International Journal of Emerging Technology and Computer Science
ISSN: 2455-9954

Android Malware Detection Using Deep Eigenspace Learning

Mr. Rahul Pawar, Akanksha Khedkar, Golu Kumar, Apurva Dhumal
apurva122dhumal@gmail.com, golu7254827920@gmail.com, akankshakhedkar27@gmail.com
Department of Computer Engineering,
Dr. D. Y. Patil College Of Engineering, Akurdi, Pune, India.

ABSTRACT

Today's open source android smartphone package is adept of executing the multifarious and enormous application that will increase the installation of various applications with increase in probability of installation of malware application. The behaviour pattern of android is represented by the requested permission of application. System explores the simplest way to discover malware application supported requested permission by the application. Detection of malware application is completed in 2 steps; start is to choosing representative features by applying the FAST algorithm. Whereas representative feature is extracted permissions, requested within the application. In second step classification of application is completed as a malware or benign application victimisation support vector machine (SVM). Using fast and SVM algorithms system will discriminate android application as malware conjointly enrich the performance of malware detection system.

Keywords - SVM- support vector machine, APK- android application package;

I. INTRODUCTION

Previously, user wont to search the application on the web as a result of there wasn't a centralized access to transfer the application. Thus installation was done by an authentication protocol that certified the appliance. Recently, the distribution of application is developed and user will able to install the appliance from net of the mobile. To enhance looking out method of applications, 1st time the App Store of Apple developed on-line store for novel user. This was terribly successful idea, resulting in alternative vendors like RIM, Microsoft or Google to implement a similar business model and developing application stores for his or her devices. This ends up in develop sizable amount of application for those platforms.

Google play store is that the store for uploading and downloading the android application for developer and user severally. Android application associates with the permission list needed for accessing the special services of the device like GPS, Internet, SMS, etc. Developer uploads any reasonably application and game. The Google doesn't do review of the applications. Instead, throughout the installation of the appliance on user device, it shows a pop regarding needed permission list for the appliance. Here user will cancel the installation of application if he doesn't wish to grant permission to access the system resources that are requested by application. If user permits application to put in then application doesn't raise to user throughout activity the operation.

Analysts predict that's mobile technology becomes a lot of advanced and hand-held devices grow cheaper, the mobile trade are dominated by advanced mobile hand-held devices computer memory unit year 2014.

Sales of applications for mobile devices are expected to grow rapidly—annually at seventy three for Smartphone's, and 93 for tablets throughout 2010-15. The revenue from paid mobile applications for Smartphone's and tablets is calculable to be $2.2 billion worldwide for two010, with associate degree expected Compound Annual rate of eighty two through 2015 to $37.5 billion.

Newer and a lot of advanced mobile applications are being designed for Smartphone's and tablets with the development expected to continue driving higher levels of innovation within the mobile trade.

As Smartphone's are used for business, dealing, education, etc., it's simple to attach the various forms of network, terminal, while not knowing to user. This get out privacy of user info. This ends up in malicious activity by the appliance, by concealment necessary knowledge like login papers, payment info, etc. while not authorization of user. Smartphone doesn't have capability of running detection mechanism like laptop thus new sort of malware detection mechanism is needed.

II. REALATED EORK

Permission of the system utilized by application is studied by several researchers from number of years. All of them study however the permissions are utilized by numerous applications in android package. Barrera et al. [5] shows a way for the analysis of permission based mostly security models in their analysis paper. They need studied the strengths and weaknesses of the model by analysing the permission model. The Self-Organizing Map (SOM) algorithm planned for checking the similarity between the application's permission. They need created two dimensional, discretized representations of high dimensional knowledge. To make this they need assign the labels.

To analyze the android permission they have used 1,100 applications dataset conjointly they need marked the highest 50 applications within the android marked from 22 classes.

Results of their numerous experiments show that permissions that are used very frequently have little set wherever large subsets of permissions were utilized by only a few applications. They urged that the often used permissions, specifically a.p.INTERNET, didn't give sufficient quality and therefore might benefit from being divided into subcategories, maybe in a hierarchical. Conversely, normally category self-defined and the complementary permissions (e.g., install/uninstall) from the rare permissions are wrapped. Combining rare permissions and frequent permissions with finer roughness enhances the quality of the permission model. This is done without increasing the complexness.

