BORANG PENGESAHAN STATUS TESIS*

JUDUL: ANALYZING THE ANSERVERBOT ANDROID
MALWARE BEHAVIOR THROUGH STATIC
SESI PENGAJIAAN: 2012/2013

Saya, SHAIDATUL NURMAYANTI BINTI YA'ACOB

mengaku membenarkan tesis Projek Sarjana Muda ini disimpan di Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dengan syarat-syarat kegunaan seperti berikut:

1. Tesis dan projek adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan Fakulti Teknologi Maklumat dan Komunikasi dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ** Sila tandakan (/)

_________ SULIT (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

_________ TERHAD (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

_________ TIDAK TERHAD

_________________________	___________________________
(TANDATANGAN PENULIS)	(TANDATANGAN PENYELIA)
Alamat Tetap: ___________________	___________________________
_______________________________	Nama Penyelia
_______________________________	
Tarikh: ________________________	Tarikh: _______________________

CATATAN: * Tesis dimaksudkan sebagai Laporan Projek Sarjana Muda (PSM).
** Jika tesis ini SULIT atau atau TERHAD, sila lampirkan surat daripada pihak berkuasa.

ANALYZING THE ANSERVERBOT ANDROID MALWARE BEHAVIOR THROUGH STATIC

SHAIDATUL NURMAYANTI BINTI YA'ACOB

This report is submitted in partial fulfilment of the requirements for the
Bachelor of Computer Science (Networking)

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
2013

DECLARATION

I hereby declare that this project report entitled

ANALYZING THE ANSERVERBOT ANDROID MALWARE BEHAVIOR THROUGH STATIC

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT: ___________________________ Date: ________________

(SHAIDATUL NURMAYANTI BINTI YA'ACOB)

SUPERVISOR: ___________________________ Date: ________________

(DR. ROBIAH BT YUSOF)

DEDICATION

Dear Allah

Alhamdulillah for everything You giving to me. May my life is full of Your blessing.

Dear Parent

Thank you for your love, doa’, sacrifice and encouragement that give me strength to finosh this study. May Allah guide us, bless us and protect us to be good Muslims.

Dear Teachers and Supervisors

Thank you for all the knowledge and words that is useful for all humanity. Only Allah can repay your kindness.

Dear Siblings

Thank you for your motivation and understanding. May Allah forgive us.

Dear Friends

Thank you for all the knowledge, love, guide and patient. May our friendship last until Jannah.

ACKNOWLEDGEMENTS

With all the success that I acquired in developing of this project, my deepest appreciation and my special thanks to my supervisor, Dr. Robiah binti Yusof for their guidance, knowledge, ideas, and enthusiasm that are very useful. Thank you for always be there when I needed and patience throughout became my supervisor. Thank you also to my project evaluator Prof Dr. Mohd Faizal Bin Abdollah on the developing criticism and sharing knowledge during presentation. Do not forget thanks to En.Zaki Bin Masod and all lecturers who provide guidance during the making of this project.

Special thanks to my parents Mr. Ya'acob Bin Melia and Mrs. Normah Binti Abu Bakar also my siblings for their prayer, advice, guidance and sacrifice for me to complete this project and to finish my degree. Last but not least, to my dearest friend who always stay besides me in every situation that I have faced. May Allah bless all of us.

ABSTRACT

In this highly sophisticated-age, many people have been using Android in their Smartphones. However, the large numbers of Smartphones users are not aware of the breach security laws and malicious attacks. Therefore, Analyzing the Anserverbot Android Malware through Static Behaviors is a static analysis conducted to identify Anserverbot malware behavior in the Android operating system. Static analysis is the process of obtaining the necessary information through the reading of the syntax and coding of some specific programming language. The main objective of this project is to produce a script that can prove the existence of malware in an application that is installed on the Android operating system. This is because the problem faced by consumers is that they are not able to identify the behavior of malware in the Android environment. Among the main reasons is AnserveBot is a type of malware that moves silently behind an application and it remains passive while not getting any instructions from the main server. With the analyzing tools for android malware such ad JD-GUI, Dex2jar, Apktool, Wireshark and Android SDK, it may help a lot to do this project and cerate the script by using NetBeans. Other than to obtain information and analysis results, the script is also expected to facilitate the researchers demonstrate other behaviors that may occur in the future by improve this script.

