International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume. 1, Issue 4, August 2014, PP 32-37
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Android Application Analyzer

Shaikh Bushra Almin
Department of Information Technology
Pillai's Institute of Information Technology
Navi Mumbai, India
skbushra78691@gmail.com

Madhumita Chatterjee
Department of Computer Engineering
Pillai's Institute of Information Technology
Navi Mumbai, India
mchatterjeee@mes.ac.in

Abstract: *Android is an open source that is based on the modified version of Linux. Due to its openness, it offers a unified approach to application development. In the world of smart phones, applications are the most important part of the success chain. Android-based Smartphone users can get free applications from Android Application Market. However, these applications were not certified by legitimate organizations and they may contain malware applications that can steal privacy information of the users. In this work, a secure system to identify malicious android applications is proposed to help users comprehend security risk. The proposed framework intends to develop a advisory system on Android to identify and remove malicious applications, to enhance security and privacy of Smartphone users. The system is intended to follow a user-centric approach which means that it would be designed keeping the user in mind and for the user. It aims on reducing the burden of analyzing the risk of applications on the user. The proposed system extracts various permissions requested by applications and analyses whether the application is harmful or safe.*

Keywords: *Android; Malware; Permissions; App; Cluster; Malicious; Benign.*

1. INTRODUCTION

The numbers of Android users are increasing day by day due to third party apps provided by Android developers on the market. However with an increasing number of users, an increasing number of security threats targeting mobile devices have also emerged. The users are more attracted towards Android platform because of the fact that these apps are freely downloadable, having most of the useful feature that the user needs in day to day life, such as free messaging, social networking, entertainment, etc. This has made Android a real target for many attackers and has resulted in rise of malicious app. Although android provides a permission mechanism to restrict an access to system information or user's information and through this feature it tries to minimize the damages that could be caused by the malicious apps. The malware writers takes an advantage of the fact that the user doesn't understand the permissions requested by application, will grant all permission at installation time in order to use that application and tries to misuse the system by requesting dangerous permissions. The malicious activity ranges from the stealing user's private information and sending on the internet to signing/subscribing the users to send SMS to premium numbers, etc.

In order to protect the user getting affected from this malicious activities, a secure system is needed which would help the user in identifying and removing those malicious / harmful applications , protecting user's personal information and thereby securing android phone. Hence the proposed work attempts to identify such harmful applications based on the permission requested by them. Initially a clustering algorithm such as k-means is applied on the permissions of installed applications to categorize all the installed applications into one of those malicious applications. Once an application is categorized then a classification algorithm such as naïve Bayesian is used to accurately classify whether an application is benign or malicious one.

The rest of this paper is organized as follows: Section 2 provides some related works; Section 3 and 4 describes motivation and problem statement respectively; Section 5 gives an overview/background of Android OS and its architecture; Section 6 outlines the proposed system in detail; finally, we make our conclusion in Section 7.

2. RELATED WORK

There are many related work on Android security to make sure that the user's data and privacy are not compromised. Takayuki et al. proposed reputation based security evaluation ***[1]***. Initially a rule set is made to decide whether the combination of permissions is with risk. At the installation time of an application, the system searches the combination of permissions of the application from those in the rule set and if the system finds the combination in the rule set, the result of risk assessment of the application is presented to the user and the user decides whether to install or delete it based on the presented information. However a problem with this system is that the risk calculation involves the number of downloads and user rating or review from the market which itself cannot be trusted.

Suleiman Y. et al. have presented an effective approach to alleviate the problem of detecting malicious app based on Bayesian classification models obtained from static code analysis ***[2]***. The models are built from a collection of code and app characteristics that provide indicators of potential malicious activities. These indicators then form the basis for Bayesian classifier, which is used to determine whether a given Android app is harmless or suspicious. However, static code analysis is very time consuming and it is not done in runtime.

