International Journal of Computer
Science and Engineering (IJCSE)
ISSN(P): 2278–9960; ISSN(E): 2278–9979
Vol. 9, Issue 4, Jun–Jul 2020; 1–10

International Academy of Science,
Engineering and Technology
Connecting Researchers; Nurturing Innovations

EVALUATING SECURITY DETECTION METHODS OF ANDROID APPLICATIONS

Srinivas B K[1] & Sinchana Gowda R[2]

[1]Assistant Professor, Information Science Department, RVCE, Bengaluru, India

[2]Research Scholar, Information Science Department, RVCE, Bengaluru, India

ABSTRACT

This paper deals with the survey of android application security detection methods based on the permissions. To combat a serious malware campaign, we need a scalable malware detection approach that can effectively and efficiently identify malware apps. The proposed survey is to provide the different models to detect the android security of the applications based on the permissions that the applications request from the users during installations. Accuracy, precisions, recall values are calculated using true and false positive and negative rates, depending on these the best models are selected to detect the security of the android application.

KEYWORDS: *Attacks, Android Security, Permissions, Methods*

Article History

Received: *21 May 2020* **|** ***Revised:*** *30 May 2020* **|** ***Accepted:*** *04 Jun 2020*

INTRODUCTION

Quick advancement in android Technology has expanded the danger of intrusion. At that point, the android operating system uses the permission based model, which allows Android applications to access user information, system information, device information, and external resources of smart phones. The designer needs to declare the permissions for the Android application. The user needs to accept these permissions for the successful installation of an Android application. These permissions are declarations. At the time of installation, if the permissions are allowed by the user, the app can access resources and information anytime. It needs not request for permissions again. The android operating system is susceptible to different security assaults because of its shortcoming in security.

Why Android Application Security is Important

With the number of android device users increasing year after year, the need for mobile security is also gaining ground. Android phones have now become vulnerable because of the rapid progress in the mobile phone industry and the introduction of cloud services and apps. In other words, android security has yet to reach millions of users, unlike the progress seen in the overall usage of smart phones.

The Two main reasons why android security are important due,

- Android security protects you against malvertisers
- Android security protects your private data

To provide the security, android security detections methods play a important role in safeguarding the android applications.

SECURITY ATTACKS IN ANDROID

Permission Escalation Attack

It allows a malicious application to collaborate with other applications so as to access critical resources without requesting for corresponding permissions explicitly.

Collision Attack Android Supports Shared User ID

It is technique, wherein two or more application shares the same user id so that can access the permissions, which are granted to each other. For example, if application A has permissions to READ_CONTACTS, READ_PHONE_STATUS and B has permissions to READ_MESSAGES, LOCATION_ACCESS, if both the applications use the same user id SHAREDUSERID, then it is possible for application A to use the permissions granted to it and the permissions granted to B.

Time of Check and Time of Use Attack

The primary purpose behind the TOCTOU Attack is naming collision. No naming rule or constraint is applied to a new permission declaration. In addition, permissions in Android are represented as strings, and any two permissions with the similar name string are treated as equivalent even if they belong to discrete applications.

Spyware

Spyware is a kind of malware. It is an apk file which is downloaded automatically when the user visits malicious site and applications installed from obscure sources. In Android, other than Google play store, it is possible to install the applications from unknown sources. Spyware is one of the main reasons for significant security threats Android operating system.

UNDERSTANDING PERMISSIONS

The Android operating system utilizes the permission-based model to access different resources and information. These permissions are not requests; they are declarations. These permissions are declared in the AndroidManifest.xml file. In Android, android versions 7 and higher the application permissions are classified into normal permissions and dangerous permissions.

Normal Permissions

Normal permissions don't unequivocally risk the customer's protection and permission need not be pronounced in the AndroidManifest.xml file. These permissions are allowed naturally. Example:

KILL_BACKGROUND_PROCESSES

SET_WALLPAPER

UNINSTALL_SHORTCUT

WRITE_SYNC_SETTINGS

Dangerous Permissions

Dangerous Permissions can access critical resources of the mobile. Dangerous permissions can give the app access to the user's private information. If the app lists normal permission in its manifest, the system grants permission automatically. If the app list dangerous permission, the user has to explicitly give approval for the app for the successful installation of the application. Example:

CONTACTS: READ_CONTACTS, WRITE_CONTACTS,

LOCATION: ACESS_FINE_LOCATION, ACCESS_COARSE_LOCATION,

SMS: SEND_SMS, RECEIVE_SMS, READ_SMS,

STORAGE: READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE.

