UNIVERSITI PUTRA MALAYSIA

PERMISSION EXTRACTION FRAMEWORK FOR ANDROID MALWARE DETECTION

ALI GHASEMPOUR

FSKTM 2019 24

PERMISSION EXTRACTION FRAMEWORK FOR ANDROID MALWARE DETECTION

By

ALI GHASEMPOUR

Thesis submitted to Faculty of Computer Science and Information Technology, University Putra Malaysia, in fulfillment of the requirements for the Master of Information Security

JULY 2019

DEDICATIONS

This thesis is dedicated to:

My great parents who never stop giving of themselves in countless ways,

My beloved friends, who encourage and support me.

Abstract of thesis presented to the Senate of University Putra Malaysia in fulfillment of the requirement for the degree of Master of Information Security

PERMISSION EXTRACTION FRAMEWORK FOR ANDROID MALWARE DETECTION

By

ALI GHASEMPOUR

JULY 2019

Supervisor: Assoc. Prof. Dr. Nor Fazlida Mohd Sani

Faculty: Faculty of Computer Science and Information Technology

Nowadays Android base’s devices have more popularity in compare to other platforms. Statistics represent that market share for Android on mobile devices on March 2018 is 84.8% in compare only 15.1% iOS. These numbers indicate that most of attacks are subjected to Android devices. In addition, most of people are keeping their confidential information on their mobile phone therefore securing Android should taking high concern. Recently there are a lot of researches on detecting malicious applications on Android platform. There are two main approaches for investigation over applications, first static analysis and second behavioral or dynamic analysis. Static analysis mostly focusing on Android Packaging File (APK). On the other hand, dynamic analysis, study on application behavior in isolated environment to analyze intention of application. Due to high number of applications, all of mentioned approaches need automation techniques for classification, however current researches could not provide satisfying accuracy while dealing with large number of applications. To clarify, precision, recall and false positive with growing number of dataset has negative relation. Also this point needs to be considered that, using high number of features can leads to less performance in implementation. Permission is one of noticeable features to expose the intention of developer. In this project, proposed static analysis method for detecting malicious application. The framework is hiring permission extraction approach to label malicious applications by analyzing permissions. Different statistical methods have been used to optimal distinguish malicious and benign applications. Machine learning is used for classification and detection. While increasing input data, model tries to keep detection accuracy in acceptable level. Outcome of proposed framework shows with almost 60,000 number of applications, 94.00% f-score is achievable.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
Sebagai memenuhi keperluan untuk Ijazah Sarjana Keselamatan Makumat

PERMISSION EXTRACTION FRAMEWORK FOR ANDROID MALWARE DETECTION

Oleh

ALI GHASEMPOUR

JULY 2019

Penyelia: Assoc. Prof. Dr. Nor Fazlida Mohd Sani
Fakulti: Fakulti Sains Komputer dan Teknologi Maklumat

Kemajuan teknologi peranti asas Android semakin terkenal pada masa kini berbanding penggunaan sistem operasi lain. Statistik menujukan kemajuan pasaran untuk peranti mudah alih Android pada bulan Mac tahun 2018 adalah 84.8% dibandingkan iOS hanya 15.1%.
Hasil daripada nombor peratus ini membawa maksud bahawa kebanyakan serangan adalah tertakluk kepada peranti Android. Di samping itu, majoriti pengguna akan menyimpan maklumat rahsia dan privasi mereka di dalam peranti mudah alih. Oleh itu untuk mengurangkan lebih banyak kesan buruk sistem operasi ini harus di beri perhatian yang tinggi. Baru - baru ini terdapat banyak penyelidikan untuk mengenal pasti aplikasi malware pada sistem operasi Android. Terdapat dua kaedah utama dalam penyelidikan yang di lakukan ke atas aplikasi ini, pertama statik analisis dan kedua behavioural atau dinamik analisis. statik analisis biasanya fokus pada Android Packaging File (APK). Manakala