Detection of the malware in application is completed using 2 methods: dynamic observance and static analysis. In Dynamic observance repeatedly there's want of change the appliance. This can be done to monitor the application that run in Dalvik Virtual Machine (DVM) or native environment. Crowdroid and Andromaly are observance the phone activity. Once recording the activity of user it collect the vital information.

In Crowdroid, it collects the information from totally different users and creates the feature vector. Here the information is shipped to remote server by victimization the network affiliation. Whenever the call happen it gets monitored and this becomes the information assortment. From all of the user information sent to server and at the server facet the all user data is hold on. At server facet, it uses the k-means algorithmic program to cluster {the information the info the information} on the collected data. Bunch portray the appliance as malicious application. Here the user privacy information escape downside can occur throughout method the method} of implementation as a result of this process wants users participation, and desires to gather the user's behaviour information after they use the appliance information [1].

Andromaly detects malware by observance multiple things like Smartphone and also the user's activities, recording the device activity, cpu usage rate and then on.

However, as there's not solely a method of observance mobile phones however conjointly the knowledge transmission might value extra resources and traffic, which ends up in lack of providing detection to itinerant in time, with time and resource consumption[2]. In Taintdroid to spot any privacy escape in android application it uses dynamic taint trailing, similar privacy escape detection system [4].

PiOS is meant for Apple iOS). Enck et al. conjointly uses dynamic taint analysis technologies to research true and monitors the phones sensitive information access, however they didn't proposes specific malicious code detection theme [3].

Other analysis targeted on establish malware by using machine learning techniques. Sanz et al. [7] applied range of sorts of classifiers to the static string, ratings, likewise as permissions of around 820 apps to predict application classes. They also presented puma that uses the extracted permissions from the appliance itself, for police work malicious android applications through machine learning techniques by analyzing it. Shabtai et al. [2] 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 supported fast clustering and SVM classifier. The system design is shown in figure one

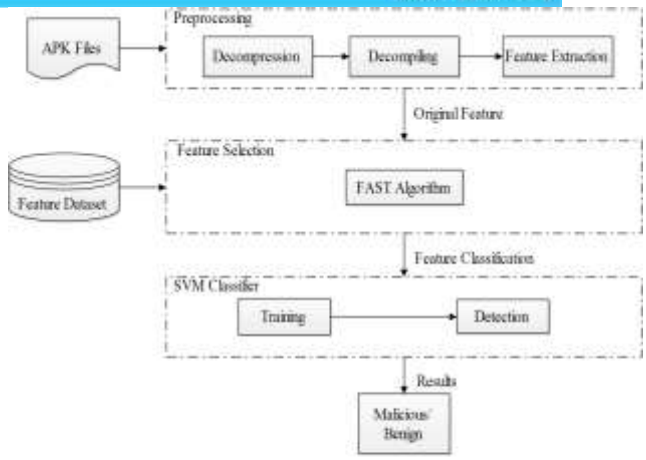


Fig. 1. System design

There are four modules style in system design diagram as follows,

1. Preprocessing (Decompression, decompiling, Feature Extraction)

2. Feature choice (FAST Algorithm)

3. Feature Classification (Support Vector Machine)

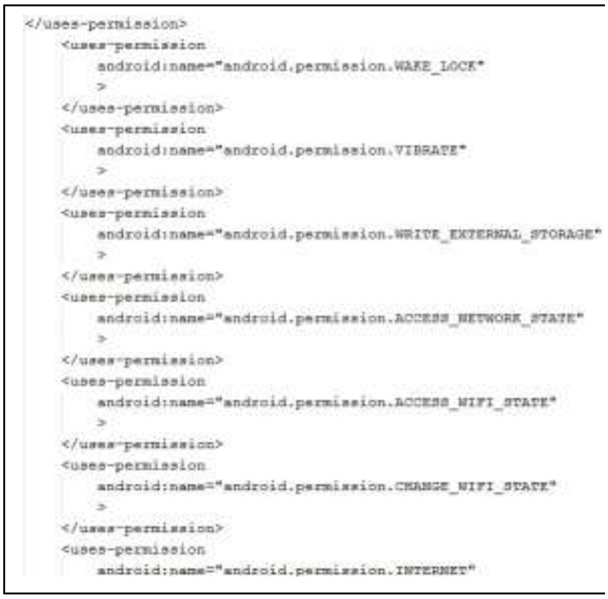
4. Malware detection

B. Preprocessing

Preprocessing need to decompress every golem application package file. When decompression get AndroidManifest.xml file from the extracted content, then to urge permission, decompile the xml file. Finally get every APKs permission list from its decompiled AndroidManifest.xml. Of these permission vectors kind the initial feature set.