Misuse of Application Permissions and Failure of Two Factor Authentication

Because of various app permissions, it is possible for various security threats. Among various threats, it is possible for Android applications to read messages, send messages. All confidential information based on two-factor authentication has been sent as a text message. For instance, different banks, online websites, etc., utilizes two-factor authentication. The primary target of two-factor authentication is to increase the security and integrity of the users and to avoid various security attacks that are based on the traditional username and password approach.

SECURITY DETECTION METHODS BASED ON PERMISSIONS

1. Metropolis Detection Method

Android application's security detection method based on the Metropolis algorithm is designed. This method analyzes Android's 24 dangerous permissions using the Metropolis algorithm, removes uncertainty permissions, and extracts certain permission features. Classification technology of machine learning is used to learn and classify the feature. Example, this methodology uses 15 1 Android Apps samples. From the analysis, this method reduces the detection features and the accuracy of malicious applications. Detection accuracy can reach 93.5% [1].

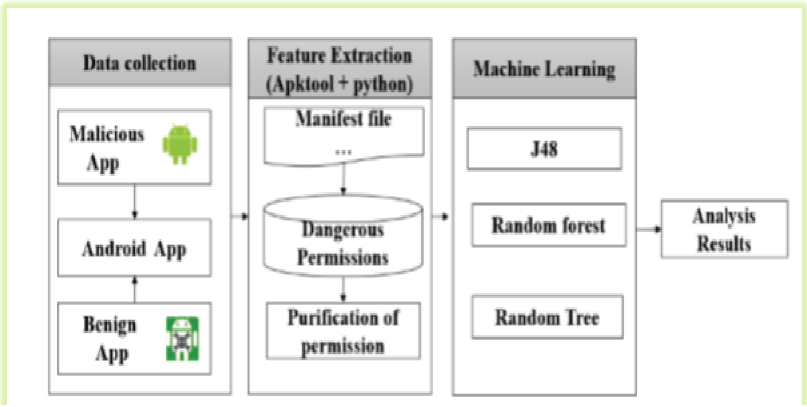

Figure 1: System Model of Detection Method using Metropolis Algorithm.

Metropolis algorithm gets the certain permissions from application. This system model consists of four steps: dataset collection, Then decompiling apk and extract apk's certain permissions, using machine learning to train and test, and result analysis. Figure 1 shows the system model of android security detection method using metropolis algorithm.

Collect Apk Samples

In this method, python program is used to crawl being Apps from the site. Malicious Apps were downloaded in the drain dataset.

Apk Decompile and Feature Extraction

Decompile the apk file with Apk Tool. Extracted the Android Manifest.xml file containing the permission declaration. Extract the dangerous permissions of 24 Google claims from android in the android Manifest.xml file. Then, using the Metropolis algorithm to extract the certain permission for malicious app detection.

Machine Learning-Based Classification

Machine learning is performed by deterministic permissions on the app classification. The data was trained and tested by J48 machine learning classification algorithms. The test method uses 1 - fold cross-validation test method.

Experimental Results Evaluation and Analysis

Through machine learning and classification, divides apps into benign apps and malicious apps. Then, using the Precision, Recall, F-Measure, Accuracy is calculated to evaluate a result.

Group	*number*	*Dangerous Permissions*
CONTACTS	3	WRITE_CONTACTS GET_ACCOUNTS READ_CONTACTS
PHONE	7	READ_CALL_LOG READ_PHONE_STATE CALL_PHONE WRITE_CALL_LOG USE_SIP PROCESS_OUTGOING_CALLS ADD_VOICEMAIL
CALENDAR	2	READ_CALENDAR WRITE_CALENDAR
CAMERA	1	CAMERA
SENSORS	1	BODY_SENSORS
LOCATION	2	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION
STORAGE	2	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE
MICROPHONE	1	RECORD_AUDIO
SMS	5	READ_SMS RECEIVE_WAP_PUSH RECEIVE_SMS RECEIVE_MMS SEND_SMS

Figure 2: Shows the Features Considered in Metropolis Method.