kaedah yang kedua, dalam dinamik analisis, kajian fokus pada aplikasi yang di asingkan untuk di kaji. Oleh kerana bilangan aplikasi yang tinggi, semua pendekatan yang disebutkan memerlukan teknik automasi untuk klasifikasi, namun penyelidikan semasa tidak dapat memberikan ketepatan yang memuaskan ketika berurusan dengan banyak aplikasi. Untuk memperjelaskan, ketepatan, ingat dan positif palsu dengan semakin banyak kumpulan data mempunyai hubungan negatif. Juga titik ini perlu dipertimbangkan, dengan menggunakan ciri-ciri yang tinggi boleh menyebabkan kurang prestasi dalam pelaksanaan. Kebenaran adalah salah satu ciri yang ketara untuk mendedahkan niat pemaju. Dalam projek ini, cadangan kaedah analisis statik untuk mengesan permohonan berniat jahat. Oleh itu, hasil dapatan kajian ini memberi kebenaran, mengkaji dan melabel aplikasi malware yang dikenal pasti. Statik yang berlainan di gunakan untuk mengoptimumkan dan membezakan aplikasi malware dan aplikasi jinak. Mesin pembelajaran digunakan untuk mengklasifikasi dan mengesan malware serta bagi meningkatkan pemasukan data dan untuk memastikan ketepatan pengesanan dalam tahap yang boleh di terima. Hasil rangka kerja yang dicadangkan menunjukkan dengan hampir 60,000 bilangan aplikasi, 94.00% f-score dapat dicapai.

ACKNOWLEDGMENT

During this period, I am able to learn so many new things especially related to security in Information Technology. First of all, I would like to dedicate this appreciation to my parent Mohammadebrahim and Leila who always give me endless moral support and encouragement. My appreciation also goes to my supervisor, Assoc. Prof. Dr. Nor Fazlida Mohd Sani who always patient in helping me and giving guideline for improvements. I am thankful toward her effort and sharing of knowledge to support me until the end of this project. Not to be forgotten, thanks to my fellow friends who are not tired of giving opinion and constructive comments that able to improve my project.

APPROVAL FORM

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfillment of the requirement for degree of Master Information Security. The members of the Supervisory Committee were as follows:

ASSOC. PROF. DR. NOR FAZLIDA BT MOHD SANI

Faculty of Computer Science and Information Technology

Universiti Putra Malaysia

(Supervisor)

Data: July 2019

DECLARATION FORM

Declaration by graduate student

I hereby confirm that:

- This thesis is my original work
- Quotations, illustrations and citations have been duly referenced
- This thesis has not been submitted previously or concurrently for any other degree at any other institutions
- Intellectual property from the thesis and copyright of thesis are fully owned by Universiti Putra Malaysia (UPM)
- Written permission must be obtained from supervisor and Deputy Vice-Chancellor (Research and Innovation) before thesis is published in book form.
- There is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity was unhelp as according to Rule 59 in Rules 2003 (Revision 2013-2014). The thesis has undergone plagiarism detection software.

Signature: ________________________ Date: ________________________

Name and Matric No: ALI GHASEMPOUR (GS50931)

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 RESEARCH BACKGROUND

In recent decade, Android is most famous platform for mobile devices (Li et al., 2018). Saving and storing confidential data such as banking information or contact numbers is part of every mobile devices. Hence providing barrier between information and attacker is highly suggested. Nowadays antiviruses are greatly developed and provide wide range of option based on user needs, however studies shows most of Android users do not relay on antiviruses to secure their phone (Walls & Choo, 2015). So this can be great motivation for attacker to focus on Android platform. Kaspersky which is one of biggest security solution company released information related to malicious activity on Android platform on 2018. There are 5,730,916 malicious packages were detected by their lab (Roman Unuchek, 2018). So protecting Android devices from misuse or any malicious application is one of the necessity on this field.