Agematsu, H et al have introduced a system called as ADMS: an application development and management system that is operated and maintained by application developers and the market manager ***[3]***. It focuses on three things as follows: a security manager, an event notification, and a market manager. It requires all application developers to insert an event notification code into applications to tell every event to the security manager whenever an application launches a security-related event, and market manager to remove all such applications that don't include the event notification code.

Ghorbanian, M. et al have proposed a host based intrusion detection model in their research work ***[4]*** which covers four sections: Log File Reading, Log File Analysis, Controlling Output, and Storage. There is a collecting and viewing system output mechanism which is provided by the Android logging system. The Log Files are inserted by logcat command and are given to Analysing module. The module decodes and pre-processes the records, and then invokes the matching engine to detect intrusions. At the last step, the matching results are sent to Output Control Module which chooses to alert or record into log files regarding to the results.

Dong-Jie Wu et al. have implemented a DroidMat, a static feature-based mechanism to provide a static analysis paradigm for detecting the Android malware. This mechanism considers the static information including permissions, deployment of components, Intent messages passing and API calls for characterizing the Android applications behavior. As a part of static behavior analysis, the DroidMat extracts the information (e.g., requested permissions, Intent messages passing, etc.) from each application's manifest file, and regards components (Activity, Service, Receiver) as entry points drilling down for tracing API Calls related to permissions. It applies K-means algorithm that enhances the malware modeling capability and then uses kNN algorithm to classify the application as benign or malicious as a part of functional behavior recognition ***[5]***.

Wei Tang et al. proposed a new extending of Android Security Enforcement with a Security Distance (SD) model, ASESD, to mitigate malware ***[6]***. To express the difference between permission and permission combinations with security problems, proposes a set of SD rules. SD is a figure that represents the risk level of permission pairs and SD rules defines how to determine SD of certain permission pairs. A permission combination's SD is the quantitative representation of the security threat this combination may cause. A permission pair's SD is consisting of a threat point which represents the danger level and related characters. Permission combinations with SD of high threat point means applications with this permission combination have a big chance to threat mobile phone security. Permission pairs that have SD with low threat point indicate a safe application.

3. MOTIVATION

If we want to use the OS on tablets or mobiles for business/personal use, security will have to be our number one priority and the malicious software is an unfortunate reality on popular platforms and through its features Android tries to minimize the impact of malware. However, even

unprivileged app that gets installed on an Android device (perhaps by pretending to be a useful application) can still temporarily wreck the user's experience. Users in this unfortunate state will have to identify and remove the hostile application. When user tries to remove any of the unwanted features from his/her phone thinking that it can cause damage, it results in affecting other apps. It is because the affecting apps need that features for its functionality. Generally a user's response to annoying, buggy or malicious software is simply to uninstall it. If the software is disrupting the phone enough that the user can't uninstall it, they can reboot the phone (optionally in safe mode, which stops non-system code from running) and then remove the software before it has a chance to run again. Unfortunately Android doesn't provide any method to prevent the harm done by those apps since it has already given a permission to perform what those apps wants at the time of installation.

Thus it gives a motivation or a need to consider how to keep users safe by providing a secure advisory system as well as how to deal with the issues of isolating those harmful apps from a safe one and still allow a user to enjoy other apps without worrying about its permissions

4. PROBLEM STATEMENT

In order to protect user's data and privacy, comprehend security risk and reduce administrative burden, a secure solution is needed which would help the users in identifying whether an installed app is a harmful or a safe one. A simple solution for this would be to analyze the permissions accessed by the installed apps and determine whether an app is harmful or not and then inform the same to the user. In this way we can protect the user by providing him/her with information of apps that could be harmful. In addition to this, we must also provide a way which will let user to remove those harmful apps and thereby secures the Android phone.

5. BACKGROUND INFORMATION

Android is a Linux-based operating system designed primarily for touch-screen mobile devices such as Smartphone and tablets. It is divided into four layers from top to bottom; the Linux kernel is responsible for the management of hardware resources. The middleware and Libraries run on the top of Linux. The application developer can call the application API that provided in the third layer, and is executed within a Virtual Machine called Dalvik.

Android defines four component types. Activity is used to define the interface. Service used to define the program running in the background. Content provider store and share data using a relational database interface. Broadcast receiver act as mailboxes for message from other application. For security, Android mediates IPC based on permission labels. A permission label is simply a unique text string that can be defines by developer. A list of permission labels required in its package manifest. Android assigns permission to the application at the install time and the Permissions can't be changed at runtime. However, requested permissions are not allowed ***[7].***

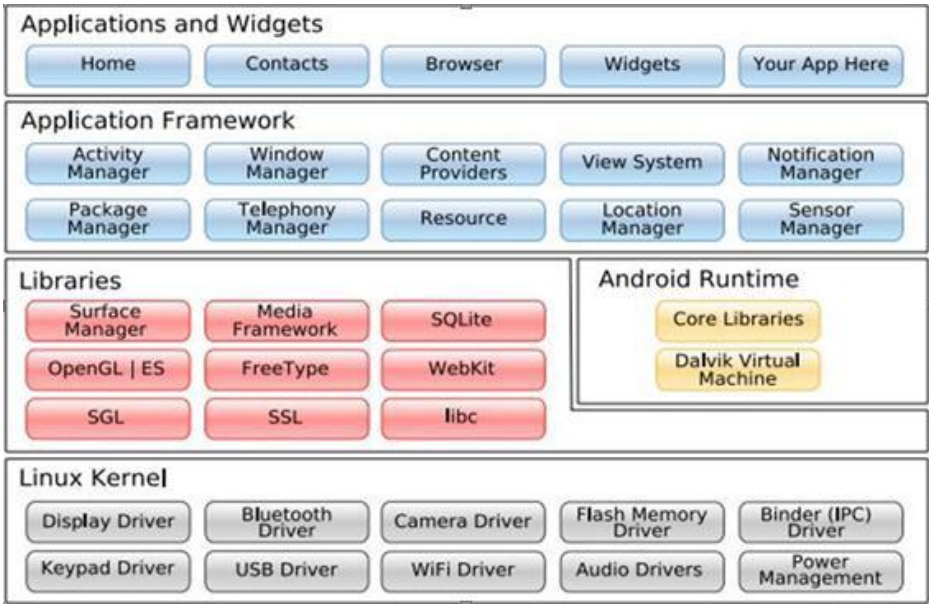

Fig1. *Android System Architecture [7]*

6. PROPOSED SYSTEM

The proposed system an "Android Application Analyzer" allows a user to identify harmful or malicious apps installed on his/her phone by analysing the permissions requested by those apps. The system analyses the permissions using techniques such as clustering and classification.

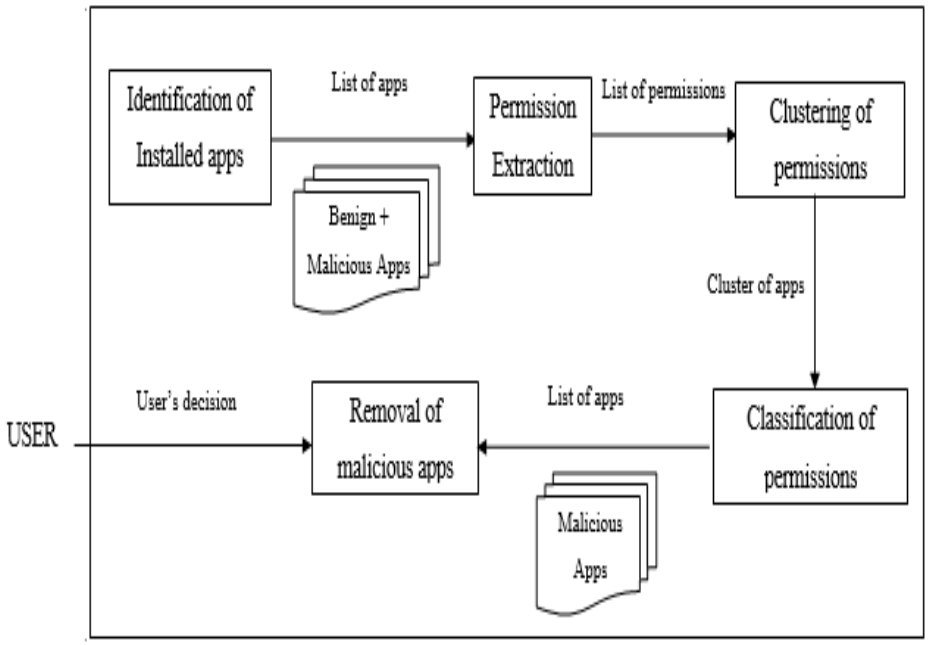

Fig2. *Proposed System for Android Application Analyzer*

An Android Application Analyser is a secure advisory system which helps the user in determining which apps are harmful and needs to be removed. This decision is based on the analysis of the permissions of installed applications and hence the burden of analysing and understanding the risks involved due to those harmful permissions on the user is completely removed. The proposed system identifies the harmful apps and also provides a user with an option to remove those apps completely from the phone. In this way, it aims to reduce administration burden on the user and thereby secures Android phone.

The Figure 2 describes the proposed system for Android Application Analyser which consists of one actor involved, the user and it takes input as the list of installed applications. The entire system consist of major five modules such as identification of installed applications, permission extraction, clustering of known permissions into categories, classification of benign and malicious apps and removal of malicious apps. The modules thus are explained in detail below.

6.1 Finding List of Installed Apps

This module identifies the list of installed apps on the phone programmatically using Android PackageManager API. Android PackageManager class is used for retrieving various kinds of information related to the application packages that are currently installed on the device. An instance of this class is called through get Package Manager ().

6.2 Permission Extraction

This module deals with simply extracting the permissions of installed applications on the phone. An input to this module is a list of installed apps. PackageInfo contains overall information about the contents of a package. This corresponds to all of the information collected from AndroidManifest.xml.

6.3 Clustering of Permissions

This module aims to create clusters of known harmful permissions. The number of clusters created is based on the known families of malicious apps having dangerous permissions. Each cluster will be having a combination of known harmful permissions. An input to this module will be a list of permissions of each app which was obtained from permission extraction module. The

permissions stored in each cluster will be compared with the permissions of installed app. If a match is found then an app is declared as malicious or else benign. In order to perform clustering, k-means algorithm is used. It is an iterative algorithm, organizing numerical data in k number of clusters. The numerical data, or training sets, are organized in vectors with a dimension equal to the number of features to be evaluated. K-means consists of following steps:

1. Select k centroids arbitrarily (called as seed) for each cluster Ci, i ε [1, k]
2. Assign each data point to the cluster whose centroid is closest to the data point.
3. Calculate the centroid Ci of cluster Ci, i ε [1, k].
4. Repeat steps 2 and 3 until no points change between clusters

The steps are repeated until the algorithm converges. Convergence is achieved when the second step no longer assigns any vectors to new cluster centroids ***[8].*** When running the k-means algorithm on the applications permission set, the different applications will be grouped together based on how similar they are in their permission patterns.

6.4 Classification of Permission

After clustering, matching set of known permissions are simply categorized into one or the other malicious clusters. However it also results in benign app being declared as malicious. Hence this module is designed with an aim to classify whether an app is really a benign or malicious one. It achieves this using Naïve Bayesian classification algorithm which classifies an app into malicious and benign one. Bayesian classifiers are statistical classifiers. For each new sample they provide a probability that the sample belongs to a class (for all classes). The Naive Bayes algorithm is based on conditional probabilities. It uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data. Bayes' theorem can be stated as follows ***[9]***:

Prob(B given A) = Prob(A and B)/Prob(A).