2. Two Layered Model

The detection model is designed based on the declared permission and consists of two layers. The first layer of detection utilizes an improved random forest algorithm for the analysis. The second layer detection uses sensitive permission rules matching to analyze the fuzzy sets generated by the first layer detection. At last, a series of evaluation methods were used to evaluate this detection [2]

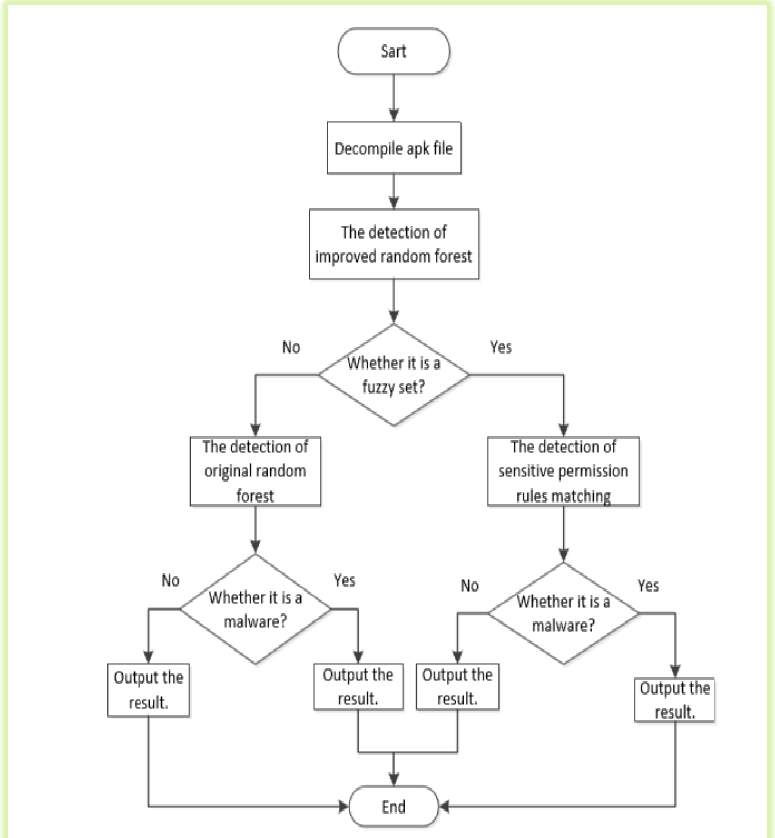

Figure 3: Two Layered Model.

READ_SMS,WRITE_SMS, ACCESS_NETWORK_STATE
RECEIVE_SMS,SEND_SMS, READ_PHONE_STATE
SEND_SMS, READ_CONTACTS
CALL_PHONE, SEND_SMS,READ_SMS, INTERNET
SET_DEBUG_APP
PHONE_STATE, RECORD_AUDIO, INTERNET

PROCESS_OUTGOING_CALL, RECORD_AUDIO,INTERNET
ACCESS_FINE_LOCATION, INTERNET,RECEIVE_BOOT_COMP LETE
ACCESS_COARSE_LOCATION, INTERNET,RECEIVE_BOOT_COMP LETE
RECEIVE_SMS, WRITE_SMS
SEND_SMS, WRITE_SMS
INSTALL_SHORTCUT, UNINSTALL_SHORTCUT
SET_PREFERRED_APPLICATION

Figure 4: Shows the Features Consider in the Two Layered Method.

Steps Followed First Layer in the Two Layered Model

Improved Random Forest Algorithm

Improved the random forest algorithm by introducing fuzzy sets which contain the samples with uncertain permission features. If the random forest is composed of M decision trees, N1 trees results are benign (malware), N2 trees results are malware, Where N1+N2=M. Only if $|N1-N2|<\delta$, this sample will be classified as a fuzzy sample. Otherwise, it will be classified as a determinate sample.

Collection of Android Samples

In this method, collected 1, 3 different types of Android applications from Baidu application market, used Virus Total and other anti-virus software for testing.

Number of Decision Trees

The number of decision trees is determined by the specific data set.

The Second Layer: Fuzzy Sets Detection

In order to analyze the fuzzy sets, the sensitive permission groups of the malware and benign samples. The fuzzy sets are detected by the matching rule of the sensitive permission groups.

Steps Followed First Layer in the Two Layered Model

- Read the permission list of the application and sensitive permission rules.
- Match the application permission list with the sensitive permission rules.
- Output the result of detection

3. SIGPID Method

SIGPID, a malware detection system based on permission usage analysis. In this method, developed 3-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. SIGPID then uses AI-based classification techniques to characterize various groups of malware and kind applications. Evaluation finds that only 22 permissions are significant. Then compare the performance of the SIGPID approach using only 22 permissions, against a baseline approach that analyzes all permissions. The results indicate that when Support Vector Machine (SVM) is used as the classifier; it can achieve over 9 % of precision, recall, accuracy, and F-measure. SIGPID is more effective by detecting 93.62% of malware in the data set, and 91.4% unknown/new malware samples [3].