Mobile malware are generally can be classified into three sections: (Dua & Bansal, 2014)

1. Malware: focus on gaining access to personal data or damaging to hardware. Unprivileged access to user personal information is one of main purpose of attack. Malware itself can be divided into following attacks:
    a. SMS attack: With SMS which mostly related to phishing and adware.
    b. Bluetooth attack: Through Bluetooth attacker can steal user personal information or location services.
    c. GPS attack: can compromise GPS device on mobile phone to steal user location.
    d. Root access: within this attack, attacker can get privileged access to phone operating system to install or remove application.
2. Grayware: which mostly focusing on marketing side without any damaging to phone or user data.
3. Spyware: is most frequent type of attack related to mobile phone which stealing user data and send it to unwanted application rather than original one.

Moreover, Android malware can be specifically for can categorized into Worm, Trojan, Back-doors, Botnet, Spyware and Ransomware (Zachariah, Yousef, & Chacko, 2017).

Different researches had been done within last decade to analyze behavior of application in Android platform. To overcome this issues two main approaches suggested. Static based and dynamic based (Zachariah et al., 2017).

Static analysis mostly applicable for analyzing application by disassemble package and extracting source. Some researches focused on finding sequence of op-code which reflect malicious activity. The reason which op-code can be named as feature is, every instruction in computer program follow specific structure. In this structure op-code determine action of instruction and rest are address in memory. Malicious packages may follow same action in instruction so op-code can be good classifier for detection. Permission and source code analysis are other example of static analysis.

On the other hand, dynamic analysis focus on application behavior during run time. As an example requested permissions while application is running can determine application intend. Additionally system call is one of main feature which can be measure for application behavior in runtime. To address why system call can be named as dynamic analysis, Android operating system is based on Linux. System call is provided by application to request specific resources whether hardware or library from Linux kernel. Requesting irrelevant resource for application can be mark as suspicious activity. (Wei, Li, Roy, Ou, & Zhou, 2017)

Other than that, permission is famous feature for android malware detection. Permission exposes exact intent of application. Permission is designed to protect user privacy on Android platform. Permission is requested by application to authorize itself to access specific hardware or software resource. Mentioned resources are sensitive user data (such as SMS or contacts) or system features (such as camera or internet). Based on application nature, system may grant permission or ask user to grant it. Basically no application has authorization to access or read other application data, system files and user private data (such as SMS). Details related to permission are discussed in chapter three. (Google, 2019)

Both static and dynamic analysis are signature based analysis. To overlook on different analysis methods, signature based and anomaly based analysis can be named. In signature based, specific or ensemble of features for detecting malicious application are focused however on anomaly based there is not any specific feature to differentiate applications. In this method, behavior of application in general and over period of time is studied. Data analysis and pattern recognition technique are used for blind search on application intent. One of example is study network traffic for period of time. This method will be similar to network intrusion detection system.

To conclude for different methods for analyzing Android application Figure 1.1 illustrates abstract on how Android malware detection recently categorized.

After feature selection, next phase is design model for using in real world input sample data. This model need to modified and arranged into our requested aim for detection. Well-structured model can improve detection accuracy. In this phase different statistical and mathematical techniques have been used to prune unnecessary features from input data as well as highlight selected features. In this thesis we focusing on permission as input feature. As far as we have only one feature to analyze therefore our model mostly focused on binary classification. The proposed model using different statistical technique to boost computation while keep accuracy high. We adopt permission support and permission ranking in early steps of model. Principal component analysis is used to find different view of permission correlation. Dimensionally reduction using Kaiser and Cumulative techniques implemented to select most significant permissions. Selected permissions are compared to Li (Li et al., 2018) results to select similar permission. Also separate analysis has been done on our method without any overlap with Li results.

Last phase of this project is focusing on classification. Classification could be done by different methods such as machine learning, deep learning or statistical models. Although most of techniques are based on mathematics but terms are different.

