ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

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

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

A Review on Malware Detection on Android Smartphones Using Keywords Vector and SVM

Vrushal R. Patil[1], Prof. Shital Jadhav[2]

Research Scholar, Department of Computer Engineering, G.H. Raisoni College of Engineering and Management, Pune, India [1]

Assistant Professor, Department of Computer Engineering, G.H. Raisoni College of Engineering and Management, Pune, India [2]

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 use the Keyword Correlation Distance for computing the correlation between key codes such as APL 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. 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 [1]. 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 [4]. But analyzing Apps' runtime dynamic behaviors requires sophisticated skills and platforms which is time consuming process and will cause high cost overhead. Motivated by the above observations, propose feature extracted method based on the keywords vector [6]. Every keywords vector is a set of keywords which can common complete a malicious attack. Only some request may be no harm to users. Harm is often done by a series of malicious operations.

The primary focus of this paper is 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

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

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

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

signatures. 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.

II. LITERATURE SURVEY

Rudi L, Paul M.B. [1]. The Google similarity distance, of words and phrases from the WWW using Google page counts. The WWW is the largest database on earth, and the context information entered by millions of independent users averages out to provide automatic semantics of useful quality. We give applications in hierarchical clustering, classification, and language translation. We give 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, we use the WordNet database as an objective baseline against which to judge the performance of our method.

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, our system also uncovered two zero-day malware (in 40 apps): one from the official Android Market and the other from alternative market-places. The results show that current marketplaces are functional and relatively healthy. However, there is also a clear need for a rigorous policing process, especially for non-regulated alternative marketplaces.

Peiravian N, Zhu X [4]. By using permissions and API calls as features to characterize each Apps, we can learn a classifier to identify whether an App is potentially malicious or not. An inherent advantage of our 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, our method can be generalized to all mobile applications. Experiments on real-world Apps with more than 1200 malware and 1200 benign samples validate the algorithm performance.

Chang C C, Lin C J. [5]. LIBSVM is a library for Support Vector Machines (SVMs). We have been actively developing this package since the year 2000. 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, we 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. In order to recognize different intentions of Android malware, different kinds of clustering algorithms can be applied to enhance the malware modeling capability.

IJIRSET

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

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

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

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. [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. Use a variant of security requirements engineering techniques to perform an in-depth security analysis of Android to produce a set of rules that match malware characteristics.

Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. [10]. Android applications to determine whether Android developers follow least privilege with their permission requests. We 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. We used automated testing tools on the Android API in order to build the permission map that is necessary for detecting over privilege.

III. PROPOSED SYSTEM ARCHITECTURE

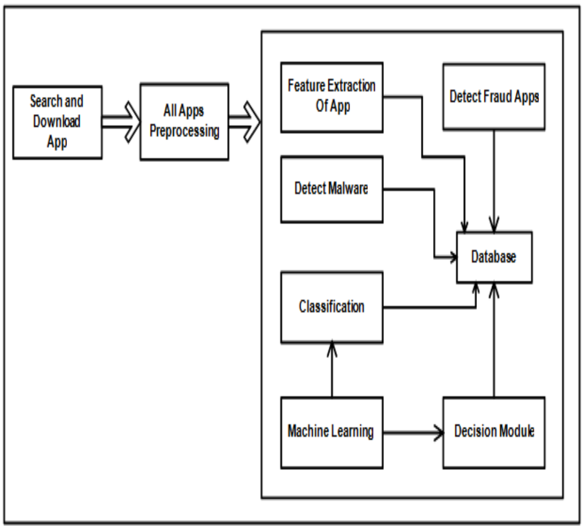

Fig 1: System Architecture

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

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

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

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

supervised learning model with associated learning algorithms that analyze data used for classification and regression analysis.

The proposed system advantage is that users are not only protected from malware that are detected by most recently updated signatures, but also protected from new malware.

IV. MATHEMATICAL MODEL

Mapping Diagram:

Where, A=Applications M1, m2, m3=Malwares D=Detect malware occurs in the application (like m1)

Set Theory:

S=s, e, X, Y, Where, s = Start of the program. 1. Log in with webpage. 2. Upload Applications.
e = End of the program. Detection of malware occurred in application. X =
A, Nm
X = Input of the program. F = Application. Nm = Number of Malwares
Y = Output of the program.

Firstly search and download application and also give review to applications. If any malwares occurs in application then detect that malware and safe to user.

X, Y, U
Let U be the Set of System.
U= Client, M, D
Where, Client, M, C are the elements of the set. Client=Admin M=Malwares in application
D=Detect malware from application.

Space Complexity: The space complexity depends on Presentation and visualization of discovered patterns. More the storage of data more is the space complexity.

Time Complexity: Check No. of patterns available in the datasets= n If (n¿1) then retrieving of information can be time consuming. So the time complexity of this algorithm is O. Failures and Success conditions.

Failures:
1. Huge database can lead to more time consumption to get the information.
2. Hardware failure.
3. Software failure.

Success:
1. Get Proper information of application.
2. Detect malwares occurred in application.

V. CONCLUSION

Smart-phones are becoming popular in terms of power, sensor and communication. Modern, smart-phones provide lots of services such as messaging, browsing internet, emailing, playing games in addition to traditional voice services. Due to its multi-functionality, new security threats are emerged for mobile devices 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.

ISSN (Online): 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

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

Visit: www.ijirset.com

Vol. 7, Issue 1, January 2018

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.