International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 5 2587 - 2590

# Malware security for Android Components using Layer permission

Rohit Kale
Department of Computer Engineering
Jaywantrao Sawant College of Engineering, Hadapsar
Pune, India
*rohitkale8521@yahoo.com*

Prof. P D. Lambhate
Department of Information Technology,
Jaywantrao Sawant College of Engineering, Hadapsar
Pune, India
*lambhatepoonam9@gmail.com*

**Abstract**— Today's open source android smartphone operating system is adept of executing the multifaceted and enormous application, which increases the installation of diverse applications with increase in chance of installation of malware application. The behavior pattern of android is depicted by the requested permission of application. System explores a way to detect malware application based on requested permission by the application. Detection of malware application is done in two steps; first step is to selecting representative features by applying the FAST algorithm. Whereas representative feature is extracted permissions, requested in the application. In second step classification of application is done as a malware or benign application using support vector machine (SVM). Using FAST and SVM algorithms system can discriminate android application as malware also enrich the performance of malware detection system.

***Keywords***— *SVM- support vector machine, APK- Android application package.*

*****

## I. INTRODUCTION

Previously, user used to search the application on the Internet because there was not a centralized access to download the application. So installation was done by an authentication protocol that certified the application. Recently, the distribution of application is developed and user can able to install the application from Internet of the mobile. To enhance searching process of applications, first time the App Store of Apple developed online store for novel user. This was very successful concept, leading to other vendors such as RIM, Microsoft or Google to implement the same business model and developing application stores for their devices. This leads to develop large number of application for those platforms.

Google play store is the store for uploading and downloading the Android application for developer and end user respectively. Android application associates with the permission list required for accessing the special services of the device like GPS, Internet, SMS, etc. Developer uploads any kind of application and game. The Google does not do review of the applications. Instead, during the installation of the application on user device, it shows a pop up about required permission list for the application. Here user can cancel the installation of application if he doesn't want to grant permission to access the system resources that are requested by application. If user allows application to install then application doesn't ask to user during performing the operation.

Analysts predict that's mobile technology becomes more advanced[1] and handheld devices grow cheaper, the mobile industry will be dominated by advanced mobile hand-held devices by the year 2014.

Sales of applications for mobile devices are also expected to grow rapidly—annually at 73% for Smartphone's, and 93% for tablets during 2010-15. The revenue from paid mobile applications for Smartphone's and tablets is estimated to be $2.2 billion worldwide for 2010, with an expected Compound Annual Growth Rate of 82% through 2015 to $37.5 billion.

Newer and more advanced mobile applications are being designed for Smartphone's and tablets with the phenomenon expected to continue driving higher levels of innovation in the mobile industry.

As Smartphone's are used for business, transaction, education, etc., It is easy to connect the different kinds of network, terminal, without knowing to user. This leak out privacy of user information. This leads to malicious activity by the application, by hiding important data like login credential, payment information, etc without authorization of user. Smartphone doesn't have capacity of running detection mechanism like PC so new type of malware detection mechanism is required[2].

## II. REALATED EORK

Permission of the system[3] used by application is studied by many researchers from couple of years. All of them study how the permissions are used by various applications in android operating system. Barrera et al.[4] shows a method for the analysis of permission based security models in their research paper. They have studied the strengths and weaknesses of the model by analyzing the permission model. The Self-Organizing Map (SOM) algorithm proposed for checking the similarity between the application's permission. They have created 2 dimensional, discretized representations of high dimensional data. To create this they have assign the labels.

To analyze the android permission they have used 1,100 applications dataset also they have marked the top 50 applications in the Android marked from 22 categories.

Results of their various experiments show that permissions that are used very frequently have small subset where large subsets of permissions were used by very few applications. They suggested that the frequently used

permissions, specifically a.p.INTERNET, did not provide sufficient expressiveness and hence may benefit from being divided into subcategories, perhaps in a hierarchical. Conversely, In general category self-defined and the complementary permissions (e.g., install/uninstall) from the infrequent permissions are wrapped. Combining infrequent permissions and frequent permissions with finer granularity enhances the expressiveness of the permission model. This is done without increasing the complexity.

Detection of the malware in application is done using two methods: dynamic monitoring and static analysis. In Dynamic monitoring many times there is need of updating the application. This is done to monitor the application, which run in Dalvik Virtual Machine (DVM) or native environment. Crowdroid and Andromaly are monitoring the phone activity. After recording the activity of user it collect the important data.