Machine learning techniques is one of basis and famous technique. Machine learning is general term and contains many different algorithms inside. Main idea is make model based on known input data to predict unseen input data. Machine trying to predict as precious as possible for unknown input data which calls test data. Most of algorithm follows two steps, train and test. In train phase, machine try to make relevant model based on labeled input data and secondly in test phase machine will predict based on its knowledge on pervious learning. Moreover machine may work on unforeseen situation where the train phase does not exists. Therefore machine try to find pattern to distinguish between objects.

Two main machine learning techniques are classification and clustering: (Alpaydin, 2010)

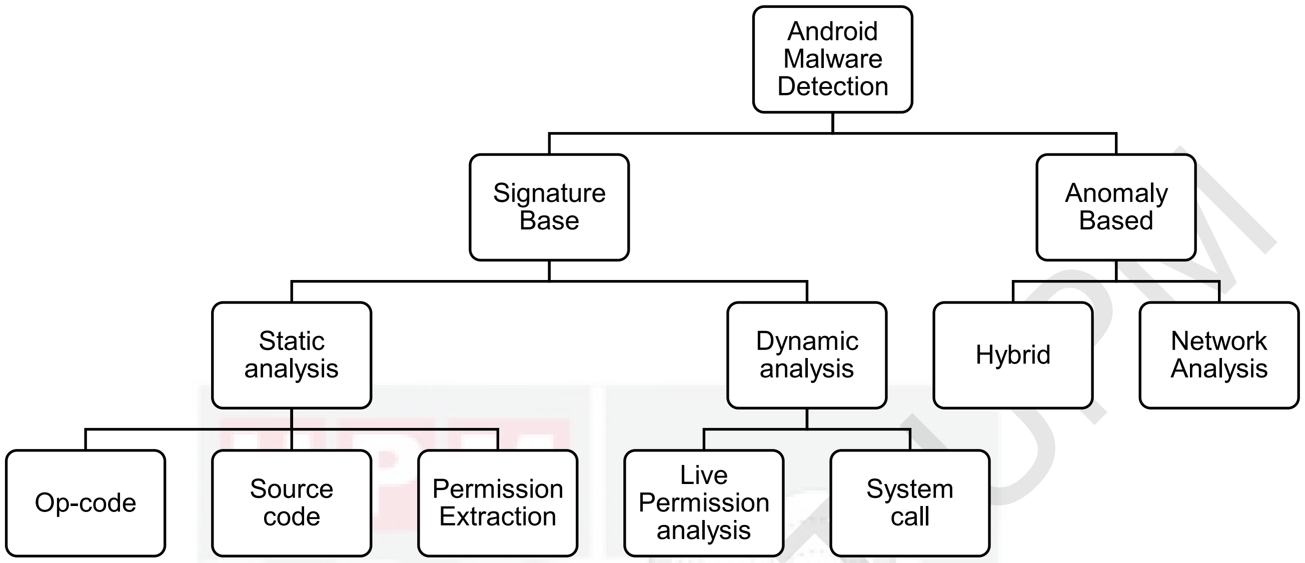

FIGURE 1.1: OVERVIEW OF DIFFERENT ANDROID MALWARE DETECTION METHODS

1. Classification or supervised learning is using labeled data for classifying. In classification problems, features need to be defined by operator. Results are describe mapping of input to output by rules that provided by supervisor.
2. Clustering or unsupervised learning focusing on finding pattern in unlabeled or not fully labeled data. Results mostly are grouping of input data. In clustering problems, machine try to group data with same features instead of classify them. This method in statistic calls density estimation.

Despite of different methods for detection phase, machine learning is used for this thesis based on popularity among other related researches. Support vector machine (SVM) and tree based algorithms are mostly concentrated in this thesis.