Multi-Level Data Pruning (MLDP)

The First component of SIGPID is the multi-level data pruning process to identify significant permissions to eliminate the need of considering all available permissions in Android.

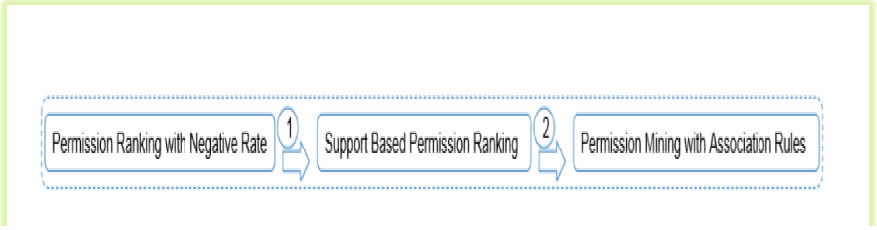

Figure 5: Multi-Level Data Pruning Process.

MLDP	
22 Permissions	
ACCESS_WIFI_STATE	READ_LOGS
CAMERA	READ_PHONE_STATE
CHANGE_NETWORK_STATE	READ_SMS
CHANGE_WIFI_STATE	RECEIVE_BOOT_COMPLETED
DISABLE_KEYGUARD	RESTART_PACKAGES
GET_TASKS	SEND_SMS
INSTALL_PACKAGES	SET_WALLPAPER
READ_CALL_LOG	SYSTEM_ALERT_WINDOW
READ_CONTACTS	WRITE_APN_SETTINGS
READ_EXTERNAL_STORAGE	WRITE_CONTACTS
READ_HISTORY_BOOKMARKS	WRITE_SETTINGS
Mutual Information	

Figure 6: Shows the Features Consider in SIGPID Method.

This system MLDP consists of three major components, designed based on real-time data analysis:

Permission Ranking with Negative Rate (PRNR)

This 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.

$$S_B(P_j) = \frac{\sum_i B_{ij}}{size(B_j)} * size(M_j), \quad (1)$$

Where, Pj denotes the j^{th} permission and SB (Pj) represents the support of j^{th} permission in matrix B. PRNR can then be implemented using Equation 2:

$$R(P_j) = \frac{\sum_i M_{ij} - S_B(P_j)}{\sum_i M_{ij} + S_B(P_j)} \quad (2)$$

Support Based Permission Ranking (SPR)

To further reduce the number of permissions is to focus on *n* the support of each permission. However, the support of permission is too low, it does not have much impact on malware detection performance

Permission Mining with Association Rules

Further permissions that occur together can be grouped into one, which has a higher support using permission mining with association rules (PMAR) mechanism using association rule mining algorithm.

After pruning, SIGPID method employs supervised machine learning classification methods, to identify potential Android malware.

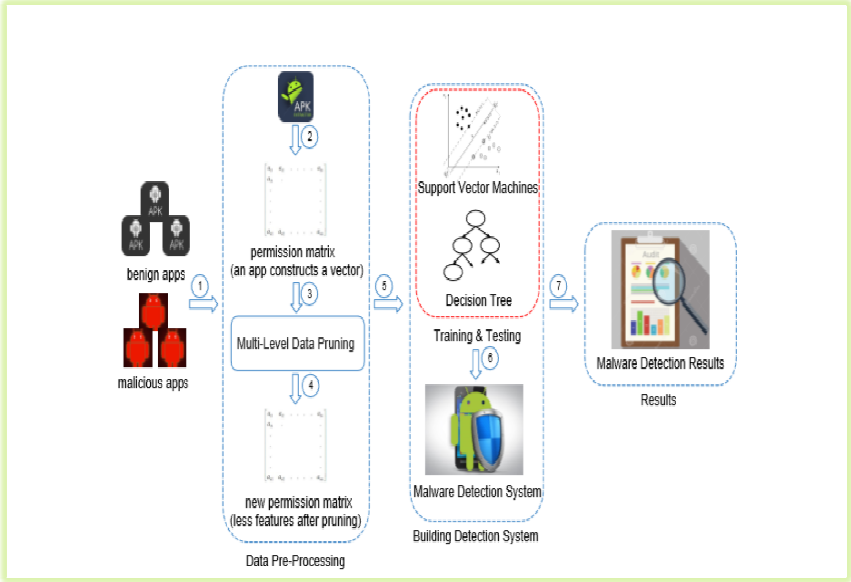

Figure 7: System Model of SIGPID.

