# ANDROID APP FOR AUTOMATIC MALWARE DETECTION VIA REVIEWS AND PERMISSIONS

Pramila Kumari[1], Gori Shanker[2]
Dept. of Computer Science & Engineering. St. Margret Engineering Collage, Neemrana, Rajasthan, India

***Abstract: In this paper a new technique is stated True App which will be used in the Automatic Malware Detection. "True" is that the truth, a word that it self depicts that it don't lies. "App" is any applicatio n; here it's a mechanical man application. In our app we have a tendency to be serving to our users in police investigation any harmful application. The users don't seem to be tuned in to the intensions of the developers WHO somewhere aiming to fetch your necessary files, deleting your knowledge, etc." tru e app" can facilitate these sort of users in police investigation any app that is already put in in there mechanical man phones or are aiming to install any new app in their phones. All the knowledge of the put in apps is saved on a server. The knowledge fetched from the server is going to be within the type of: package details, version range, installation and last changed dates, permission supported these permissions the malware ratings are given to every application. If user attempting to put in any new app then an alert message relating to the confirmation can get displayed and at that time the knowledge are going to be accessed from the server.***

***Keywords: android, malware, mobile operating system***

## I. INTRODUCTION

Being a mobile operating system, android OS is a modified version of Linux, originally developed by a start-up, Android, Inc. As Google entered mobile market, it purchased Android and in a bid to encourage independent development works, it released the developer tools under the open source Apache License. The permissive licensing allows the OS and related software to be modified and distributed by enthusiastic developers, network operators & device manufacturers.


Fig. 1 Android Architecture

## II. DESIGN OF PROPOSED WORK

Working of Malware Detection

In order to solve the problem related to the malware detection we have adopted two policies.

- Permission Based
- Review Based

Permission Based: To keep up security for the system and clients, mechanical man needs application to ask for permission before the applying will utilize bound system data and alternatives. To secure the system's honesty and hence the client's protection, mechanical man runs each application in an exceptionally limited access sandbox. In the event that the applying needs to utilize resources or information outside of its sandbox, the applying must explicitly ask for permission. looking on the sort of permission the applying demands, the system may concede the permission mechanically, or the system may raise the client to give the permission

Algorithmis based on Permission Counts

Step 1: Store all the permission DPERMISSIONS array.

Step 2: Access all permissions mentioned in Android Manifest File store in APER array.

Step 3: Store count of permissions in COUNT.

Step 4: Set DCOUNT=0 (for counting the number of dangerous permission allotted).

Step 5: FOR I= 1 to APER.LENGTH

Step 6: FOR J=1 TO DPERMISSIONS.LENGTH

Step 7: IF APER[I]==DPERMISSIONS[I] THEN

SET DCOUNT=DCOUNT+1

[End of Inner For Loop]

[End of Outer For Loop]

Step 8: IF DCOUNT<=3 THEN

Ok Application Use it safely

ELSE IF DCOUNT<=5 THEN

Potentially Dangerous

ELSE

Very Dangerous

[End of If structure]

Step 9: Exit.

Review Based:

Second Section of our proposed work relies on Reviews in which we have a concept of getting the reviews related on the application from the users. And the users can openly submit the reviews and on the basis of the reviews submitted by the users the classifications will be done.

In this we have taken up a hosting space on the server, we have written the scripts for submission of the reviews, listing of all reviews and searching the reviews regarding the application.

According the concept which we have mentioned about we have the following algorithms for the performing the

mentioned tasks.

Algorithm for Submission of Review

Step 1: Read all the application regarding details from the user.

Step 2: Mention what you find regarding the application and reason for its functionality as malware.

Step 3: User submit the review, the script of storing the review will grab the details and interact with the server database and store all the information on the server database so that it can be accessed anywhere.

Algorithm for Searching for Review

Step 1: Access the SD Card or Memory card attached with the android device.

Step 2: For the list of application stored on the SD Card, select the name of applications whose review which we want to search on server.

Step 3: User submit the application name , the script of searching the review will grab the application name and interact with the server database and will all the review related information from the server database.

Algorithm for Listing All Review

Step 1: The script of listing all the review will interact with the server database and list all the information from the server database so that it can be accessed anywhere.