1.2 PROBLEM STATEMENT

Detecting malicious applications in Android market is becoming very complex procedure. As number of malicious applications as well as type of attacks are increasing, the complexity of feature selection and classification techniques are growing.(Riad & Ke, 2018)

In order to make it easier to deal with different type of attacks, researchers come up with wide variety of solution. These solutions are trying to increase accuracy of features extraction and classification phases, however enormous number of applications in common Android markets has caused that current techniques are not able to handle large amount of data.

The Significant Permission Identification framework as like as other researchers with different attitudes on detecting malware application are facing different problems. One of major issue is, relationship of parameters like precision, recall[1], false positive with growing number of dataset has negative relation. Either current studies with small dataset are facing to 0.5% to 1% false negative rate (Wang et al., 2018a) (Arshad et al., 2018), (As an example: in 100000 data, it misses 1000 applications). As matter of fact, these research cannot be used in real Android market due to high missing rate.

[1] High recall rate returns most of relevant results, in other word, recall determine, accuracy of algorithm to detect malicious applications in whole of training dataset, on the other hand, high precisions returns more relevant results than irrelevant one. (Walters, 2009)

Permission based malware detection with significant permission identification technique (Li et al., 2018) is framework of this project. Un-optimized permission selection leads to achieving high false positive rate while increasing input data.

1.3 RESEARCH OBJECTIVE

The objective of this project is to propose an optimize permission extraction framework to improve Android malware detection using static analysis.

The proposed enhanced framework is to handle large number of applications while keep performance metrics optimized. Mentioned performance metrics are precision and recall besides true positive and false negative. F-score is also desired to measure performance of new frameworks.

1.4 SCOPE OF STUDY

This project is mainly focus on Android malware detection based on permission. Permission is main feature for this research and it is extracted from APK files. For processing data, principal component analysis and other factor selection methods suggested. In classification phase, machine learning algorithms are used. Support vector machine (SVM) and decision tree are selected algorithms. To implement framework, Python 3.7 besides other tools such as Androguard are used. The dataset collected from different resources. List of projects have been used in this thesis are:

1. M0droid project (Damshenas, Dehghantanha, Choo, & Mahmud, 2015) ,
2. Android Botnet project (Mahindru & Singh, 2017)
3. AndroZoo (Allix, Bissyandé, Klein, & Le Traon, 2016)
4. Android PRAGuard Dataset (Maiorca, Ariu, Corona, Aresu, & Giacinto, 2015)
5. Android Malware Dataset (Wei et al., 2017)

1.5 CONTRIBUTION OF PROJECT

This project is trying to propose solution to overcome on real world issues in Android market. The research framework is trying to align performance metrics with high number of input data. Using multi-level permission filtering and statistical techniques to enhance accuracy and performance of model. Mentioned improvement applicable on current Android markets. Also, compare to other related researches, proposed framework shows more promising results. Additionally, proposed method is only using permission as feature which reduce complexity of feature extraction compare to other researches.

1.6 THESIS STRUCTURE

The structure of this thesis consists of six chapters including Introduction, Literature Review, Research methodology, Implementation, Results and Discussion and last chapter is Conclusion.

Chapter one, briefly explain about background and motivation for this research. Besides that, problems statement and objective are this thesis are clearly explained.

Chapter two, is focusing on current researches on Android malware detection. Taxonomy of analysis is formed for ease of reading. Each method briefly explained and advantage of it summarized into table. Also discussion for comparison between methods is provided.

Chapter three, explain research methodology. In this chapter permission extraction framework introduced. All components related to Android packaging (APK) file discussed in detail. Desired machine learning models also identified in the end of this chapter.

Chapter four, includes project implementation details. In this chapter, the approach for coding and designing framework is detailed. Besides that some sample output from every steps illustrated.

Chapter 5, is explaining results. The outcomes of proposed algorithm besides all finding are described. Also comparison between proposed method results with other techniques are discussed.

Chapter 6 is conclusion chapter for this thesis. This chapter summarized the objective and results for this research.

REFERENCES

Agrawal, R., & S&ant, R. (1994). Fast Algorithms for Mining Association Rules. *20th VLDB Conference Santiago, Chile,*. Retrieved from http://www.vldb.org/conf/1994/P487.PDF

Allix, K., Bissyandé, T. F., Klein, J., & Le Traon, Y. (2016). AndroZoo. In *Proceedings of the 13th International Workshop on Mining Software Repositories - MSR '16* (pp. 468–471). New York, New York, USA: ACM Press. https://doi.org/10.1145/2901739.2903508

Alpaydin, E. (2010). *Introduction to machine learning*. MIT Press.

Anthony Desnos, S. H. (2011). Androguard. Retrieved from https://github.com/androguard/

Arshad, S., Shah, M. A., Wahid, A., Mehmood, A., Song, H., Member, S., & Yu, H. (2018). SAMADroid : A Novel 3-Level Hybrid Malware Detection Model for Android Operating System, 4321–4339.

Badhani, S., & Muttoo, S. K. (2019). CENDroid—A cluster–ensemble classifier for detecting malicious Android applications. *Computers & Security*. https://doi.org/10.1016/J.COSE.2019.04.004

Banchs, R. (2013). *Text Mining with MATLAB*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-4151-9

Damshenas, M., Dehghantanha, A., Choo, K.-K. R., & Mahmud, R. (2015). M0Droid: An Android Behavioral-Based Malware Detection Model. *Journal* *of Information Privacy and Security*, *11*(3), 141–157. https://doi.org/10.1080/15536548.2015.1073510

Dua, L., & Bansal, D. (2014). TAXONOMY: MOBILE MALWARE THREATS AND DETECTION TECHNIQUES, 213–221. https://doi.org/10.5121/csit.2014.4522

Feng, P., Ma, J., Sun, C., & Ma, Y. (2018). A Novel Dynamic Android Malware Detection System With Ensemble Learning. *IEEE Access*, *6*(c), 30996–31011. https://doi.org/10.1109/ACCESS.2018.2844349

Gary, Chen, B., Wu, Annie, S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. In *Proceedings of the 43rd annual southeast regional conference on - ACM-SE 43* (Vol. 2, p. 136). New York, New York, USA: ACM Press. https://doi.org/10.1145/1167253.1167288

Google. (2010). The Structure of Android Package (APK) Files.

Google. (2019). Permissions overview | Android Developers. Retrieved June 1, 2019, from https://developer.android.com/guide/topics/permissions/overview

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features (pp. 137–142). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026683

Karbab, E. M. B., Debbabi, M., & Mouheb, D. (2016). Fingerprinting Android packaging: Generating DNAs for malware detection. *Digital Investigation*, *18*, S33–S45. https://doi.org/10.1016/j.diin.2016.04.013

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, J., Sun, L., Yan, Q., Li, Z., Srisa-an, W., & Ye, H. (2018). Significant Permission Identification for Machine Learning Based Android Malware Detection. *IEEE Transactions on Industrial Informatics*, *3203*(c). https://doi.org/10.1109/TII.2017.2789219

Mahindru, A., & Singh, P. (2017). Dynamic Permissions based Android Malware Detection using Machine Learning Techniques. In *Proceedings of the 10th Innovations in Software Engineering Conference on - ISEC '17* (pp. 202–210). New York, New York, USA: ACM Press. https://doi.org/10.1145/3021460.3021485

Maiorca, D., Ariu, D., Corona, I., Aresu, M., & Giacinto, G. (2015). Stealth attacks: An extended insight into the obfuscation effects on Android malware. *Computers & Security*, *51*, 16–31. https://doi.org/10.1016/j.cose.2015.02.007

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. https://doi.org/10.1371/journal.pone.0194889

McGiff, J., Hatcher, W. G., Nguyen, J., Yu, W., Blasch, E., & Lu, C. (2019). Towards Multimodal Learning for Android Malware Detection. In *2019*