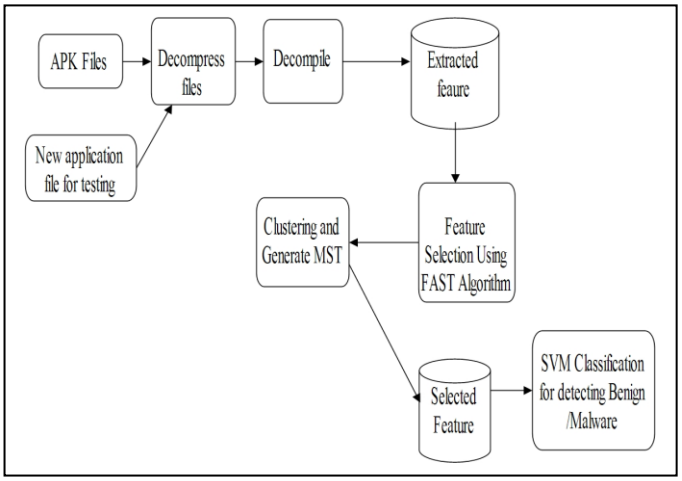
In Crowdroid, it collects the data from different users and creates the feature vector. Here the data is sent to remote server by using the network connection. Whenever the system call happen it gets monitored and this becomes the data collection. From all of the user the data sent to server and at the server side the all user data is stored. At server side, it uses the k-means algorithm to cluster the data on the collected data. Clustering portray the application as malicious application. Here the user privacy data leakage problem will occur during the process of implementation because this process needs users participation, and needs to collect the user's behavior data when they use the application data [5].

Andromaly detects malware by monitoring multiple things like Smartphone and the user's activities, recording the sensor activity, CPU usage rate and so on. However, as there is not only a process of monitoring mobile phones but also the information transmission may cost additional resources and traffic, which results in lack of providing detection to mobile phone in time, with high time and resource consumption[6]. In Taintdroid to identify any privacy leakage in Android application it uses dynamic taint tracking, similar privacy leakage detection system [7].

PiOS is designed for Apple iOS). Enck et al. also uses dynamic taint analysis technologies to analyze the situation and monitors the phones sensitive data access, but they did not put forward specific malicious code detection scheme [8].

Other research focused on identify malware by using machine learning techniques. Sanz et al.[9] applied number of types of classifiers to the static string, ratings, as well as permissions of around 820 apps to predict application categories. They also presented PUMA, which uses the extracted permissions from the application itself[10], for detecting malicious Android applications through machine learning techniques by analyzing it. Shabtai et al.[6] used requested permissions to classify Android toos and Android games.

## III. PRAPOSE WORK




### *A. System Architecture*

As per above study, designed a framework system for detection of malware application for Android platform based on FAST clustering [11] and SVM classifier. The system architecture is shown in figure 1

Fig. 1. System Architecture

There are 4 modules design in system architecture diagram as follows,

1. Preprocessing (Decompression, decompiling, Feature Extraction)

2. Feature Selection (FAST Algorithm)

3. Feature Classification (Support Vector Machine)

4. Malware detection

### *B. Preprocessing*

Preprocessing require to decompress each Android application package file. After decompressing get AndroidManifest.xml file from the extracted content, and then to get permission, decompile the xml file. Finally get each APKs permission list from its decompiled AndroidManifest.xml. All these permission vectors form the original feature set.

The primary mission is to extract the entire feature from the samples. Firstly we need to unzip the file to get the APK (Android application package) from he zipped file and then it decompile of APK done. AndroidManifest.xml file for an android application is a resource file which contains all the details needed by the android system about the application. This xml file contains activity, services, permissions, package name, minimum SDK support, etc. After decompiling XML file, permissions of the APK can be in readable form. The figure below is permissions list in AndroidManifest.xml.

```
</uses-permission>
    <uses-permission
        android:name="android.permission.WAKE_LOCK"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.VIBRATE"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.ACCESS_WIFI_STATE"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.CHANGE_WIFI_STATE"
        >
    </uses-permission>
    <uses-permission
        android:name="android.permission.INTERNET"
```

Fig. 2. Permission List

### C. Feature selection

In FAST algorithm, by using the graph clustering method the feature of the applications are clustered. A subset of feature is created from the cluster. The subset is the most related representative feature that is strongly related to the that class. There are different clusters get formed and those are relatively independent feature. Clustering based FAST algorithm produces a independent useful feature with a high probability. To check the highly efficient of FAST, it uses efficient minimum spanning tree clustering method. To produce the feature by FAST it uses different steps:

1. The irrelevant features are removed.
2. Then next step is to create Minimum Spanning Tree from the relative once
3. Then selection of the representative feature is done by partition the Minimum Spanning Tree.

Here are the mathematical steps to follow above steps

1. Input: D (F1,F2,...Fm,C)- the given set
2. $\theta$-the T-Relevance threshold
3. Output: S- selected feature subset

=====Part1: irrelevent Feature Removal=======

4. for i=1 to m do
5. T-Relevance=SU(Fi,C)
6. $S=S \cup \{Fi\}$

==Part2: minimum Spanning Tree Construction===

7. G= NULL;// G is a complete graph
8. for each pair of feature {Fi', Fj'} $\rightarrow$ S do
9. F-Correlation=SU(Fi', Fj')
10. Add Fi'and Fj' to G with F-Correlation as the weight of the corresponding edge;
11. minSpanTree= Prim(G);//Using Prim Algorithm to generate the minimum spanning tree

==Part 3:Tree Partition and representative Feature Selection ===

12. Forest= minSpanTree
13. For each edge Eij $\in$ Forest do
14. If SU(Fi',Fj' $\leq$ SU(Fi',C) $\wedge$ SU(Fi',Fj' $\leq$ SU(Fi',C))
15. then
16. Forest=Forest - Eij
17. S = $\emptyset$
18. For each tree Ti $\in$ Forest do
19. FRj = argmaxFk' $\in$ TiSU(Fi',C))
20. S=S U {FRj}
21. return S

### D. Feature Classification

*The SVM classifier* —Using trained set entails feature vectors of malware samples and benign software samples with classifier, construct trained dataset. When the new APK comes, we can use the trained classifier to classify the features vector of new APK according to feature values defined by classifier.

*Feature Dataset*- This module is responsible for storing and updating features extracted from samples.

*E. Mathemetical model*

let S={P,FS,C,TD,I}

I=input APK zip format file

P is pre processing ={Dc, Decompile, FE}

DC= De compress file & get xml files {f1, f2, f3....}

De compile=for readable xml {x1, x2, x3....}

FE=extract feature i.e permission from xml file {P1, P2, P3....} pass feature set to FAST algorithm

FS={C1,F,SV,F',G,E,V, $\theta$,MST}

G=(V,E)

C1=clusters

$\theta$ = threshold value

V={(Fi', Fj')—Fi'$\epsilon$Fi $\Lambda$ i$\epsilon$ [i, k]}

E = {(Fi', Fj' )—(Fi', Fj'$\epsilon$Fi'$\Lambda$ i, j$\epsilon$ [i, k] $\Lambda$ i $\neq$j) }

SU(x,y) = $\frac{2XGain(X|Y)}{H(X)+H(Y)}$

H(X)$\rightarrow$ Entropy = $-\sum_{x\epsilon X} p(xlog_2p(x))$

Gain (X|Y)= H(X) H(X|Y ) = H(Y ) H(Y |X)

C=Classify Algorithm SVM

TD is trained dataset Evaluation

## IV. Evaluation

For this proposed method use two types of database. Below discuss the statistical characteristic of the requested permissions.

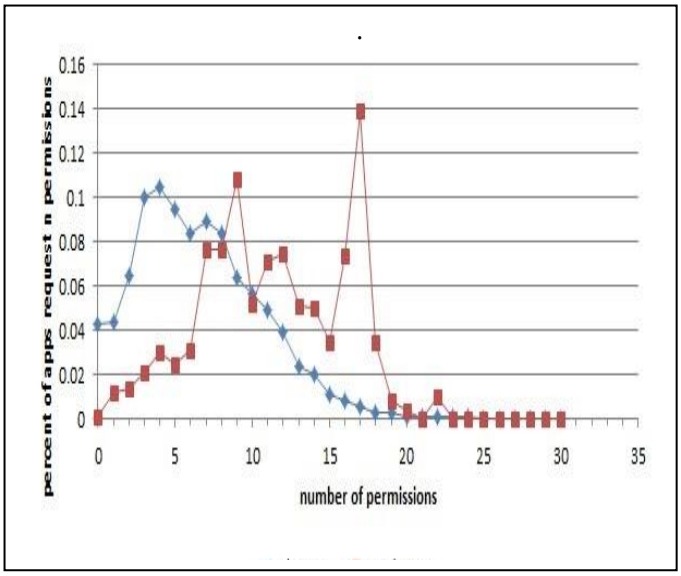
*A. Benign Dataset*

I have collected benign applications from Google Play store and China's app[12] store i.e. AppChina. I have downloaded applications from different kind of categories available on those stores. As there are large number of application present on those store, it may contain the malware applications also, but as we have downloaded the famous applications only. So our dataset contains mostly the benign applications.