Working of Malware Detection

In this chapter we will discuss the implementation part of the application containing the algorithm which we have proposed. In our proposed the implementation which we have created is via making use of the integrated development environment Eclipse Indigo edition. In this we have used Android as the main programming concept and for the server based implementation we have made use of the PHP and MYSQL.

## III. EXPERIMENTAL RESULTS AND ANALYSIS

Simulation Results

The simulation results shows the behavior and permission analysis of the application and this simulation is carried on the number of applications installed on the mobile phone and the result achieved are shown.

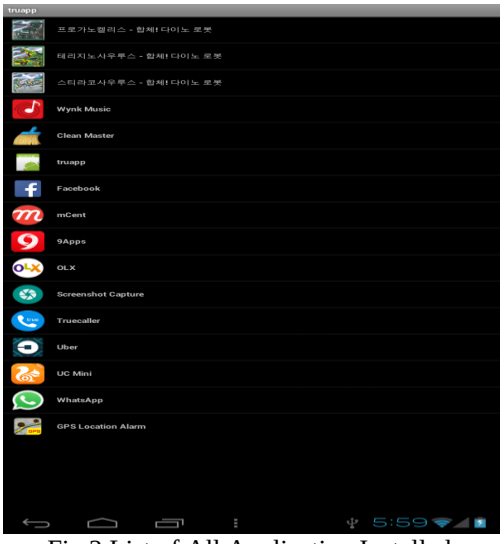



Fig 2 List of All Application Installed


Fig 3 Rating of WhatsApp App


Fig 4 Malware Rating of True Caller App

PROPOSED EXPERIMENT PERFORMANCE IN DIFFERENT CASES

Case I: Malware Detection for Group I of Social Networking Applications using Permission Control

In this case, we have analyzed the applications related to the Social Networking and the result of some applications we have listed in the table.

Table 1 Social Networking Permission Table List

| Serial # | Application Name | Dangerous Permission Count | Malware Behavior |
|---|---|---|---|
| 1 | WhatsApp | 9 | Yes |
| 2 | FaceBook | 5 | Yes |
| 3 | Instagram | 7 | Yes |
| 4 | QQ | 5 | Yes |
| 5 | WeChat | 3 | Yes |
| 6 | Ozone | 2 | No |
| 7 | Tumbler | 5 | Yes |
| 8 | Baidu Tieba | 11 | Yes |

Case II: Malware Detection for Group II of Games Applications using Permission Control
In this case, we have analyzed the applications related to the Social Networking and the result of some applications we have listed in the table.

Table 2 Games Permission Table List

| Serial # | Application Name | Dangerous Permission Count | Malware Behavior |
|---|---|---|---|
| 1 | Chef Judy | 14 | Yes |
| 2 | Fashion Judy | 11 | Yes |
| 3 | Ben 10 | 2 | No |
| 4 | FIFA | 5 | Yes |
| 5 | FlappyCat | 7 | Yes |
| 6 | HummmingWhale | 7 | Yes |
| 7 | Angry Birds Rio | 2 | No |

Case III: Malware Detection for Group I of Social Networking Applications using Opinion Analysis
In this case, we have analyzed the applications related to the Social Networking and the result of some applications we have listed in the table.

Table 3 Social Networking Review Table List (Review Threshold is 10)

| Serial # | Application Name | Reviews Negative | Malware Behavior |
|---|---|---|---|
| 1 | WhatsApp | 3 | No |
| 2 | FaceBook | 1 | No |
| 3 | Instagram | 3 | No |
| 4 | QQ | 4 | No |
| 5 | WeChat | 10 | Yes |
| 6 | Ozone | 5 | No |
| 7 | Tumbler | 9 | No |
| 8 | Baidu Tieba | 4 | No |

Case IV: Malware Detection for Group II of Games Applications Opinion Analysis
In this case, we have analyzed the applications related to the Social Networking the result of some applications we have listed in the table.

Table 4 Games Review Table List (Review Threshold is 10)

| Serial # | Application Name | Reviews Negative | Malware Behavior |
|---|---|---|---|
| 1 | Chef Judy | 7 | No |
| 2 | Fashion Judy | 10 | Yes |
| 3 | Ben 10 | 14 | No |
| 4 | FIFA | 10 | Yes |
| 5 | FlappyCat | 20 | Yes |
| 6 | HummmingWhale | 21 | Yes |
| 7 | Angry Birds Rio | 6 | No |

Case V: Malware Detection for Group I of Social Networking Applications using Opinion Analysis and Permission Control
In this case, we have analyzed the applications related to the Social Networking and the result of some applications we have listed in the table.

