International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 4, July-Aug-2020, ISSN (Online): 2395-566X

Android Malware Detection by Using Random Forest Algorithm

Prof. Yogesh Pawar, Yash Gudhka, Manish Sutar, Pavan Godhani
Department of Information Technology Engineering
D Y Patil Institute of Engineering & Technology, Ambi Pune, India.

***Abstract*-Android is one of the biggest Operating System platforms in the world. By the end of 2019, Android has much advancement in its Operating System. Android is widely used due to its openness and the backup provided by Google. Google has launched its Google Playstore where Android users find applications for every need. Since Android is open to applications from other sources apart from Google Playstore, attackers find it easy to insert malicious code inside the Android application package file which can further harm the users. Hence to protect users from malicious applications Malware detectors are developed to find the malicious code within the APK file.**

***Keywords*-Machine Learning, Random Forest Algorithm, APK (Android Application Package), androidmanifest.xml.**

I.INTRODUCTION

Android is the biggest market in the field of technology. Most of the devices like smart phones, tablets, etc run on Android O.S. Android are open source system i.e. it can download applications from any source. This can make him vulnerable to malicious attacks and may hamper the user's device. In Android some applications are restricted due to certain reasons like unsupported version, unsupported region/country, unsupported device requirements, etc so users tend to download it from unverified sources like unsecured websites, links, etc. Most of the time the applications source code is changed or a malicious code is inserted in it to attack the user's device. This also jeopardise the company's reputation whose application has been affected. To prevent this situation every application must be verified and validated. We need to dug deep into application's permissions to find if there's a change in it, because as the user allows or grants the permission the malicious application can harm the user's device and data.

An android application works according to the permissions given to him by the user. On installing the application, it requests certain permissions to use device components, alter data and use network. As the user grants him permission it can perform actions on the particular component. The attackers use these permissions to attack the user. They alter the permissions according to their personal use and when the user grants these permissions their attack begins.

These permissions include access to device network, camera, microphone, data, as well as access to other applications too. If these permissions are misused, it will hamper user's privacy and integrity. Hence to protect these, malware detectors are developed and installed in antivirus systems so to detect whether the application is good or bad.

II. LITERATURE REVIEW

A lot of Malware detectors have been developed before but most of them lacked proper algorithm, accuracy, etc. A proper malware detector must have a good algorithm as well as good space and time complexity. Old malware detectors need to be updated of the upcoming malwares so that their efficiency will be maintained. Following are some papers published on malware detection-

Machine Learning for Android Malware Detection Using Permission and API Calls (2013):

In this paper Naser Peiravian and Xingquan Zhu developed an Android application to detect malware in the .APK file by extracting its permissions and API calls and then further using them as features. These features are then run through a classifier which further detects them as good or bad permission or API call. Since they have developed an Android application there's a chance of the malware getting activated before installing the application and hampering the device. Hence a desktop application is preferred to check whether the application is malware or not.

Android Malware Detection using Genetic Algorithm (2019):

Here SVM (Support Vector Machine) algorithm was used. SVM provided an accuracy of 96% in detection of malware application. They were successful in creating an algorithm but it was complex as it contained neural network along for feature selection process.

III. OVERALL FRAMEWORK OF THE SYSTEM

A desktop GUI has been developed which contains the algorithms and functions within it to ease the process. The APK file is first downloaded in the particular computer.

1. APK Unzipping:
The APK file downloaded is unzipped using some commands and the files are traversed. Among the files, an "androidmanifest.xml" file is extracted. That file contains permissions for the android application written in encoded format. To access or alter these permissions the attacker needs to decode it first.

2. Extracting Permissions:
After decoding the xml file, permissions are extracted from the file. The permissions are used as a set of features. Some of the permissions are android. permission. SEND_SMS,android. permission. WRITE_CONTACTS, android. hardware. camera, etc. These permissions can be altered by the attacker easily and user doesn't know about this. Hence, they need to be verified before installing the application.

3. Applying Random Forest Algorithm:
The feature set is extracted & classified using the random forest algorithm. The permissions are categorised into its specific categories or domains i.e. hardware, network, calling, SMS, etc. A decision tree is created which contains the permissions as the leaf nodes of the tree.

4. Matching with Data-Set:
The leaf nodes are then matched with our training dataset and checked whether the permission is altered or not. If the permission is harmful then it will not consider it. If all the permissions are correct then maximum efficiency of the algorithm is achieved.

