RESEARCH ON DATA MINING OF PERMISSION-INDUCED RISK FOR ANDROID DEVICES

K. Madhubala[1], R. Shanmuga priya[2], Mr. B. Sathishkumar[3]

[1,2]UG Scholar, Computer Science and Engineering, Chettinad College of Engineering & Technology, Tamilnadu, India
[3]Assistant professor of Computer Science and Engineering, Chettinad College of Engineering & Technology, Tamilnadu, India

---***---

Abstract - *in a recent study malicious application can be developed in every 10 seconds. In our project have to develop a malware free application to detect the malware application in our android phone. For detecting the malware application Permission ranking, similarity-based Permission feature selection and association rule for permission mining techniques are used in data mining concept. In these techniques we use permission ranking to detect the malware application. Based on the permission ranking we have to find the malware in our phone. In this concept we track the other application in our phone. Using that we have to find which permissions are given to that application. We have to ranking the permission, with the help of that ranking we have to find the malwares. After finding the malware application we use random forest algorithm. This algorithm will be used to increase the accuracy of the malware. Finally the malicious application will be detected. If we identify that the application will be malicious then we have to notify to the user about that application. And also detect which files are accessed by that application will be notify to the user. If the user wants to block that application then the application will be blocked.*

Key Words: **malware free application, permission ranking, find malware, random forest, increase accuracy, etc.**

1. INTRODUCTION

Nowadays smart phones are used everywhere. That smart phones contain many malicious application. Because of this malicious application our phone will be affect. Then the application will be detected whether it is malicious application or not to protect the phone. For that there is a many application like antivirus, malware byte security. These applications are developed by many malware detection technique. The techniques are anomaly-based detection and signature based detection. These technique have many types like static, dynamic and hybrid .Anomaly based detection usually occurs in two phases that are training phase and detection phase. During the training phase the detector will learning the normal behavior. During the detection phase the detector will monitor the behavior of the host. Based on these two phase the detector will detect the malware application. The Signature-based detection attempts to model the malicious behavior of malware and uses this model in the detection of malware. The accuracy of these applications will be low. Its need to develop a more accurate application to detect the malicious application. In this system we use permission ranking algorithm to detect the malicious application, and the random forest algorithm will be used to increase the accuracy of the malware application.

1.1 LITERATURE REVIEW

Smart phones contain many malware applications. Because of this it is difficult to separate the malware and benign application. For identifying the malware application there was many malware detection algorithms are used, that are anomaly based detection algorithm and signature based detection algorithm. In anomaly based detection algorithm will contains two phases that are training phase and detection phase. In training phase the detector will learning the normal behavior of the application. And the detection phase the detector will monitor the behavior of the host. Using those two phases the malware application will be detected.

2. EXISTING SYSTEM

For detecting the malware app, we have many android Application like Malware bytes Security: Virus Cleaner, Anti-Malware. Enable new security settings in android system to detect malware apps. This study involves the malware detection process for the Android platform. There are known and unknown malware and benign apps in the market. Known malware is removed from the market place. Unknown malware evades the detection engine by hiding its malicious activities. We identify these unknown malwares and the benign apps based on binary classification. If there are a finite set of classes, then the classifier will determine the class of a given object. Binary classification involves only two sets of possible classes. In our stated problem, we must classify apps into two finite classes: malware or benign ware. Thus, we can represent this problem as a binary classification problem. We can consider all apps as a set of applications. From the literature review, we have not located any approach to predict the behaviour of malware from past actions. Only some researchers used data bases to store app behaviours. However, they did not mention any approach to predict future behaviours from past actions.

3. PROPOSED SYSTEM

Firstly, we adopted three data mining techniques Permission Ranking, similarity-based Permission feature selection and association rule for permission mining .These techniques are used to detect the malware application. The random forest algorithm is used to improve the accuracy for malware detection. The accuracy of random forest was improved by modifying selective parameters of the algorithm, iteratively Removing the unnecessary features, by setting the upper limit on the number of trees in the random forest. Feature selection is an important task in malware detection. There are many features, but we need to select those among them which will provide better accuracy in the classification process. Widely used feature selection methods for Mobile Malware detection.