ABSTRAK

Dizaman serba canggih ini, ramai orang telah menggunakan Android didalam telefon pintar masing-masing. Namun, ramai bilangan pengguna telefon pintar ini tidak sedar akan perkara yang melanggar undang-undang keselamatan dan serangan berniat jahat. Oleh itu, *Analyzing the Anserverbot Android Malware Behaviors through Static* adalah sebuah analisis yang dijalankan secara statik bagi mengenalpasti tingkah laku Anserverbot malware didalam sistem operasi Android. Analisis secara statik ialah proses mendapatkan maklumat yang diperlukan melalui bacaan pada sintaks dan coding daripada beberapa bahasa pengaturcaraan yang tertentu. Objektif utama menjalankan projek ini ialah untuk menghasilkan skrip yang dapat membuktikan wujudnya malware didalam sesebuah aplikasi yang dipasang pada sistem operasi Android. Ini kerana, masalah yang dihadapi oleh pengguna ialah mereka tidak dapat mengenalpasti tingkah laku malware didalam persekitaran Android. Antara sebab utamanya ialah Anservebot adalah sejenis malware yang bergerak secara senyap dibelakang sesebuah aplikasi dan ia kekal pasif selagi tidak mendapat apa-apa arahan dari pelayan utama. Dengan menggunakan peralatan perkakasan seperti JD-GUI, Dex2jar, Apktool, Wireshark dan Android SDK, ia banyak membantu bagi menjalakan projek ini dan dalam menghasilkan skrip menggunakan NetBeans. Selain daripada untuk mendapatkan maklumat dan keputusan analisa, skrip ini juga diharap dapat memudahkan para penyelidik membuktikan lain-lain tingkah laku yang mungkin terjadi pada masa akan datang dengan menambah baik skrip ini.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGES
	DECLARATION	i
	DEDICATION	ii
	ACKNOWLEDGEMENTS	iii
	ABSTRACT	iv
	ABSTRAK	v
	LIST OF TABLES	ix
	LIST OF FIGURES	x
	LIST OF ABBREVIATIONS	xii
CHAPTER 1	**INTRODUCTION**	
	1.1 Project Background	1
	1.2 Research Problem	2
	1.3 Research Question	2
	1.4 Project Objective	3
	1.5 Project Scope	3
	1.6 Research Contribution	3
	1.7 Expected Output	4
	1.8 Report Organization	4
	1.9 Conclusion	5
CHAPTER 2	**LITERATURE REVIEW**	
	2.1 Introduction	6
	2.2 What is Android?	7
	2.2.1 Android Package and Code files	7
	2.2.2 Android Application	8
	2.2.3 Android Version	11
	2.3 Ice Cream Sandwich	17
	2.4 What is Android Malware?	19
	2.4.1 Current Android Malware	25
	2.5 About AnserverBot Trojan	28
	2.5.1 AnserverBot Key Technique	29
	2.5.2 How It Works	31
	2.6 Malware Detection Techniques	33
	2.6.1 Static Analysis	34

	2.7 Tools of Malware Analysis	35
	2.7.1 Data Analysis Technique	36
	2.8 Parameter Used	37
	2.9 Related Works	38
	2.10 Conclusion	41
CHAPTER 3	**METHODOLOGY**	
	3.1 Introduction	3
	3.2 Methodology	42
	3.2.1 Enhance Methodology	44
	3.3 Milestone	50
	3.4 Conclusion	51
CHAPTER 4	**DESIGN AND IMPLEMENTATION**	
	4.1 Introduction	52
	4.2 Design	53
	4.2.1 Logical Design	53
	4.2.2 Android Malware Attack Model	54
	4.2.3 Scripting Design	55
	4.3 Hardware and Software	59
	4.3.1 Software Configuration Management	63
	4.4 Implementation	65
	4.4.1 Setup Isolated Environment	65
	4.4.2 Extracting AnserverBot Android Malware	68
	4.4.3 Installing Android Application	73
	4.4.4 Using a Wireshark	75
	4.5 Conclusion	77
CHAPTER 5	**TESTING AND ANALYSIS**	
	5.1 Introduction	78
	5.2 Test Plan	79
	5.2.1 Test Organization	79
	5.2.2 Test Environment	79
	5.2.3 Test Schedule	80
	5.3 Test Strategy	84
	5.3.1 Classes of Test	84
	5.4 Test Design.	84
	5.4.1 Test Description	85
	5.4.2 Test Data	85
	5.5 Test Result	86
	5.5.1 Test Verification Result	88
	5.6 Conclusion	91

CHAPTER 6 **CONCLUSION**
6.1 Introduction 92
6.2 Research Summarization 93
6.3 Limitation 93
6.4 Contribution 94
6.5 Future Works 95
6.6 Conclusion 95