Table 5: Social Networking Review and Permission Table List (Threshold count is 10)

| Serial # | Application Name | Reviews Negative | Dangerous Permission Count | Malware Behavior |
|---|---|---|---|---|
| 1 | WhatsApp | 3 | 9 | Yes |
| 2 | FaceBook | 1 | 5 | Yes |
| 3 | Instagram | 3 | 7 | Yes |
| 4 | QQ | 4 | 5 | Yes |
| 5 | WeChat | 10 | 3 | Yes |
| 6 | Ozone | 5 | 2 | No |
| 7 | Tumbler | 9 | 5 | Yes |
| 8 | Baidu Tieba | 4 | 11 | Yes |

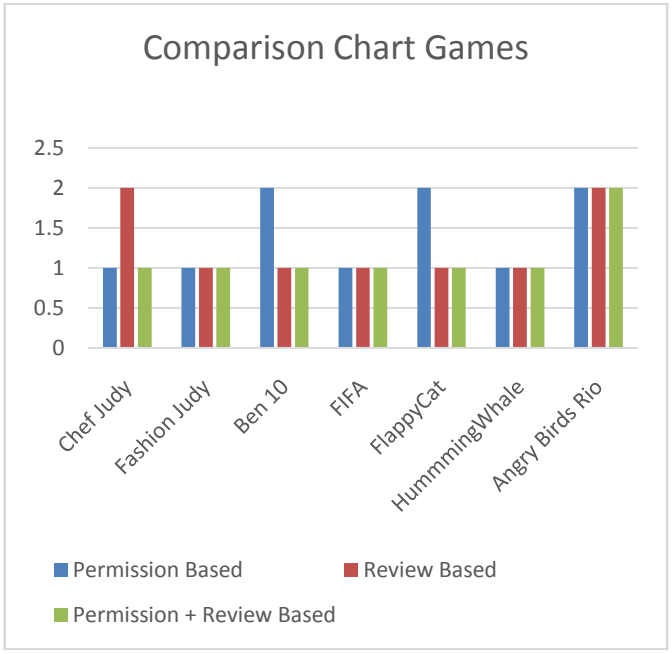
Case VI: Malware Detection for Group II of Games Applications Opinion Analysis and Permission control
In this case, we have analyzed the applications related to the Social Networking and the count of such applications is around 44 and the result of some applications we have listed in the table.

Table 6 Games Review and Permission Table List (Threshold count is 10)

| Serial # | Application Name | Reviews Negative | Dangerous Permission Count | Malware Behavior |
|---|---|---|---|---|
| 1 | Chef Judy | 7 | 14 | Yes |
| 2 | Fashion Judy | 10 | 11 | Yes |
| 3 | Ben 10 | 14 | 2 | Yes |
| 4 | FIFA | 10 | 5 | Yes |
| 5 | FlappyCat | 20 | 7 | Yes |
| 6 | Hummming Whale | 21 | 7 | Yes |
| 7 | Angry Birds Rio | 6 | 2 | No |

