# Momentous Permission Identification for Android Apps Malware Detection

K. Sudha Devi
Professor
Dept. of Information Technology
Vivekanandha College of Technology for Women
Namakkal, Tamil Nadu

Gornisha. C
Dept. of Information Technology
Vivekanandha College of Technology for Women,
Namakkal, Tamil Nadu

Abirami. R
Dept. of Information Technology
Vivekanandha College of Technology for Women
Namakkal, Tamil Nadu

Sonipriya. M
Dept. of Information Technology
Vivekanandha College of Technology for Women
Namakkal, Tamil Nadu

***Abstract:-*** **The project titled "Momentous Permission Identification for Android Apps Malware Detection" Unlike other competing smart-mobile device platforms, such as iOS, Android allows users to install applications from unverified sources such as third-party app stores and file-sharing websites. The malware infection issue has been so serious that a recent report indicates that 97% of all mobile malware target Android devices. To address the elevating security concerns, researchers and analysts have used various approaches to develop Android malware detection tools. So a scalable malware detection approach is required that effectively and efficiently identifies malwares. Various malware detection tools have been developed, including system-level and network level approaches. However, scaling the detection for a large bundle of apps remains a challenging task. So this project introduces Significant Permission IDentification (SigPID), a malware detection system based on permission usage analysis to cope with the rapid increase in the number of Android malware. Instead of extracting and analyzing all Android permissions, this project develops three levels of pruning by mining the permission data to identify the most significant permissions that can be effective in distinguishing between benign and malicious apps. Then it utilizes machine-learning-based classification methods to classify different families of malware and benign apps. This project identifies dangerous permission list, benign permission list and reduce non-sensitive permissions and apply SVM classification on the new data set.**

## 1. INTRODUCTION

The first component of SIGPID is the MLDP process to identify significant permissions to eliminate the need of considering all available permissions in Android. No app requests all the permissions, and the ones that an app requests are listed in the Android application package (APK) as part of manifest.xml. When we need to analyze a large number of apps (e.g., several hundred thousand), the total number of permissions requested by all apps can be overwhelmingly large, resulting in long analysis time. This high analysis overhead can negatively affect the malware detection efficiency as it reduces analyst productivity. We propose three levels of data pruning methods to filter out permissions that contribute little to the malware detection effectiveness.

Thus, they can be safely removed without negatively affecting malware detection accuracy. The complete three-step procedure is illustrated in Fig. 2. We then describe each level in the pruning process.

1) Permission Ranking with Negative Rate: Each permission describes a particular operation that an app is allowed to perform.

For instance, permission INTERNET indicates whether the app has access to the Internet. Different types of benign apps and malicious apps may request a variety of permissions corresponding to their operational needs. For malicious apps, we hypothesize that their needs may have common subsets and we do not need to analyze all the permissions to build an effective malware detection system.

As a result, on one hand, our focus is more on the permissions that create high-risk attack surfaces and are frequently requested by malware samples. On the other hand, the permissions that are rarely requested by malware samples are also good indicators in differentiating between malicious and benign apps. Therefore, our pruning procedure identifies both types of highly differentiable permissions so that we can use this information to classify malicious and benign apps. At the same time, we exclude permissions that are commonly used by both benign and malicious apps, as they introduce ambiguity in the malware detection process.

For instance, permission INTERNET are frequently requested by both malware and benign apps, as almost all apps will request to access the Internet. Therefore, this approach prunes permission INTERNET. To identify these two types of significant permissions, we design a permission ranking scheme to rank permissions based on how they are used by malicious and benign apps. Ranking is not a new concept. Prior works have also used a generic permission ranking strategy such as mutual information to identify high-risk permissions.

However, their approaches tend to only focus on high-risk permissions and ignore all the low-risk permissions, which are defined as significant permissions in this approach. There as on that prior works ignoring low-risk permissions is that they

are interested in identifying the permissions abused by malware, while the goal is to differentiate between malware and benign apps. In essence, risky permissions only focus on the permissions that can help detect the malware, while significant permissions not only care about the identification of the malware, but also take into account whether benign apps can be identified or not.

## II. EXISTING SYSTEM

The existing system focuses on Significant Permission Identification (SIGPID), an approach that extracts significant permissions from apps and uses the extracted information to effectively detect malware using supervised learning algorithms. The design objective of SIGPID is to detect malware efficiently and accurately. As stated earlier, the number of newly introduced malware is growing at an alarming rate. As such, being able to detect malware efficiently would allow analysts to be more productive in identifying and analyzing them. This approach analyzes permissions and then identifies only the ones that are significant in distinguishing between malicious and benign apps. This includes a multilevel data pruning (MLDP) approach including permission ranking with negative rate (PRNR), permission mining with association rules (PMAR), and support-based permission ranking (SPR) to extract significant permissions strategically.

**Existing System Disadvantages**

- SVM Classification is not considered so that probability of benign/suspicious apps in the given new test data is not possible.
- Feature reduction (based on unique values in permission list) before malware identification is not carried out.
- Comparison between all permission list and feature reduced permission list based SVM classification is not included.