REFERENCE 96
APPENDIX A 100
APPENDIX B 102
APPENDIX C 107
APPENDIX D 112

LIST OF TABLES

TABLE	TITLE	PAGES
Table 1.1	Research Problem	2
Table 1.2	Research Question	2
Table 1.3	Research Objective	3
Table 1.4	Research Contribution	3
Table 2.1	History of Android Malware	20
Table 2.2	Behavioral Classification overview for Mobile Malware in the wild	23
Table 2.3	Legend	25
Table 2.4	Current Android Malware	25
Table 2.5	Analysis for Android malware features	27
Table 2.6	Parameter used	37
Table 3.1	Milestone	50
Table 4.1	Android Device Specification	54
Table 4.2	Flow Chart Legend	56
Table 4.3	Summary of Hardware and Software used on this project	62
Table 5.1	Teaser Name and Team	79
Table 5.2	Software Requirement	79
Table 5.3	Hardware Requirement	80
Table 5.4	Testing Schedule using AnserverBot set A	80
Table 5.5	Testing Schedule using AnserverBot set B	81
Table 5.6	Testing Schedule using AnserverBot set C	81
Table 5.7	Testing Schedule using BeanBot set A	81
Table 5.8	Testing Schedule using BeanBot set B	82
Table 5.9	Testing Schedule using BeanBot set C	82
Table 5.10	Testing Schedule using GoldDream set A	82
Table 5.11	Testing Schedule using GoldDream set B	83
Table 5.12	Testing Schedule using GoldDream set C	83
Table 5.13	Testing schedule for network generalization for AnserverBot	83
Table 5.14	Data to be Tested	85
Table 5.15	Data to be Verified	86
Table 5.16	Test result for AnserverBot sets	87
Table 5.17	Test verification result for BeanBot sets	89
Table 5.18	Test verification result for GoldDream sets	89
Table 5.19	Parameter Summarization	89

LIST OF FIGURES

FIGURES	TITLE	PAGES
Figure 2.1	Framework of Literature Review	7
Figure 2.2	The original modules in normal apk files	8
Figure 2.3	The infected apps have additional modules in the apk files	9
Figure 2.4	The different original (right) and infected (left) assets/ directory	10
Figure 2.5	Original (left) and additional (right) permission in installed apps	10
Figure 2.6	Android Cupcake Key Features	11
Figure 2.7	Android Donut 1.6 Key Features	12
Figure 2.8	Android Éclair 2.0/2.1 Key Features	12
Figure 2.9	Android Froyo Key Features	13
Figure 2.10	Android Gingerbread Key Features	14
Figure 2.11	Android Honeycomb Key Features	14
Figure 2.12	Android Ice Cream Sandwich Key Features	15
Figure 2.13	Android Jellybean Key Features	16
Figure 2.14	Platform Version	17
Figure 2.15	Ice Cream Sandwich Adaptation	18
Figure 2.16	PC'S and Android Malware Evolution	19
Figure 2.17	Emergence of malware families for Android	23
Figure 2.18	Anserverbot Key Technique	29
Figure 2.19	How Anserverbot Malware Works	31
Figure 2.20	Classification of malware detection techniques	33
Figure 2.21	Static Analysis Technique for Android	34
Figure 2.22	Data Collection Technique	36
Figure 2.23	Malicious code is embedded in the package	39
Figure 2.24	Malicious activity "com.state.eguar.Dialogue"	40
Figure 2.25	The actual service of activity "com.allen.txthej"	40
Figure 3.1	Malware Analysis Administrator Methodology	43
Figure 3.2	Anserverbot Malware Analysis through Static Methodology	44
Figure 3.3	Summary of detail Research Methodology	45
Figure 3.4	Network Setup Environment Phase	46
Figure 3.5	Implementation Phase	47
Figure 3.6	Information Collection and Analysis Phase	48
Figure 3.7	Design Phase	48
Figure 3.8	Testing and Evaluation Phase	49
Figure 3.9	Documenting the Script Phase	50

