DroidDetector: An Android application based on Contrasting Permission Patterns

Chinmay Sanjay Kapare, Omkar Sharad Joshi, Ms. Vanessa Rumao

Abstract: **Malware detection is necessary in android devices because of Android OS recent popularity. To tackle this issue the proposed system is going to use Android permissions to characterize whether an application is malware or clean. By taking contrasting permission patterns into consideration, a classifier for detecting potentially malicious application is developed. In this classifier each permission pattern will act as a weak classifier. Weighted prediction of weak classifiers are aggregated to give final result.**

I. INTRODUCTION

Android devices has become a small computer having computing and connectivity features. It installs applications from sources like Google play store or other third party application. In many cases the apps from third party sources are malwares that are harmful. Android OS uses permissions that should be granted by user for an application to interact with system API, or with database. Android application's API provides access to the phone hardware, Wi-Fi, and also to cellular network. To give access priority to system resource, Android has access control through permissions granted by system.

A framework that contain hybrid profile based malware detection system is presented. For implementing this system Ensemble Classifier is made for malware detection.

1. Background and Related Works

Android is a Linux-based OS made up of different layers consisting of the Operating System, Java libraries and basic built-in applications. Additional application can be downloaded from official app stores or third party markets. An Android application includes two folders and one file: (i) Class, (ii) Resources and (iii) AndroidManifest.xml. This AndroidManifest.xml file contains the required permission that are explicitly declared by the developers.

Google uses permission system to restrict access to privileged system resources and users private data. These permission mentioned in the AndroidManifest.xml file needs user approval upon installation. There are 130 official Android permissions. These permissions can be further classified as Normal permissions and Dangerous permissions. In these Normal permissions doesn't need users approval whereas it is mandatory for users to approve of the Dangerous permission before installation. After installation the application is able to execute set of API calls. Each API calls is associated with a specific android permission. So android checks before executing the API call whether the permission associated with it is approved by the user.

A number of research works has be conducted to understand Android permissions and their combinations. Zhou and Jiang [1] did analysis of required permissions of 1260 malicious applications and 1260 clean ones and listed the top required permissions for both of them. With this research they were able to find significant differences in permissions requested by malwares and clean applications respectively. It was found that malicious applications requested more permissions compared to clean applications. These research works inspired in discovering contrasting permission patterns for characterizing and differentiating clean and malicious applications. Frank et al. [2] proposed a probability model that can be used to identify the commonly required permission patterns. Veelasha et al. [3] developed a contrast mining algorithm to identify the permissions that can distinguish malware from clean Android applications. Shaowu Liu [4] has proposed the mining of contrasting permission patterns considering required as well as used permission in his paper.

II. MALWARE DETECTION WITH PERMISSIONS

2.1Contrasting permission pattern

Considering a training dataset, it is split into two subsets: One containing malicious android applications and other containing clean applications. By applying Associative Rule mining on these two subsets, three types of permission patterns were created:

(1) Malicious Permission Patterns: These are unique frequently required permission patterns found only in malware dataset. Hence, the support degree of their item sets in clean dataset is 0.

(2) Clean Permission Patterns: These are unique frequently required permission patterns found only in clean dataset. Hence, the support degree of their item sets in malware dataset is 0.

(3) Commonly required Permission Patterns: These are the frequently required permission patterns that are found in both the dataset. In such case these patterns have different support degree in two datasets

These frequent item-sets are called as Contrasting Permission Patterns. If an unknown application has more malicious permission patterns than the clean patterns then it can be said that the application is a malware and vice versa. But if the unknown application contains commonly required permission patterns the difference in support degrees in respective datasets is considered for distinguishing the application. Suppose the common permission pattern has a higher support degree in malware dataset then the app is more likely to be a malware.

2.2 Detection Framework

In this section, Android malware detection framework that is based on contrasting permission patterns and the architecture of the DroidDetector application is described.

By using Contrasting permission patterns the Hybrid Profile created used by classifier for distinguishing malware and clean applications. The Hybrid profile consist of three sub-profiles as follows:

(1) Malware Profile:
Contains all the unique permission patterns in malware dataset.

(2) Clean Profile:
Contains all the unique permission patterns in clean dataset.

(3) Common Profile:
Contains all commonly required permission patterns in both datasets. The difference in support degree is used to classify unknown applications.

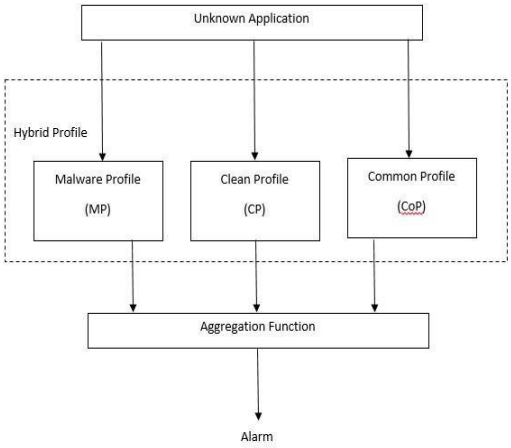

Fig.2.2 Hybrid Profile in the framework

2.3 Architecture of DroidDetector

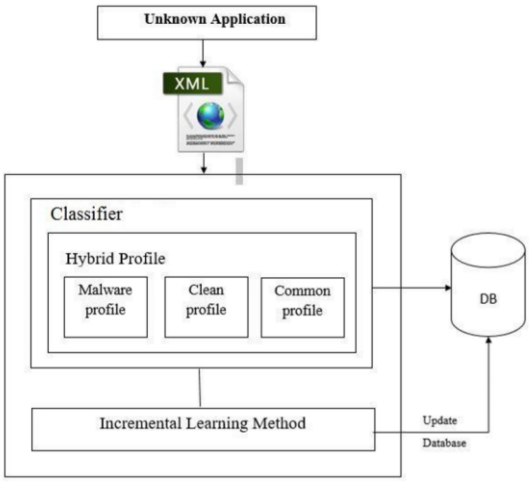

Fig.2.3.1 Block Diagram of DroidDetector

The above block diagram presents the architecture of DroidDetector. The application will first will scan the AndroidManifest.xml file of the unknown application that has to be determined whether it is malware or clean. DroidDetector will extract all the permission from the AndroidManifest.xml file and stored in a permission file so that it can be further send for pattern matching. The classification framework with the hybrid profile contains the 3 sub-profiles malware, clean and common profile. All the permission patterns present in this three profiles are considered as weak classifiers. For each weak classifier a support degree value is associated with it. The permission file is then compared with all the weak classifiers present in all three profiles for pattern matching so that it can get weighted predictions to calculate aggregated final result. This final result will decide whether the unknown application is a malware or clean application. Since the application has to be updated with latest malware types and varieties it will use incremental learning method to update the database from time to time.

Structure of Classifier

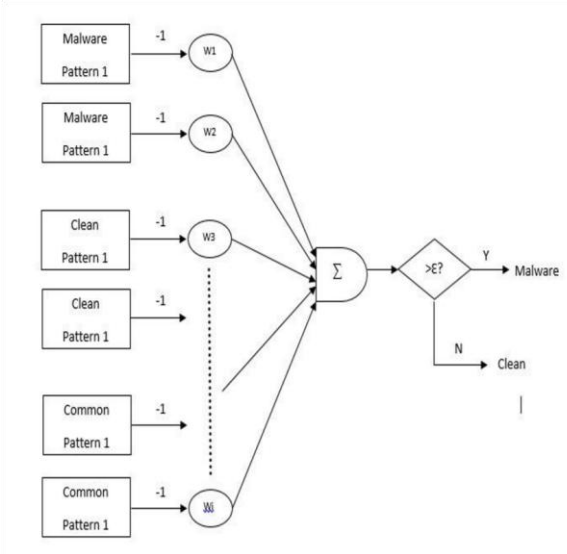

Fig.2.3.2 Structure of Classifier

Classification framework consists of the contrasting permission patterns that was discovered in the training set and along with the set of parameters used for detection of malwares. For that, the training set is divided into two subsets- Malware and Clean. Then association rules mining is conducted on them to discover all the contrasting permission patterns that are then divided into MP, ClP and CoP. All the contrasting permission patterns contained in respective subset are the weak classifiers in classification framework. Each weak classifier has a support degree associated with it in both dataset. For example, if a particular permission pattern x is present in malware profile, output is '+1', if the pattern is found in clean profile, output is '-1', and for patterns in common profile, the output is '+1', but the final result is decided by the difference in the support degree of that pattern in the clean and malware profile. The logical structure of classification framework is illustrated in Fig. 3.3.2 At the end, aggregated result of all considered weak classifiers is used to decide whether the scanned application is malware or not.

III. CONCLUSION

In this paper, a malware detection in android platform using Contrasting permission patterns is presented. These contrasting permission patterns and their support degrees are the characteristics that help us in discriminating and classifying malicious applications from clean applications. For malware detection, classifier is created. This classification framework contains hybrid profile, which contains three sub-profiles: Malware, clean and common profiles. The Malware profile and clean profile contains unique permission patterns that are found in malware and clean datasets respectively and the common profile contains the permission patterns that are found common in both dataset but have varying support degree in both datasets. With the help of support degree values and prediction values, the aggregate result of all weak classifiers are found and consequently decide whether the application is a malware or clean. DroidDetector provide the incremental learning techniques for being up-to-date will new type and varieties of malware.

IV. REFERENCES

[1] ZHOU Y, JIANG X., "Dissecting Android Malware: Characterization and Evolution" [C]//Proceedings of the 2012 IEEE Symposium on Security and Privacy: May 21-23, 2012, San Francisco, California, USA: IEEE Computer Society, 2012: 95-109.

[2] FRANK M, DONG B, FELT A P, et al., "Mining Permission Request Patterns from Android and Facebook Applications"[C]//Proceedings of the 2012 IEEE 12th International Conference on Data Mining: Dec 10-13, 2012, Brussels, Belgium: IEEE Computer Society, 2012: 870-875.

[3] MOONSAMY V, RONG J, LIU S, et al., "Contrasting Permission Patterns between Clean and Malicious Android Applications"[M]//ZIA T.,

A. ZOMAYA, V. VARADHARAJAN, and M. MAO. Security and Privacy in Communication Networks. Springer International Publishing, 2013: 69-85.

[4] Moonsamy V, Rong J, Liu S, et al., "Mining Permission Patterns for Contrasting Clean and Malicious Android. Applications"//Proceedings of the IEEE Symposium on Security and Privacy (SP 2012), pages 95–109, San Francisco, CA, May 2012.

Chinmay Sanjay Kapare,B.E. (Computer Science Engineering) pursuing from St. John College of Engineering and Technology, Palghar.

Omkar Sharad Joshi,B.E. (Computer Science Engineering) pursuing from St. John College of Engineering and Technology, Palghar.

Ms. Vanessa Rumao,Assistant Professor at St. John College of Engineering and Technology, Palghar.