IV. EXPERIMENTAL RESULTS

The Experiment result is calculated using Machine Learning formulas. The performance can be evaluated by True Positive Rate (TPR), False Positive Rate (FPR) & Accuracy. They are given by:

TPR= TP/ (TP+FN),
FPR=FP/ (FP+TN),
Accuracy= (TP+TN)/ (TP+FP+TN+FN)

TP defines the number of benign applications identified correctly; TN defines the number of malware applications indentified correctly; FP defines the number of malware applications identified wrongly; FN defines the number of benign applications identifies wrongly. An accuracy of 92.20% is achieved by using Random Forest algorithm.

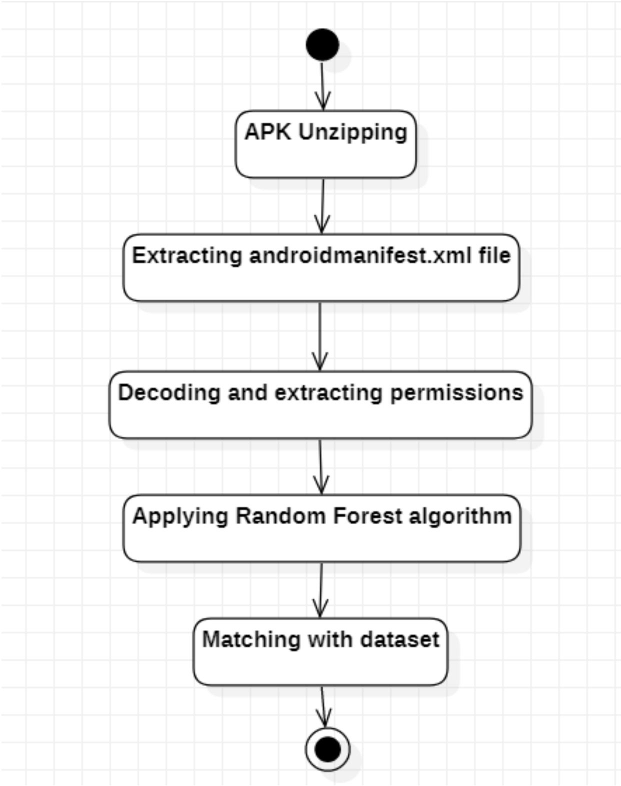
APK Unzipping
Extracting androidmanifest.xml file
Decoding and extracting permissions
Applying Random Forest algorithm
Matching with dataset

Fig 1 State-chart of malware detection.

VI. CONCLUSION

We can conclude that applications can be altered easily by changing their permissions. Hence we need to test any android .APK file before installing it. This application will test every .APK file and give results on the basis of the algorithm.

REFERENCES

[1] Xiang Li, Jianyi Liu, YanyuHuo, Ru Zhang, Yuangang Yao "An Android malware detection method based on AndroidManifest file" 2016 4th International Conference on Cloud Computing and Intelligence Systems , 19 Aug 2016

[2] Hyo-Sik Ham, Mi-Jung Choi "Analysis of Android malware detection performance using machine learning classifiers" 2013 International Conference on ICT Convergence, 16 Oct. 2013

[3] Patrick P. K. Chan, Wen-Kai Song "Static Detection Of Android Malware by Using Permissions and Api Calls" 2014 International Conference on Machine Learning and Cybernetics , 16 July 2014.

[4] Fengjun Shang, Yalin Li, Xiaolin Deng &Dexiang He "Android malware detection method based on naive Bayes and permission correlation algorithm" 17th June 2017.

[5] Dong-Jie Wu, Ching-Hao Mao, Te-En Wei, Hahn-Ming Lee "DroidMat: Android Malware Detection through Manifest and API Calls Tracing", 2012

Seventh Asia Joint Conference on Information Security.

[6] Jin Li, Lichao Sun, Qiben Yan “Significant Permission Identification for Machine-Learning-Based Android Malware Detection”, IEEE Transactions on Industrial Informatics (Volume: 14, Issue: 7, July 2018).

[7] Zarni Aung, Win Zaw “Permission-Based Android Malware Detection” INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 2, ISSUE 3, MARCH 2013

[8] Justin Sahs, Latifur Khan “A Machine Learning Approach to Android Malware Detection”, 2012 European Intelligence and Security Informatics Conference

[9] Naser Peiravian, Xingquan Zhu “Machine Learning for Android Malware Detection Using Permission and API Calls”, 2013 IEEE 25th International Conference on Tools with Artificial Intelligence.

[10] Brandon Amos, Hamilton Turner, Jules White, “Applying machine learning classifiers to dynamic Android malware detection at scale”, 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC).