



Android Hybrid Malware Detection Approaches Using Machine Learning Algorithms

B.K.G.P.N. Weerawardhana

Reg. No: MS20901226

M.Sc. in IT

Specialized in Cyber Security

Supervisor: Dr. Lakmal Rupasinghe

October 2021

**Department of Computing
Faculty of Graduate Studies and Research
Sri Lanka Institute of Information Technology**

Table of Contents

Table of Contents.....	2
List of Figures	5
List of Tables	7
Acknowledgements.....	8
Chapter 1 Introduction	11
1.1 Purpose	13
1.2 Problem Definition.....	15
1.3 Research Questions	16
1.4 Proposed Solution.....	17
1.5 Significance of the Solution.....	18
1.6 Tools and Technique	19
Chapter 2 : Android Operating System Overview.....	21
2.1 Android Architecture	21
2.2 Android Application Components.....	22
2.3 Literature Review	23
2.3.1 Introduction	23
2.3.2 Related Works.....	24
2.4 Android Third Party Application Market.....	30
2.5 Landscape on Android Security.....	30
2.5.1 Security on Android	30
2.5.2 Application Sandbox	31
2.5.3 File Access	31
2.5.4 Application Signing.....	32
2.5.5 Dalvik Virtual Machine (DVM) and Memory Management Unit (MMU).....	32
2.5.6 Android Permission System	32
2.5.7 Component Encapsulation and security	33
2.6 Android Security Mechanism in Each android layer	33
Chapter 3 Android Malware Landscape.....	35
3.1 Android Malware Overview	35
3.2 Malware Life Cycle	35
3.3 Android Malware Category.....	36
3.3.1 Adware	37
3.3.2 Backdoor	38
3.3.3 Ransomware	38
	2

3.3.4 File Infector	39
3.3.5 Potential Unwanted Applications (PUA)	39
3.3.6 Riskware	39
3.3.7 Trojan	40
3.3.8 Privilege escalation attacks	40
3.4 Malware Technique	40
3.4.1 Repacking	41
3.4.2 Premium Rate Short Message Service (SMS)	41
3.4.3 Remote Control	42
3.4.4 Update Attack	42
3.4.5 Drive By Downloads	42
3.4.6 Browser Exploit	43
3.4.7 Information leaking	43
3.5 Malware analyzing technology	44
3.6 Malware Detection Technologies	45
3.7 Malware Detection Tools	47
3.7.1 Firewall	48
3.7.2 Intrusion Detection System	49
3.7.3 Antiviruses	49
Chapter 4 Research Methodology	50
4.1 Proposed Hybrid Malware Detection Approach	50
4.2 Data Collection	54
4.3 Static feature extraction	55
4.4 Dynamic Feature Extraction	66
4.5 Static feature vector	69
4.6 Dynamic Feature Vector	71
4.7 Classifiers use for training	71
4.7.1 Naïve Bayes Algorithm	73
4.7.2 Support Vector Machine (SVM)	74
4.7.3 Performance Metrix	75
4.7.4 Confusion metrics	75
Chapter 5 Achievement result and discussion	78
5.1 Dataset	79
5.2 Classification	80
5.3 Classification result with feature selection	81

5.3.1 Static analysis classification	81
5.3.2 Dynamic analysis classification	87
5.3.3 URL analysis classification	91
5.3.4 Achievement	92
Chapter 6 Conclusion	94
Chapter 7 Future Work	97
Chapter 8 References.....	98
Appendix	105
Appendix 1: Python code for static feature extraction.....	105
Appendix 2: Python code for dynamic feature extraction.....	106
Appendix 3: Code for malware classification.....	107
Appendix 4: Code for URLs Prediction	109

List of Figures

Figure 1 2.1 Android Application Architecture	22
Figure 2 3.1 Android malware category.....	37
Figure 3 3.2 Malware Repacking Technology	41
Figure 4 4.1 Decompiling Android Application	55
Figure 5 4.1 Static Analysis Resources	56
Figure 6 4.3 Main Activity Class	57
Figure 7 4.4 Santoku OS connect with Android Virtual Machine	67
Figure 8 4.5 Install Malware application using gdb	68
Figure 9 4.6 Process ID of the malware application	68
Figure 10 4.7 Extract Log files of malware application	68
Figure 11 4.8 Extracted Dynamic behaviors of application	69
Figure 12 4.9 Python program for create static feature vector (Part 1).....	70
Figure 13 4.10 Python program for create static feature vector (part 2).....	71
Figure 14 5.1 classification result using decision tree algorithm.....	82
Figure 15 5.2 classification result using support vector machine algorithm	83
Figure 16 5.3 classification result using naive bayes algorithm.....	83
Figure 17 5.4 logistic regression classification result.....	84
Figure 18 5.5 random forest classification result.....	84
Figure 19 5.5 classification results of static data using a single machine learning algorithm	85
Figure 20 5.6 WEKA correctly and incorrectly classified features using Random Forest classifier algorithm	86
Figure 21 5.7 Weka classification result summary of Random tree	86
Figure 22 5.8 Random Forest classification result using WEKA.....	87
Figure 23 5.9 Limited dynamic features classification result using naive bayes algorithm	88
Figure 24 5.10 Limited dynamic features classification result using Random Forest Algorithm.....	89
Figure 25 5.11 Limited dynamic features classification result using Simple logistic algorithms	89
Figure 26 5.12 Limited dynamic features classification result using J48 algorithm	90
Figure 27 5.16 Static and Dynamic feature classification result for newly application	92
Figure 28 5.15 Achievement results (check unknown URLs).....	93
Figure 29 5.16 Classification results according to insert URLs.....	93
Figure 30 image 1 for static feature extraction	105
Figure 31 image 2 for static feature extraction	105
Figure 32 image 1 for dynamic feature extraction	106
Figure 33 image 2 for dynamic feature extraction	106
Figure 34 image 1 for malware classification	107
Figure 35 image 2 for malware classification	107
Figure 36 image 3 for malware classification	108
Figure 37 image 4 for malware classification	108
Figure 38 image 5 for malware classification	109
Figure 39 image 1 for URLs prediction.....	110

Figure 40 image 2 for URLs prediction..... 110
Figure 41 image 3 for URLs prediction..... 111

List of Tables

Table 1 2.1 Android Security Mechanism in Each Layer	33
Table 2 4.1 Drebian malware Families.....	58
Table 3 4.2 Drebian Malware Families Extracted Features	59

Acknowledgements

Before I start, I must dedicate praise and exaltation to few peoples in my research project.

Firstly, I would like to express, my sincerest appreciation, my thanks and gratitude to my research supervisor and co-supervisor Dr. lakmal rupasinghe who is senior lecture in cyber security at Sri Lanka Institute of Information Technology at Sri Lanka. I sincerely thanks for his providing me to more valuable advice, boundaries, directions, assistance, guidance and blessing to complete my research project during each step. He monitors me in every step and gave me to necessary advice and guidance till end of my project. I bottom of my heart thankful to his support, technical advice, and encouragement in this my research area. Sir, as you enabled me to understand and develop of the subject area you guided me to how to build better good experiment project, sir I will always remember you in my every future steps.

Finally, I want to highly be indebted my parents and my family since they stay will me in during entire project. They are assistance, encourage, support and their expectation had guided me to complete my experiment project. They were major part of my life when completing my project. I want to heartily tankful to their since they allow to complete my experiment successfully.

.

Abstract

Working title: Android Malware analysis using Reverse Engineering, machine learning algorithms and android code analysis, and implement framework detect android malware.

Smart phones are a major part of a life in modern life. Among them android is the most usable mobile operating system. According to IDC corporate report in USA android operating system use 84.5% from market share [3]. currently most mobile attacks [22] happen with android operating system. Most of the attackers use chunks of malware code attached with android application java code to attack devices. The purpose of android malware writes is to get financial benefits; most of the famous type of android malware is ransomware which after executing malicious application on the device The malware will encrypt all the device valuable information of the device. To decrypt all data owners should be pay for decryption key. Due to android openness and free availability of market, android mobile operating system has become major attractive target for Cyber criminals. In this research paper focus issue of mobile application, analyze malware using reverse engineering, static and dynamic malware analysis, Malicious URL analysis and application code analysis of the android application and implement framework using machine learning based on Supervised machine learning approach for detect and classify android malware. static malware analysis based on reverse engineering of application and extracted application features without executing application. This recognizes application information flow, code structure, permissions, network details and static related features. Dynamic analysis examines the dynamic behaviors of the application during run time of the application in a fully controlled virtual environment. comparing both analysis static analysis consists with pattern-based approach; same time dynamic detection approach can be provided additional protecting from malicious application since it consists dynamic behaviors of the application including memory logs, CPU usage, system call logs, etc. Also, used malicious URL analysis to users protect from unawares downloading malware by using untrusted web URLs. Finally, the outcome will be developed platform which will be identified and protected from malware affected functions. Also, this framework will be using both static, dynamic malware analysis and URL analysis technique, and will solution for traditional malware detection tools problems and Final outcome framework called as Hybrid android malware detection [92] [93] system. Application will be based on machine learning algorithms and python programming. This application can protect from both malware codes and functions which functions are previously analyze using reverse engineering [11], machine learning algorithms, android code analysis and traditional malware features. Especially malware functions consisting of both

traditional and newly coming malware features. My experimental result project depicts that based machine learning based android malware classification and my project can be classify unknown applications malware analyzing android application static and dynamic features. In my project primarily based on android applications permissions and all dynamic related features. Also, users can classify their used accessed URLs are malicious or not and can be safe from android attacks.