4. METHODOLOGY

A. STATIC ANALYSIS

In the following section, we will discuss the static analysis of mobile malware detection. We mention the name of the research work, year, type, a short description, data set, detection method, major outcome, and limitations/future work. Single Category features: The advantages of single category features are easy to extract, and low power computation. The limitations associated with this method are code obstruction, imitation attack and low precision. Multiple categories of Features: The advantages of multiple category features are easy to extract, and high accuracy. The limitations associated with this method are Mimicry attack, high computation, code obfuscation, and difficult to handle multiple features.

B. DYNAMIC ANALYSIS

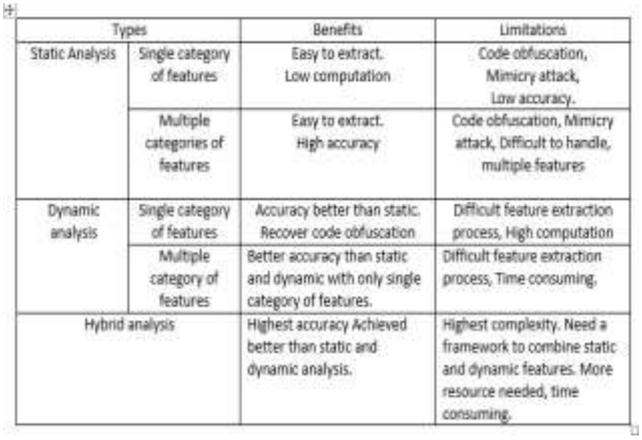
In this section, we discuss the dynamic analysis of mobile malware detection. Dynamic analysis requires execution of the malware. Dynamic analysis has been introduced to overcome the limitations of static analysis. Single Category features: it poses a better accuracy and it is easy to recover code obfuscation as compared with static analysis. However, its feature extraction process is difficult, and it consumes high resources. Multiple categories of Features: It gives better accuracy and it is easy to recover code obfuscations compared with a static and dynamic single category. The limitations of this approach are (1) difficult to handle multiple features; (2) high resources; and (3) more time needed for computation.

C. HYBRID ANALYSIS

In this section, we discuss hybrid analysis for mobile malware detection. Hybrid analysis involves both static and dynamic features. Researchers have introduced hybrid analysis to overcome the limitations of both static and dynamic analysis. Hybrid analysis helps to improve the accuracy level in the Android mobile malware detection process. The main benefits of hybrid analysis are to perform the highest accuracy as compared to static and dynamic analysis. The limitations are (1) highest complexity ;(2) framework requirement to combine the static and dynamic features; (3) more resource use; and (4) time-consumption.

DATASET:

To excavate practical significance, it can cover most Android permission models, which have their own characteristics. In this paper, our dataset composed of malware and benign application. The dataset includes 6192 benign and 5560 malware apps, collected from the Google Play Store and the Chinese App store.



Types		Benefits	Limitations
Static Analysis	Single category of features	Easy to extract. Low computation	Code obfuscation, Mimicry attack, Low accuracy.
	Multiple categories of features	Easy to extract. High accuracy	Code obfuscation, Mimicry attack, Difficult to handle, multiple features
Dynamic analysis	Single category of features	Accuracy better than static. Recover code obfuscation	Difficult feature extraction process, High computation
	Multiple category of features	Better accuracy than static and dynamic with only single category of features.	Difficult feature extraction process, Time consuming.
Hybrid analysis		Highest accuracy Achieved better than static and dynamic analysis.	Highest complexity. Need a framework to combine static and dynamic features. More resource needed, time consuming.

Fig 1: Comparison of Static, Dynamic and Hybrid

5. MODULE 1: MALWARE DETECTION

Malware detection is a process that contains different components to evaluate whether an application is malware or not. Generally, there are four components of the malware detection system namely, data set, feature extractor, feature selector, and classifier. Figure 2 illustrates a malware detection system.