EXPERIMENTAL RESULTS COMPARISON



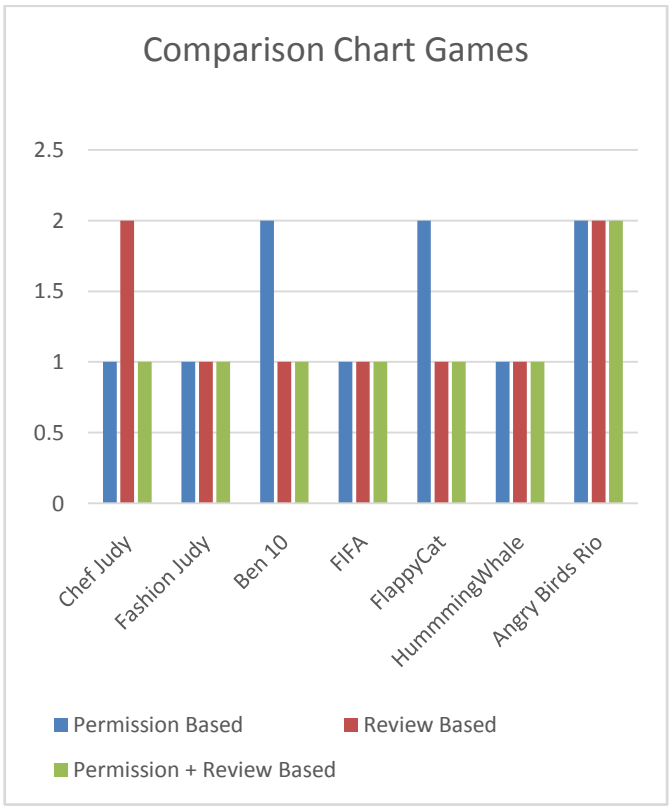




Fig 5 Comparison Graph for Games

Fig 5 shows the comparison graph for the games applications and the comparison in the graph is on basis of permission analysis, review analysis and also on the combined analysis. The value 2 in the graph shows that the application is normal app and the value 1 in graph shows it is malware app.

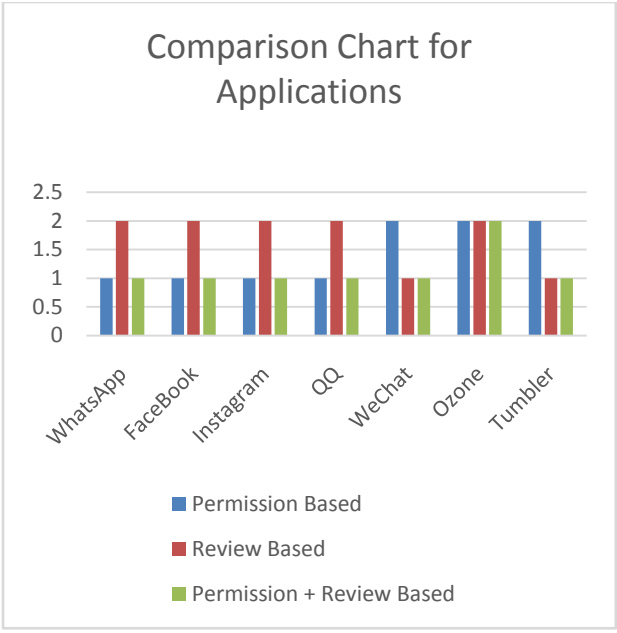



Fig 6 Comparison Graph for Applications

Fig 6 shows the comparison graph for the games applications and the comparison in the graph is on basis of permission analysis, review analysis and also on the combined analysis. The value 2 in the graph shows that the application is normal app and the value 1 in graph shows it is malware app.

## IV. IMPORTANCE AND RELEVANCE OF THE STUDY

There are many definitions of malicious software, malicious code, and malicious content, often called malware. Two similar definitions of malicious code and malicious software that are feasible for this thesis are noted below.

1. Malicious code is defined as:
Programming code that is capable of causing harm to availability, integrity of code or data, or confidentiality in a computing system encompasses Trojan horses, viruses, worms, and trapdoors.

2. Malware is defined as:
Malware is a set of instructions that run on your computer and make your system do something that an attacker wants it to do. In the digital world developing and distributing malware is of interest to individuals and organizations with unethical or illegal intentions.

A few examples of malware behaviour are to:

- Delete crucial files on a computer to render it unusable without a recovery process.
- Log every keyboard input to see what the user's t ype.
- Steal personal or sensitive information or files from a computer.
- Use a computer's resources for the purpose of the malware, e.g. send spam emails,

DdoS another system, or brute-force encryption keys. Often, the malware author will want to maximize the spread of the