The first mission is to extract the whole feature from the samples. First we'd like to unzip the file to urge the APK (Android application package) from he zipped file then it decompile of APK done. AndroidManifest.xml file for an android application may be a resource file that contains all the main points required by the android system concerning the application. This xml file contains activity, services, permissions, package name, minimum SDK support, etc. when decompiling XML file, permissions of the APK will be in legible kind.

The figure below is permissions list in AndroidManifest.x



```
</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. 1. Permission List

A. *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 class. There are different clusters get formed and those are relatively independent feature. Clustering based FAST algorithm produces an 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

```
Input: D (F1,F2,...Fm,C)- the given set
$\theta$-the T-Relevance threshold
Output: S- selected feature subset
=====Part1: irrelevent Feature Removal=======
for i=1 to m do
T-Relevance=SU(Fi,C)
S=S ∪ {Fi}
==Part2: minimum Spanning Tree Construction===
G= NULL;// G is a complete graph
for each pair of feature {Fi‘, Fj‘} ⇢ S do
F-Correlation=SU(Fi‘, Fj‘)
Add Fi‘and Fj‘ to G with F-Correlation as the weight of the corresponding edge;
minSpanTree= Prim(G);//Using Prim Algorithm to generate the minimum spanning tree
==Part 3: Tree Partition and representative Feature Selection ===
Forest= minSpanTree
For each edge Eijϵ Forest do
If SU(Fi‘,Fj‘ ≤ SU(Fi‘,C) Λ SU(Fi‘,Fj‘ ≤ SU(Fi‘,C))
then
Forest=Forest - Eij
S = ∅
For each tree Tiϵ Forest do
FRj = argmaxFk‘ϵTiSU(Fi‘,C))
S=S U {FRj}
return S
```

B. *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.

C. Mathematical model

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

I=input APK zip format file

P is preprocessing = {Dc, Decompile, and 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, θ,MST}

G= (V, E)

C1=clusters

θ = threshold value

V= {(Fi', Fj')—Fi'ϵFi Λ iϵ [i, k]}

E = {(Fi', Fj')—(Fi', Fj'ϵFi'Λ i, jϵ [i, k] Λ i ≠)}

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

H(X)→ Entropy =$-\sum_{x\epsilon X} p(X)\log_2 p(X)$

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

C=Classify Algorithm SVM

TD is trained dataset Evaluation

For this proposed methodology use 2 styles of info. 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 store i.e. AppChina. I have downloaded applications from totally different kind of classes obtainable on those stores. As there are sizable amount of application present on those store, it's going to contain the malware applications conjointly, however as we've got downloaded the known applications solely. thus our dataset contains largely the benign applications.

B. Malware Dataset

I have used malware samples from Virusshare website. I actually have downloaded 24,317 samples, that was uploaded on 2014-march-24. From this a number of the APK are used for the coaching dataset and remaining for detection section. conjointly a number of the applications are downloaded from varied sites. And reckoning on the behavior of the application it's divided into subcategory. a number of them known as as malware if the first perform is to transfer the separate payload. If malware application stole data from android device then the android application is assessed as stealing of credentials. conjointly some application classified as sent the SMS message.

IV. CONCLUSION

Features in numerous clusters are comparatively independent; the clustering based strategy of fast features a high likelihood of producing a set of useful and independent options. fast algorithm uses minimum spanning tree based methodology to cluster features. Meanwhile, it doesn't assume that knowledge points are sorted around centers or separated by a daily geometric curve. what is more and quick doesn't limit to some specific types of data. As quick algorithm has local time it's offer potency in terms of your time. it'll offer relevant feature for malware detection in android application. SVM is extremely effective method used for classification of feature.

REFERENCES

[1] 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.

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

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

[4] 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.

[5] 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

[6] 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.

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

[8] 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

[9] 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

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

[11] 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

[12] Zhao Xiaoyan*, Fang Juan and Wang Xiujuan, “ANDROID MALWARE DETECTION BASED ON PER- MISSIONS”, IEEE, 2014

[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