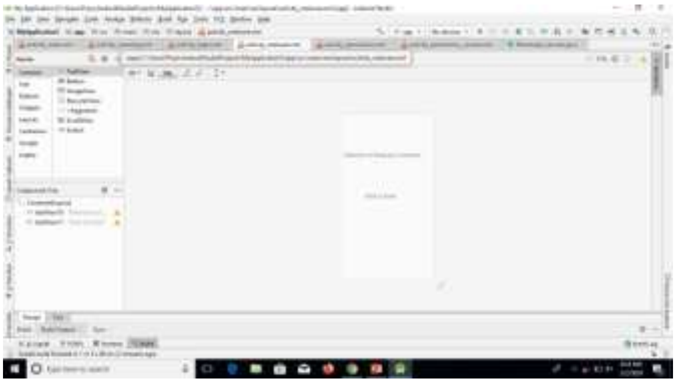


Fig 2: Malware Detection

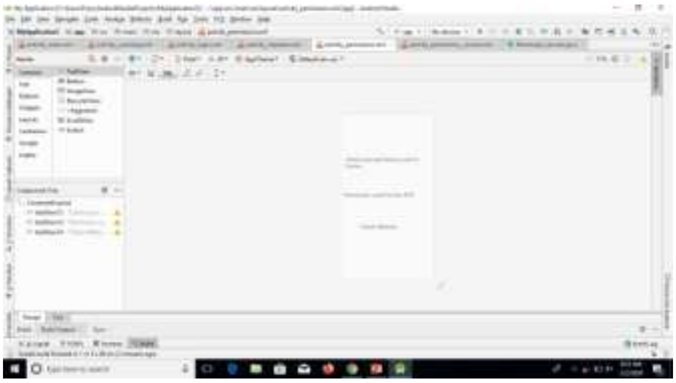


Fig 3: Check Malware

6. MODULE 2: MALWARE ACCURACY

So as to execute the Proposed System, we will specify graphs that cause us to comprehend the structure of the proposed framework.

User login: User login to the framework using ID and Password.

User registration: If a User is a new client, the Framework ask for personal details by giving User id, secret key through which he can login into the mobile.

DETECTION METHOD:

The detection method must handle many features. The main objective here is to achieve more accurate performance than static or dynamic analysis. For this reason, it is necessary to design and implement a detection method that will improve the accuracy level.

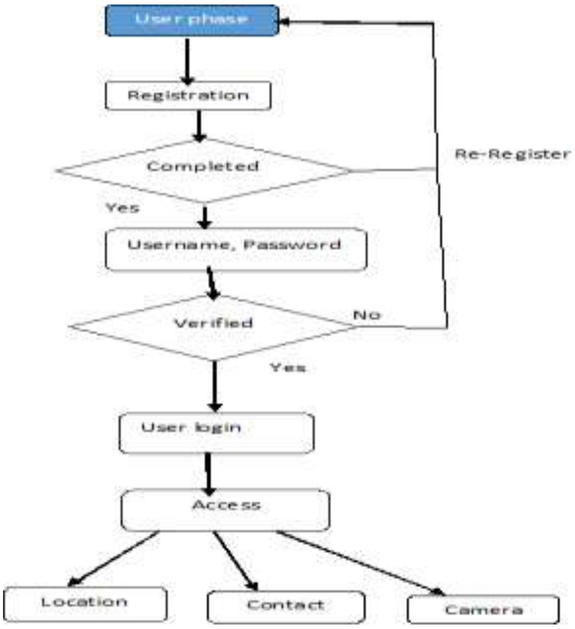


Fig 4: Malware Detection Flow chart

7. ALGORITHM

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

WORKING OF RANDOM FOREST ALGORITHM

Step 1 − First, start with the selection of random samples from a given dataset.

Step 2 − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

Step 3 − in this step, voting will be performed for every predicted result.

Step 4 − At last, select the most voted prediction result as the final prediction result.

PERMISSION ALGORITHM:

All mobile apps require permissions to run. These permissions are there to help protect your data. Yet many times, users accept all requested permissions when installing an app without reviewing why they are requested, giving mobile applications data they likely don't need. Since the Android security model is based on application permissions, the permission set was extracted from the manifest file. Every application must have the privileges needed to access different features. During the application installation, the Android platform asks the user whether to grant there quested permissions. There are some permissions, which can be exploited by malicious applications. For example, a malicious application may use the permissions to access the SD card and the Internet, in order to access and filter sensitive information on an SD card. Our approach is to model the group of Android permissions requested by malicious applications. Therefore, we propose a method that uses the appearance of a specific privilege as a feature of a machine learning algorithm. In this paper, we designed the association mining rule and ranked permission methods for the detection of the malwares.