*B. Malware Dataset*

I have used malware samples from Virusshare site. I have downloaded 24,317 samples, which was uploaded on 2014-march-24. From this some of the APK are used for the training dataset and remaining for detection phase. Also some of the applications are downloaded from various sites. And depending on the behavior of the application it is divided into subcategory. Some of them called as malware if the primary function is to download the separate payload. If malware application stole data from android device then the android application is classified as stealing of credentials. Also some application classified as sent the SMS message.

As per dataset discussion [13] in figure 3, the Android




benign applications require less permission than. As per analysis more than 12 permissions are required for more than 40% of malware application, where as less than 6 permissions are required for about 35.6% of benign application *Percentage of apps requesting certain number of permissions*

## V. Expected Experimental Result

Algorithm will differentiate application as malware or Benign. As FAST algorithm select less but correct feature for identify malware or benign, it reduces time for svm classification. So this algorithm will reduces time complexity with best result. And to find the efficiency I will use following metrics:

*True Positive Rate (TPR)*: Percentage of correctly identified benign applications

(TP / TP+FN)

*False Positive Rate (FPR)*: Percentage of wrongly identified malware applications

(FP / TN+FP)

*Overall Accuracy (ACC)*: Percentage of correctly identified applications

(TP+TN / TP+TN+FP+FN)

## VI. Conclusion

Features in different clusters are relatively independent; the clustering based strategy of FAST has a high probability of producing a subset of useful and independent features. FAST algorithm removes the redundant permissions also the irrelevant permisssion from the feature set. This will improve the power of detecting malware in android application. The SVM classifier, classify the relative permisstion as a malware or benign application. Using FAST, the efficiency of the SVM will increase.

## ACKLODGEMENT

It is a pleasure for authors to thank all the people who in different ways have supported us in completing this study and contributed to the process of writing this paper.

## REFERENCES

[1] Botha, R.A., Furnell, S.M., Clarke, N.L.: "Fromdesktop to mobile: Examining the security experience". Computer & Security 28, 130137 (2009).

[2] Zhao Xiaoyan*,Fang Juan and Wang Xiujuan, "ANDROID MALWARE DETECTION BASED ON PER- MISSIONS", IEEE, 2011

[3] A. P. Felt, K. Greenwood and D. Wagner, "The effectiveness of application permissions, Proc. 2nd USENIX conference on Web application development", USENIX Association, 2011, pp.7-7.

[4] D. Barrera, H. G. Kayacik, P. C. van Oorschot, A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android", Proc. 17 ACM conference on Computer and communications security, ACM, 2010, pp.73-84. th

[5] BURGUERA I, ZURUTUZA U, NADJM-TEHRANI S. "Crowdroid: behavior-based malware detection system for Android"[C]//Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices. New York, USA: ACM, 2011: 15-26.

[6] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer, Y. Weiss: Andromaly: "a behavioral malware detection framework for android devices". Journal of Intelligent Information Systems 38(1) (January 2011) 161-190.

[7] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. TaintDroid: "An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones". In Proc. of USENIX OSDI, 2010.

[8] M. Egele, C. Kruegel, E. Kirda, and G. Vigna. PiOS: "Detecting Privacy Leaks in iOS Applications". In Proc. of NDSS, 2011.

[9] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas and G. Alvarez, PUMA: "Permission Usage to Detect Malware in Android", International Joint Conference CISIS12-ICEUTE 12-SOCO 12 Special Sessions, Springer Berlin Heidelberg, 2013, pp.289-298.

[11] Qinbao Song, Jingjie Ni, and Guangtao Wang, "A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data" IEEE Transaction on knowledge and Data Engineering Vol. 25, 2013.

[12] S. Ye. Android Market is Currently Blocked in China. Here are your Alternatives, Sep 2011. http://techrice.com/2011/10/09/android-market-iscurrentlyblocked-in-china-here-are-your-alternatives.

[13] Xing Liu, Jiqiang Liu, "A Two-layered Permission-based Android Malware Detection Scheme", 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering.