_WIFI_STATE, ADD_VOICEMAIL RECEIVE_SMS

Table 1. Sample Database

Android applications are distributed as Android Package (APK) files. APK files are signed ZIP files that contain the app's bytecode along with all its data, resources, third-party libraries and a manifest file that describes the app's capabilities. Apk files for the android app are downloaded from internet to use in dataset. To decompile the apk file, this dataset is uploaded through admin panel to store in the database. SVM algorithm extracts the permissions from entity extract table of database. These extracted permissions are compared with permission dataset already prepared. If

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 9, September 2018

the SVM algorithm founds count of comparison greater than simple count, then detection of malware is indicated. SVM ignores necessary permissions from permission dataset while comparison with extracted permissions. If count of comparison founds lower than simple count, then SVM indicated non-finding of malware.
In a playstore, for particular category, applications are ranked based on a combination of ratings and reviews. Proposed system also analyzes reviews by the user for particular apps. In some cases, rating of particular app is improved by providing multiple ratings and good reviews by some users. This may leads to the wrong interpretation about the app. Some apps may get good rating because of such malpractice. Proposed system uses PCF algorithm that extracts app ID and review dates. If for the same app ID, on a same day, single user has offered more than one review, then string matching algorithm matches the strings of the reviews. If reviews found same on Navie-baise algorithm, then system considers the app as suspicious app. For Navie-bais algorithm, dictionary is prepared in which love, good etc words are added in positive dictionary while unlike, bad etc words are added in negative dictionary. This helps users to identify the suspicious apps as well as bias reviews and ratings. Following table describes category wise classification of normal apps, malware apps and suspicious app.

Normal App	Malware App	Suspicious App
85%	10%	5%
75%	12%	13%
91%	5%	4%
91%	4%	5%
79%	12%	9%
95%	3%	2%

Table 2: Comparison of Apps Category

In the below graph, I have shown the accuracy of proposed system is more accurate than existing. In the proposed system, I checked the permission that dangerous and normal and find out the accurate result of system.

System	Accuracy Level
Google distance clustering	74%
HMM	85%
SVM with KCD	87%
API call	90%
Proposed System	95%

Table 3: Comparison of accuracy in proposed and existing system

V. CONCLUSION

In our system, we propose an extraction method of Android malware feature based on KCD. Then we combine the feature into keywords feature vector. Finally, learn and decision by SVM to detect new malware and malicious variant. Experiments show the method is effective.

VI. FUTURE SCOPE

In future think about the, Develop such hybrid antimalware to provide better security for android devices. Extend work in future; one of the available methods is to have a password mechanism. In password mechanism the user will be

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 9, September 2018

provided with a unique code or password. When network moderator has a doubt of misuse of a particular user app the user can be asked for password confirmation.

.

REFERENCES

[1] Rudi L, Paul M.B. The Google similarity distance [J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3):370- 383.
[2] A.P. Fuchs, A. Chaudhuri, and J.S. Foster. Scandroid: Automated security certification of android applications. Manuscript, Univ. of Maryland, http://www.cs.umd.edu/ avik/projects/scandroidascaa, 2009.
[3] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. In Proceedings of the 19th Annual Network & Distributed System Security Symposium, Feb. 2012.
[4] 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.
[5] Chang C C, Lin C J. LIBSVM: A library for support vector machines [J]. Acm Transactions on Intelligent Systems & Technology, 2011, 2(3):27. https://www.csie.ntu.edu.tw/~cjlin/
[6] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, Konrad Rieck, and CERT Siemens. Drebin: Effective and explainable detection of android malware in your pocket. In Proceedings of 2014 Network and Distributed System Security Symposium (NDSS 2014), February 2014.
[7] Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee, and Kuo- Ping Wu. Droidmat: Android malware detection through manifest and API calls tracing. In Proceedings of Seventh Asia Joint Conference on Information Security (Asia JCIS 2012), pages 62–69. IEEE, 2012.
[8] Yousra Aafer, Wenliang Du, and Heng Yin. DroidAPIMiner: Mining API-level features for robust malware detection in android, in Proceedings of 9th International ICST Conference on Security and Privacy in Communication Networks (SecureComm 2013), pages 86– 103. Sydney, Australia, September 2013.
[9] William Enck, Machigar Ongtang, and Patrick McDaniel. 2009. On lightweight mobile phone application certification. In Proceedings of the 16th ACM conference on Computer and communications security (ACM CCS '09)., Chicago, IL, USA, 235-245.
[10] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. 2011. Android permissions demystified. In Proceedings of the 18th ACM conference on Computer and communications security (ACM CCS '11). Chicago, IL, USA, 627-638.