malware and will look to implement a mechanism where his software can copy itself to other, similar devices. Malware provides no legal notice to the affected user. This threat includes Trojans, worms and viruses. The implementation of malware detection systems in mobile devices is also a relatively new plan. Security tools and mechanisms used in computers don\'t appear to be doable for applying on smart phones as a results of the excessive resource consumption and battery depletion. Hence, we've got an inclination to determined to perform the entire analysis technique on an obsessive remote server. This server area unit about to be used exclusively to collect information and observe malicious and suspicious applications at intervals the golem platform.
Their framework consists of the many parts which supply enough resources and mechanisms to look at malware on the golem platform. First, we have a tendency to Have developed a light-weight shopper referred to as Crowdroid, which can be downloaded and place in from Google\'s Market. This application is answerable of look UNIX Kernel system calls and deed them preprocessed to a centralized server. In line with a crowd sourcing philosophy, users will facilitate with deed non-personal, but behavior-related data of each application they use. Another Paper is "A Survey on machine-driven Dynami c Malware Analysis Techniques and Tools". This survey article provides a top level view of techniques that area unit supported dynamic analysis that area unit accustomed analyze probably Malicious samples. It to boot covers analysis programs that use these techniques to assist a person\'s analyst in assessing, throughout a timely and applicable manner, whether or not or not a given sample deserves nearer manual review as a results of its unknown malicious behavior. This article focuses on the techniques which can be applied to research potential threats, and discriminate samples that unit mere variations of already known threats. To boot, it presents the presently offered tools and their underlying approaches to perform machine-driven dynamic analysis on likely malicious code.

Analysis:
Smartphone platforms have become a lot of and a lot of widespread lately. To shield sensitive resources within the smart phones, permission-based isolation mechanism is employed by fashionable Smartphone systems to stop untrusted apps from unauthorized accesses. In Android, Associate in nursing application has to expressly request a collection of permissions once it's put in. However, when permissions square measure granted to Associate in Nursing application, there's no thanks to examine and prohibit however these permissions square measure utilized by the app to utilize sensitive resources. Whereas these malware apps square measure clear examples containing undesirable behaviors, sadly even in purportedly benign applications, there might even be several hidden undesirable behaviors like privacy invasion. This paper can provide a systematic approach to malware analysis. A study of malware and dynamic malware analysis are going to be performed. I will be able to attempt to propose a brand new framework supported recording equipment approach to get and analyze golem application's activity. It'll be capable of identif ying between benign and malicious application.

REQUIREMENT SPECIFICATION

Hardware

- Any Android Enabled Handheld
- Android OS Version : Gingerbread & Above
- Google APIs 9

Software

- Server Side : Database Server – MySQL 4.1 or higher
- Server Side : PHP – PHP 4.4.0 or higher (5.2 recommended)
- Server Side: JSON
- Client End : Network Enabled system with Eclipse IDE and ADT Plug-in (for emulator use & debugger)

## V. CONCLUSION

Malware can be perceived as the tool or the weapon of an individual or organization intending an unethical or illegal act concerning computers and data. Such applications of malware detection are always required for protection of our device.

## VI. REFERENCE

[1] Zhongyuan Qin, YuqingXu, Yuxing Di, Qunfang Zhang and Jie Huang, "Android malware detection based on permission and behavior analysis," International Conference on Cyberspace Technology (CCT 2014), Beijing, 2014, pp. 1-4.

[2] U. Pehlivan, N. Baltaci, C. Acartürk and N. Baykal, "The analysis of feature selection methods and classification algorithms in permission based Android malware detection," 2014 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Orlando, FL, 2014, pp. 1-8.

[3] S. Liang and X. Du, "Permission-combination-based scheme for Android mobile malware detection," 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, 2014, pp. 2301-2306.

[4] X. Liu and J. Liu, "A Two-Layered Permission-Based Android Malware Detection Scheme," 2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, Oxford, 2014, pp. 142-148.

[5] Zhao Xiaoyan, Fang Juan and Wang Xiujuan, "Android malware detection based on permissions," 2014 International Conference on Information and Communications Technologies (ICT 2014), Nanjing, China, 2014, pp. 1-5.

[6] S. Hou, T. Lu, Y. Du and J. Guo, "Static detection of Android malware based on improved random forest algorithm," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 2017, pp. 200-200.

[7] M. Y. Su, J. Y. Chang and K. T. Fung, "Machine learning on merging static and dynamic features to identify malicious mobile apps," 2017 Ninth

International Conference on Ubiquitous and Future Networks (ICUFN), Milan, 2017, pp. 863-867.

[8] L. Yu; X. Luo; C. Qian; S. Wang; H. K. N. Leung, "Enhancing the Description-to-Behavior Fidelity in Android Apps with Privacy Policy," in IEEE Transactions on Software Engineering ,vol.PP, no.99, pp.1-1

[9] B. Shrestha, D. Ma, Y. Zhu, H. Li and N. Saxena, "Tap-Wave-Rub: Lightweight Human Interaction Approach to Curb Emerging Smartphone Malware.