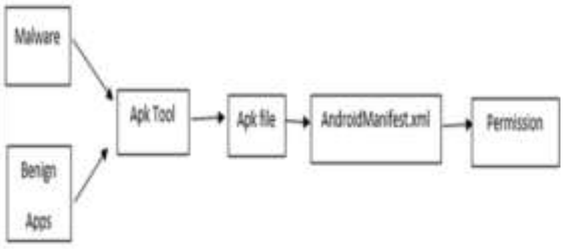


Fig 5: Permission Extraction

8. WORKING OF THE SYSTEM

Android mobile malware detection process. This comparative analysis has provided a broad sense of the Android Mobile Malware Detection process. After finding the malware application we use random forest algorithm. This algorithm will be used to increase the accuracy of the malware. Finally the malicious application will be detected. If we identify that the application will be malicious then we have to notify to the user about that application. And also detect which files are accessed by that application will be notify to the user. If the user want to block that application then the application will be blocked.

9. COMPARISON OF DETECTION METHODS

Mobile Malware detection commonly involves three techniques, namely static, dynamic, and hybrid detection techniques. The static analysis extracts features without execution of the application. The dynamic analysis involves the execution of the application, and hybrid analysis uses both static and dynamic features for the classification.

DATA SET:

The data sets that are used in the static analysis involve a wide range of malware and benign ware. The difficulty lies in obtaining a proper data set of malware because each day different types of apps are introduced in the official and third-party market places of Android. There is a corresponding quick development in the number of malwares and their types. Malware authors hide malicious activities and introduce newer techniques such as code obfuscation, update attacks, etc.

10. FEATURE EXTRACTION

Static features are extracted mainly from the Manifest file. Permissions are extracted from the Manifest file. The applications are applied to the APK tool which generates the APK file. The AndroidManifest.xml file contains the permission set. This xml file can be decompiled by the AXMLPrinter2 tool to obtain the permission.

11. FUTURE SELECTION

Feature selection is extremely important in hybrid analysis. Because of the many features available, feature selection must be precise and accurate. The time and space complexity of the selection method should also be considered. The appropriate feature set is selected using different selection methods such as information gain, mutual information, Fisher score, etc. The feature selection methods used are the same as for static or dynamic analysis. However, the importance of proper selection in hybrid analysis is much higher than in static or dynamic analysis. Unnecessary and irrelevant features can hamper the overall performance of the detection engine. Information gain and Fisher score are used as the selection methods in the research papers that are discussed here. The static features that are involved in hybrid analysis are Permission, API call, Java Package name, Intent receivers, Class structure, Crypto operations (cryptographic API) Services, Receivers, and Publisher ID. The most commonly used static feature is the permission in hybrid analysis. The more common dynamic features involved in hybrid analysis are System call, File operations, Network activity, dynamically loaded code, and dynamically registered broadcast receivers, Phone activity, and Crypto operations (cryptographic API). The most commonly used dynamic feature is the system call in hybrid analysis.

12. CONCLUSION

We have analyzed the static, dynamic, and hybrid analysis methods for mobile malware detection. The analysis includes recent literature on Zero-day detection. The detection process, feature extraction and selection process, and detection algorithms are discussed in this study. We have found that machine learning approaches are commonly used to classify malware and benign ware. The suspicious permission list, API call list, and the system call list are also identified to assist application developers. In future, we plan to design and implement a framework that will be able to detect mobile malware with a high accuracy to ensure Zero-day detection.