Naïve Bayesian Classification Algorithm ***[9]***:

Begin

For all classes ci € c = c1,…., cm

Compute P(ci);

For all features xj € x

Compute P(xj |ci);

End for;

Multiply all P(xj |ci)'s

Calculate fi(d) = P(ci)*P(xj|ci);

End for

Assign d to the class (es) of max (fi(d),….,fm(d))

End;

The other feature or attributes that could be of significant importance are, for example, list of activities, set of API's, etc. However, the other attribute is yet to be finalized.

6.5 Removal of Malicious Apps

This module simply presents a user with a list of malicious apps identified during classification stage. It takes user input decision to delete or wipe out harmful app from the phone. If user wishes to uninstall a particular harmful app suggested by analyzer then it simply removes it.

7. CONCLUSION

The objective of proposed system is to provide assistance to the user by identifying and removing those applications which can be harmful. However, removal of an application requires a user's

decision. It achieves an overall objective by using techniques such as clustering and classification. Clustering algorithm such as k means is used to group apps into similar known malicious permissions and then classifying those apps into malicious and benign class using naïve Bayesian classification algorithm.

Our proposed system is similar to the one discussed in [1] in a sense that it provides an assistance to the user by analyzing the harmful applications rather than analyzing the risk of application's permission, which reduces the burden of analyzing harmful apps on user. Secondly the approach of first clustering and then classifying has been discussed briefly in [2] and [5] respectively. We are proposing a model combining these two approaches to obtain better result.

REFERENCES

[1] Takayuki Matsudo, Eiichiro Kodama, Jiahong Wang, and Toyoo Takata., A Proposal of Security Advisory System at the Time of the Installation of Applications on Android OS, 15th IEEE International Conference on Network-Based Information Systems (NBiS), 2012.

[2] Yerima, S.Y.; Sezer, S.; McWilliams, G.; Muttik, I., A New Android Malware Detection Approach Using Bayesian Classification, Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on , vol., no., pp.121,128, 25-28 March 2013.

[3] Agematsu, H.; Kani, J.; Nasaka, K.; Kawabata, H.; Isohara, T.; Takemori, K.; Nishigaki, M., A Proposal to Realize the Provision of Secure Android Applications -- ADMS: An Application Development and Management System, Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on , vol., no., pp.677,682, 4-6 July 2012.

[4] Ghorbanian, M.; Shanmugam, B.; Narayansamy, G.; Idris, N.B., Signature-based hybrid Intrusion detection system (HIDS) for android devices, Business Engineering and Industrial Applications Colloquium (BEIAC), 2013 IEEE , vol., no., pp.827,831, 7-9 April 2013.

[5] Dong-Jie Wu; Ching-Hao Mao; Te-En Wei; Hahn-Ming Lee; Kuo-Ping Wu, DroidMat: Android Malware Detection through Manifest and API Calls Tracing, Information Security (Asia JCIS), 2012 Seventh Asia Joint Conference on , vol., no., pp.62,69, 9-10 Aug. 2012.

[6] Wei Tang; Guang Jin; Jiaming He; Xianliang Jiang, Extending Android Security Enforcement with a Security Distance Model, Internet Technology and Applications (iTAP), 2011 International Conference on , vol., no., pp.1,4, 16-18 Aug. 2011.

[7] Jesse Burns, Developing Secure Mobile Applications For Android ,iSEC partners, Online Available: https://www.isecpartners.com/media/11991/isec_securing_android_ apps .pdf

[8] Kardi Tech 's Page, "K – means Clustering Algorithm :Tutorial - How k-means clustering algorithm works?", Online Available : http://people.revoledu.com/kardi/ tutorial/kMean/Algorithm.htm

[9] Scribd. , "Classification algorithms used in Data Mining" , Online Available: http:// www.scribd.com/doc/11352378/Classification-algorithms-used-in-Data-Mining-This-is-a-lecture-given-to-Msc-students