International Conference on Computing, Networking and Communications (ICNC) (pp. 432–436). IEEE. https://doi.org/10.1109/ICCNC.2019.8685502

McLaughlin, N., Doupé, A., Joon Ahn, G., Martinez del Rincon, J., Kang, B., Yerima, S., … Zhao, Z. (2017). Deep Android Malware Detection. *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy - CODASPY '17*, 301–308. https://doi.org/10.1145/3029806.3029823

Microsoft. (2010). US20100031353A1 - Malware Detection Using Code Analysis and Behavior Monitoring - Google Patents. Retrieved April 25, 2018, from https://patents.google.com/patent/US20100031353A1/en?oq=US20100031353A1

Milosevic, N., Dehghantanha, A., & Choo, K. K. R. (2017). Machine learning aided Android malware classification. *Computers and Electrical Engineering*, *61*, 266–274. https://doi.org/10.1016/j.compeleceng.2017.02.013

Park, M., Han, J., Oh, H., & Lee, K. (2019). Threat Assessment for Android Environment with Connectivity to IoT Devices from the Perspective of Situational Awareness. *Wireless Communications and Mobile Computing*, *2019*, 1–14. https://doi.org/10.1155/2019/5121054

Riad, K., & Ke, L. (2018). RoughDroid: Operative Scheme for Functional Android Malware Detection. *Security and Communication Networks*, *2018*, 1–10. https://doi.org/10.1155/2018/8087303

Roman Unuchek. (2018). Mobile malware evolution 2017. Retrieved from https://securelist.com/mobile-malware-review-2017/84139/

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using comparison data of known factorial structure. *Psychological Assessment*, *24*(2), 282–292. https://doi.org/10.1037/a0025697

Walls, J., & Choo, K.-K. R. (2015). A Review of Free Cloud-Based Anti-Malware Apps for Android. In *2015 IEEE Trustcom/BigDataSE/ISPA* (pp. 1053–1058). IEEE. https://doi.org/10.1109/Trustcom.2015.482

Walters, W. (2009). *Google Scholar search performance: comparative recall and precision*. *Portal: Libraries and the Academy* (Vol. 9).

Wang, W., Gao, Z., Zhao, M., Li, Y., Liu, J., & Zhang, X. (2018a). DroidEnsemble: Detecting Android Malicious Applications with Ensemble of String and Structural Static Features. *IEEE Access*, 1–1. https://doi.org/10.1109/ACCESS.2018.2835654

Wang, W., Gao, Z., Zhao, M., Li, Y., Liu, J., & Zhang, X. (2018b). DroidEnsemble: Detecting Android Malicious Applications with Ensemble of String and Structural Static Features. *IEEE Access*, 1–1. https://doi.org/10.1109/ACCESS.2018.2835654

Wei, F., Li, Y., Roy, S., Ou, X., & Zhou, W. (2017). Deep Ground Truth Analysis of Current Android Malware (pp. 252–276). Springer, Cham. https://doi.org/10.1007/978-3-319-60876-1_12

Williams, B., Brown Andrys Onsman, T., Onsman, A., Brown, T., Andrys Onsman, P., & Ted Brown, P. (2012). Exploratory factor analysis: A five-step guide for novices Recommended Citation) &quot;Exploratory factor analysis: A five-step guide for novices Exploratory factor analysis: A five-step guide for novices. *This Journal Article Is Posted at Research Online*, *8*, 2010–990399. Retrieved from http://ro.ecu.edu.au/jephc/vol8/iss3/1Availableat:http://ro.ecu.edu.au/jephc/vol8/iss3/1

Zachariah, R., Yousef, M. S., & Chacko, A. M. (2017). Android Malware Detection A Survey, (Iccs).

Zhang, J., Qin, Z., Yin, H., Ou, L., & Zhang, K. (2019). A feature-hybrid malware variants detection using CNN based opcode embedding and BPNN based API embedding. *Computers & Security*. https://doi.org/10.1016/J.COSE.2019.04.005