## III. PROPOSED METHOD

The proposed system also focuses on Significant Permission Identification (SIGPID). In addition identification of dangerous, benign as well as shutdown enabled permission list is also carried out. Feature reduction is also carried out. SVM classification for both all permission list as well as feature reduced data set is included.

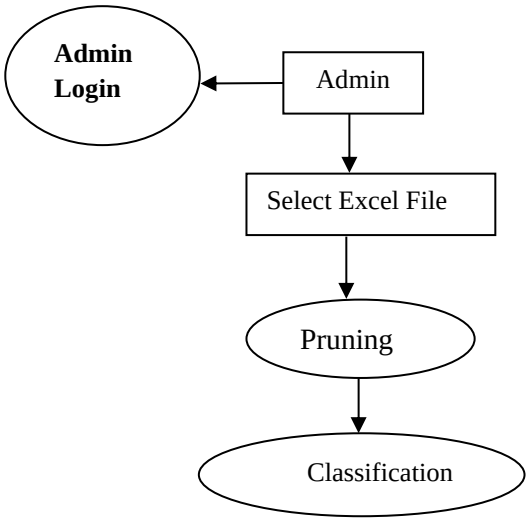
**Proposed System Advantages**

- SVM Classification is considered so that probability of benign/suspicious apps in the given new test data is possible.
- Feature reduction (based on unique values in permission list) before malware identification is carried out.
- Comparison between all permission list and feature reduced permission list based SVM classification is included.

## IV.DATA FLOW DIAGRAM

### 1. DATA SET COLLECTION

All permission details of the app are saved in a single Excel workbook as records. This is the input for the project.




### 2. FINDING DANGEROUS PERMISSIONS LIST

Certain permission values such as READ_SMS, WRITE_SMS and the like are checked for values with ‘1’ so that the apps are declared as dangerous and listed.

### 3. FINDING BENIGN PERMISSIONS LIST

Certain permission values such as BIND_SERVICE and the like are checked for values with ‘1’ so that the apps are declared as benign and listed.

### 4. PERMISSION RANKING WITH NEGATIVE RATE

This module referred to as PRNR, provides a concise ranking and comprehensible result. The approach operates on two matrices, M and B. M represents a list of permissions used by malware samples and B represents a list of permissions used by benign apps. Mij represents whether the jth permission is requested by the ith malware sample, while “1” indicates yes and“0” indicates no. Bij represents whether the jthpermission is requested by the ith benign app sample.

Before computing the support of permissions from matrices M and B, it first checks their sizes. Typically, the number of benign tends to be much larger than the number of malicious apps; therefore, the size of B is much larger than the size of M. With this ranking scheme, it prefers the dataset on the two matrices to be balanced. The PRNR algorithm is used to perform ranking of the datasets. In the formula above, R(Pj) represents the rate of the jth permission. The result of R(Pj) has a value ranging between [−1, 1]. If R(Pj)=1, this means that permission Pj is only used in the malicious dataset, which is a high-risk permission. If R(Pj) = −1, this means that permission Pj is only used in the benign dataset, which is a low-risk permission. If R(Pj)=0, this means that Pj has a very little impact on malware detection effectiveness.

**5. PERMISSION MINING WITH ASSOCIATION RULE**

In this module, after pruning some permission by using PRNR and SPR with the PIS, it can remove non-influential permissions even more. By inspecting the reduced permission list that contains some significant permissions, it finds three pairs of permissions that always appear together in an app. For example, permission WRITE_SMS and permission READ_SMS are always used together. They also both belong to the Google's "dangerous" permission list. Yet, it is unnecessary to consider both permissions, as one of them is sufficient to characterize certain behaviors. As a result, we can associate one, which has a higher support, to its partner. In this example, we can remove permission WRITE_SMS. In order to find permissions that occur together, it proposes a PMAR mechanism using the association rule mining algorithm.

**6. SVM CLASSIFICATION**

In this module, 70% of the data in given data set is taken as training data and 30% of the data is taken as test data. The model is trained with training data and then predicted with test data. Of which, most of the apps are classified as Benign and fewer apps are classified as Suspicious.

**7. FEATURES REDUCTION**

In this module, each column values are taken and find the number of '1's and '0' and their percentage is calculated. If any one of the percentage is above 95%, then the column is treated as non-sensitive and can be eliminated.

**8. SVM CLASSIFICATION IN FEATURES REDUCED DATA SET**

In this module, 70% of the data in given data set is taken as training data and 30% of the data is taken as test data but with the columns after feature reduction. The model is trained with training data and then predicted with test data. Of which, most of the apps are classified as Benign and fewer apps are classified as Suspicious.

**9. ASSOCIATION RULE MINING**

In this module, all the permissions are iterated in for loop and three columns are taken to find permission value '1' along with next fourth column with permission value '1'. If the count of three columns values matched with count of fourth column then it is found out there is an association rule and printed out. The iteration continues for all 216 permissions.

**10. MUTUAL INFORMATION**

In this module, mutual information is found out as follows: Let X denote a permission variable and C be the class variable. The relevance of X and C can be measured by mutual information of them as

$$I(X,C) = \sum_{x_i}\sum_{c_j} P(X = x_i, C = c_j)\log\frac{P(X = x_i, C = c_j)}{P(X = x_i)P(C = c_j)} \quad (1)$$

Where P(C = cj) is the frequency count of class C with value cj, P(X = xi) is the frequency count of permission X with value xi, and P(X = xi,C = cj) is the frequency count of X with value xi in class cj. In this paper, the class C has binary values, c0 for benign apps and c1 for malicious apps. Each permission X is a Boolean variable with value 1 or 0. I(X, C) is nonnegative in [0, 1]. I(X, C) = 0 indicates no correlation, while I(X,C) = 1 means that C is completely inferable by knowing X.

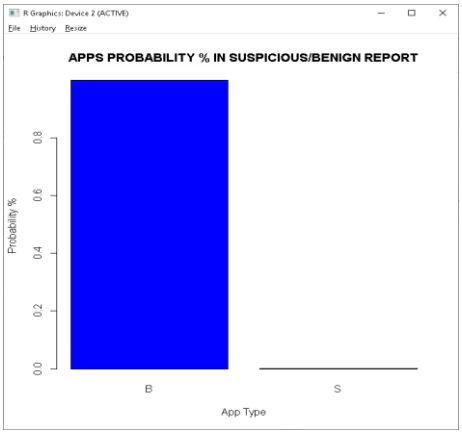
**11. PEARSON CORRELATION COEFFICIENT**

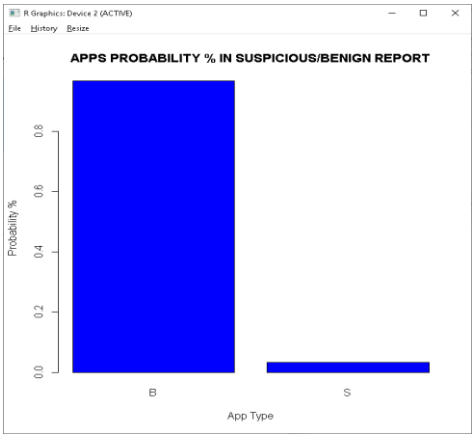
In this module, Pearson correlation coefficient is found out as follows: Pearson Correlation Coefficient measures the relevance of X and C by

$$R(X,C) = \frac{\sum_{n=1}^{N}(X_n - \bar{X})(C_n - \bar{C})}{\sqrt{\sum_{n=1}^{N}(X_n - \bar{X})^2 \sum_{n=1}^{N}(C_n - \bar{C})^2}}$$

where $\bar{X}$ (resp. $\bar{C}$) is the average of all sample values of X (resp. C), Xn (resp. Cn), n = 1...N. R(X,C) has a value in [−1, 1], where R(X,C) = 0 indicates the independency of X and C, R(X,C) = 1 indicates the strongest positive correlation of them and R(X,C) = −1 indicates the strongest negative correlation. R(X,C) = 1 means that permission request of X makes apps highest risky, while R(X,C) = −1 means that permission request of X makes apps lowest risky.

## VI. RESULT




## VII. CONCLUSION

The proposed framework demonstrated how it is possible to reduce the number of permissions to be analyzed for mobile malware detection, while maintaining high effectiveness and accuracy. It has been designed to extract only significant permissions through a systematic three-level pruning approach. The existing system considers 22 permissions for malware apps but the proposed system analyzes 47 permissions are malware apps for the given data set. The

difference is due to the non-sensitive permission features reduction. By adjusting the unique percentage in values of particular permission, the malware surety would be raised or lowered.

## REFERENCE:

[1] Baeza-Yates.R, and Ribeiro-Neto.B, "Modern Information Retrieval" Addison-Wesley, June 1999.
[2] Cristianini N and Shawe-Taylor J, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.
[3] Fergus.R, Fei-Fei.L, Perona.P, and Zisserman.A. Learning object categories from google's image search.
[4] Hand D., Mannila H. and Smyth P., Principles of Data Mining, MIT Press, 2001.
[5] Strachey and Christopher "Time Sharing in Large Fast Computers" Proceedings of the International Conference on Information processing, UNESCO, pp: 336–341.
[6] Amudha, K, Nelson Kennedy Babu, C & Balu, S 2017, 'Hybrid Baker Map with AES in Cipher Block Chaining Mode in Medical Images', International Research Journal of Pharmacy, vol.8, issue.4,pp.67-73.
[7] Larose D.T "Discovering knowledge in data: an introduction to data mining", Wiley-Interscience, 2005.
[8] Mitchell T.M., Machine learning, McGraw-Hill, 1997.
[9] Pal S.K. and Mitra P, Pattern Recognition Algorithms for Data Mining, CRC Press, 2004.
[10] Pankaj Jalole, "An Integral approach to software engineering",Narosa publishing Home-3rd Edition.
[11] Smith and David Mitchell "Hype Cycle for Cloud Computing", 2013.