13. RESULT

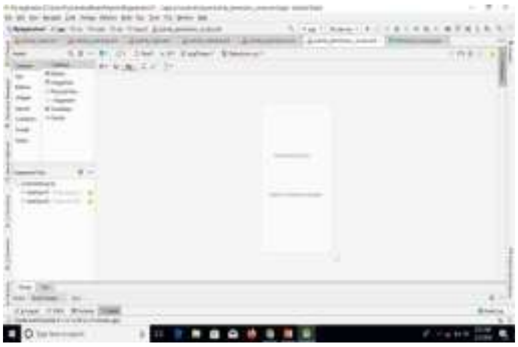


Fig 6: Malware deletion

REFERENCES

[1] Z. Aung and W. Zaw, ‘‘Permission-based Android malware detection,’’ Int. J. Sci. Technol. Res., vol. 2, no. 3, pp. 228–234, 2013.

[2] C.-Y. Huang, Y.-T. Tsai, and C.-H. Hsu, ‘‘Performance evaluation on permission-based detection for Android malware,’’ in Advances in Intelligent Systems and Applications (Smart Innovation, Systems and Technologies), vol. 2. Berlin, Germany: Springer, 2013, pp. 111–120.

[3] D.-J. Wu, C.-H. Mao, T.-E. Wei, H.-M. Lee, and K.-P. Wu, ‘‘DroidMat: Android malware detection through manifest and API calls tracing,’’ in Proc. 7th Asia Joint Conf. Inf. Secur., 2012, pp. 62–69.

[4] K. Xu, Y. Li, and R. H. Deng, ‘‘ICCDetector: ICC-based malware detection on Android,’’ IEEE Trans. Inf. Forensics Security, vol. 11, no. 6, pp. 1252–1264, Jun. 2016.

[5] H. Peng et al., ‘‘Using probabilistic generative models for ranking risks of Android apps,’’ in Proc. ACM Conf. Comput. Commun. Secur, 2012, pp. 241–252.

[6] D. Arp et al., ‘‘DREBIN: Effective and explainable detection of Android malware in your pocket,’’ in Proc. NDSS, 2014, pp. 23–26.

[7] Risteska Stojkoska, B.L.; Trivodaliev, K.V. A review of Internet of Things for smart home: Challenges andsolutions.J. Clean. Prod.2017, 140, 1454–1464.

[8] Park, J.S.; Youn, T.Y.; Kim, H.B.; Rhee, K.H.; Shin, S.U. Smart contract-based review system for an IoT datamarketplace.Sensors2018,18, 3577.

[9] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, ‘‘Supervised machine learning: A review of classification techniques,’’ Emerg. Artif. Intell. Appl. Comput. Eng., vol. 160, pp. 3– 24, 2007.

[10] S. Wu, P. Wang, X. Li, and Y. Zhang, ‘‘Effective detection of Android malware based on the usage of data flow APIs and machine learning,’’ Inf. Softw. Technol., vol. 75, pp. 17–25, Jul. 2016.

[11] M. Lindorfer, M. Neugschwandtner, and C. Platzer, ‘‘MARVIN: Efficient and comprehensive mobile app classification through static and dynamic analysis,’’ in Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf. (COMPSAC), Jul. 2015, pp. 422–433.

[12] Q. Jerome, K. Allix, R. State, and T. Engel, ‘‘Using opcode-sequences to detect malicious Android applications,’’ in Proc. IEEE Int. Conf. Commun. (ICC), Jun. 2014, pp. 914–919.

[13] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, ‘‘Supervised machine learning: A review of classification techniques,’’ Emerg. Artif. Intell. Appl. Comput. Eng., vol. 160, pp. 3–24, 2007.

[14] Yerima, S.Y.; Sezer, S.; Muttik, I. Android malware detection using parallel machine learning classifiers.In Proceedings of the 2014 8th International Conference on Next Generation Mobile Applications, Servicesand Technologies, NGMAST 2014, Oxford, UK, 10–12 September 2014.

[15] Walls, J.; Choo, K.K.R. A review of free cloud-based anti-malware apps for android. In Proceedings of the14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Helsinki, Finland, 20–22 August 2015; Volume 1, pp. 1053–1058.

[16] Damshenas, M.; Dehghantanha, A.; Choo, K.K.R.; Mahmud, R. M0Droid: An Android Behavioral-BasedMalware Detection Model.J. Inf. Priv. Secur.2015, 11, 141–157.