Machine-Learning Based Malwar Detection using Significant Permissions

SVM and a small data set to test the MLDP model. SVM determines a hyper plane that separates both classes with a maximal margin based on the training dataset that includes benign and malicious applications.

Progressively, worthwhile to use MLDP to perform malware detection as it can be as effective while notably conserving time and memory. Since time and memory are constrained in common computers and it tends to be outsourcing the task to the cloud in a reasonable manner to help efficiency.

RESULTS

For all the above methods mentioned results are evaluated using following assessment methods

- Accuracy: Accuracy = (TP+TN) / (TP+FN+FP+TN), which means, the number of samples correctly classified by the detection model divided by all the number of samples.
- Precision: Precision=TP / (TP+FP), which means the proportion of the samples, which are classified as malware by the detection model is actually a benign category.
- Recall: Recall=TP / (TP+FN) which means the proportion of the malware samples which are classified as malware by the detection mode

Table 1: Shows the Accuracy, Precision, Recall Values (%) of Security Detection Methods

Sl.no	Security Detection Methods(Based on Permissions)	Data sets	Accuracy	Precision	Recall	Advantages	Disadvantages
1	The two layered detection method	1300	86	86	91.4	Improves the detection accuracy to a certain extent. By improving the random forest algorithm to produce fuzzy sets	The accuracy rate is not very large increase due to the shortcomings of static detection itself
2	Metropolis algorithm method	1501	93.5	93.5	93.5	Method has higher accuracy while streamlining features	Only 24 dangerous permissions are considered.
3	SIGPID method	54694	93.67	95.15	92.17	Compared 24 dangerous permissions identified by Google, Algorithm can retain more significant permissions by pruning less important permissions	MLDP Conservers more time and memory. Since it's a important factor this MLDP method usage is limited in common computers

CONCLUSIONS

At long last, it infers that a malware detection methods based on permission usage analysis to adapt to the rapid increase in the number of Android malware can be improved based on different methods. SIGPID method detection can be taken as the best method compared to other methods mentioned in this paper based on the permission detection methods. SIGPID approach helps to identify more malicious apps (with higher recall rate), which is a significant property of an effective malware detector. Future can be done by considering all the advantages and implementing the method.

REFERENCES

1. *02 January 2020 ISSN: 2576-7828 2019 IEEE 19th International Conference on Communication Technology (ICCT) "Android Application Security Detection Method Based on Metropolis Algorithm"*
2. *IEEE International Conference on Computer and Communication Engineering Technology (CCET)" A Two-Layered Malware Detection Model Based on Permission for Android"*
3. *Vol 14 IEEE Transactions on Industrial Informatics, "Significant Permission Identification for Machine-Learning-Based Android Malware Detection".*
4. *2018 IEEE International Conference on Applied System Invention (ICASI), Chiba "Android (Nougats) security issues and solutions"*
5. *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, "Android security issues and solutions," 2017*
6. *Li J, Sun L, Yan Q, et al. Android Malware Detection [J]. IEEE Transactions on Industrial Informatics, 2018, PP (99):1-1.*
7. *Bartel A. Automatically securing permission-based software by reducing the attack surface: an application to Android[C]// International Conference on Automated Software Engineering. IEEE, 2012:274-277.*

8. *W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information flow tracking system for real-time privacy monitoring on smart phones," ACM Transactions on Computer Systems (TOCS), Vol. 32, No. 2, p. 5, 2014.*

9. *C. Yang, Z. Xu, G. Gu, V. Yegneswaran, and P. Porras, "Droidminer: Automated mining and characterization of fine-grained malicious behaviors in android applications," in European Symposium on Research in Computer Security. Springer, 2014, pp. 163–182.*

10. *D. Geneiatakis, I. N. Fovino, I. Kounelis, P. Stirparo, A permission verification approach for android mobile applications, Computers & Security 49 (2015) 192–205.*

11. *L. Li, T. F. Bissyand, M. Papadakis, S. Rasthofer, A. Bartel, D. Octeau, J. Klein, T. Le, Static analysis of android Apps: A systematic literature review, Information & Software Technology (2016) pgs. 67–95.*

12. *J. Li, L. Sun, Q. Yan, Z. Li, W. Srisaan, H. Ye, Significant permission identification for machine-learning-based android malware detection, IEEE Transactions on Industrial Informatics 14 (2018)3216–3225.*