Figure 4.1	Chapter 4 Framework	52
Figure 4.2	Logical Network Design	53
Figure 4.3	Basic Android Malware Attack Model	54
Figure 4.4	General Script Design	55
Figure 4.5	Process Flow of Scripting Design	55
Figure 4.6	Flow Chart of Scripting Design	57
Figure 4.7	Expected Output for Abnormal source code	58
Figure 4.8	Expected Output for Normal source code	58
Figure 4.9	VMware Workstation 8 Interface	59
Figure 4.10	Android Development Tools (ADT) Interface	59
Figure 4.11	Dalvik Debug Monitor Server Interface	60
Figure 4.12	Dex2jar Files	60
Figure 4.13	JD-GUI Interface	61
Figure 4.14	APK tool run on Command Prompt	61
Figure 4.15	Tcpdump running in Command Prompt	62
Figure 4.16	Source code fragment for read text file and get parameter from database	64
Figure 4.17	Snapshot of the VMware Installation	65
Figure 4.18	Snapshot of basic Guest OS Installation	66
Figure 4.19	Snapshot of installing Windows 7	66
Figure 4.20	Snapshot creating Android device	67
Figure 4.21	Virtual Android Device is launched	68
Figure 4.22	Functional step of custom build Android package analyzer	68
Figure 4.23	Change to .zip file	69
Figure 4.24	Extracting the Dex2jar folder	69
Figure 4.25	Creating classes_dex2jar files	70
Figure 4.26	JD-GUI application	70
Figure 4.27	The classes_dex2jar.srcis created	71
Figure 4.28	Content of folder	71
Figure 4.29	Install framework_res.apk	71
Figure 4.30	Apktool converting the .apk to .java files	72
Figure 4.31	The source folder is created	72
Figure 4.32	Import the files into Eclipse	72
Figure 4.33	Installing the .apk file	73
Figure 4.34	Application installed	73
Figure 4.35	Package Summary	74
Figure 4.36	Application request permission	74
Figure 4.37	Capture Option	75
Figure 4.38	Running capturing network traffic	76
Figure 4.39	Save Option	76
Figure 5.1	Chapter 5 Framework	78
Figure 5.2	Classes of Test	84
Figure 5.3	Process of the Script	86
Figure 5.4	AnserverBot set Onion_A.txt result	87

Figure 5.5	AnserverBot set Onion_B.txt result	88
Figure 5.6	AnserverBot set Onion_C.txt result	88
Figure 5.7	Additional response from public blog as C&C server	90
Figure 5.8	Original response from public blog	90
Figure 6.1	Chapter 6 Framework	92
Figure 6.2	Relation between botmaster and servant bot	94

LIST OF ABBREVIATIONS

ADB	Android Debug Bridge
ADT	Android Developer Tools
API	Application Programming Interface
AV	Anti-Virus
C&C	Command & Control
CFG	Control Flow Graph
DDMS	Dalvik Debug Monitor Server
GNU	Genuinely Not Unix
HD	High Definition
I/O	Input / Output
ICS	Ice Cream Sandwich
IDS	Intrusion Detection System
NAT	Network Address Translation
NFC	Near-Field Communication
OS	Operating System
PC	Personal Computer
PiOS	Pontin's International Open Series
SDK	Software Development Kit
SMS	Short Message Service
TCP/IP	Transmission Control Protocol / Internet Protocol
USB	Universal Serial Bus
VM	Virtual Machine

CHAPTER 1

INTRODUCTION

1.1 Project Background

Malicious software that is commonly referred as ‘malware’ mainly targets desktop PCs. But, in this cyber generation, cybercriminals are increasingly sets their sight on the smartphone and mobile devices such as tablets or PDAs. Vanja Svajcer (Svajcer, 2012) imagine that malware coming from a groups of hackers walled up in non-descript offices and spending their days by pinging websites to search the vulnerability for doing exploitation. Vadodil Joel Varghes (Varghese, 2011) mention that malware capability is included of penetrating into the systems, networks without user intervention and can disrupt services by compromising the confidentiality, integrity and availability of the applications, systems and operating systems.

There a two main technique used for detecting malware that can be categorized by anomaly-based detection and signature-based detection (Aditya P. Mathur, 2007). In this project, we are using static analysis in anomaly-based detection that used file structure of the program that under the inspection to detect the malicious code.

These projects are conducted with certain objectives which include:

1. To identify the behaviour of Android malware.
2. To generate the attack pattern of Android malware.
3. To formulate the procedure of extracting the attack pattern (script).

The goal of this project is to understand the behaviour of an AnserverBot Android malware by static analysis and providing a script.

1.2 Research Problem

Table 1.1: Research Problem

RP	Research Problem
RP_1	Difficulties on identifying malware behavior in Android environment.

1.3 Research Question

Table below shows the question need to be answered during this research.

Table 1.2: Research Question

RP	RQ	Research Question
RP_1	RQ_1	How the Android malware works?
	RQ_2	How to know what is malware attack?
	RQ_3	How to extracting attack pattern?

1.4 Project Objectives

The objective of this reaserch are stated as table above.

Table 1.3: Research Objectives

RP	RQ	RO	Research Objectives
RP_1	RQ_1	RO_1	To identify the behaviour of Android malware.
	RQ_2	RO_2	To generate the attack pattern of Android malware.
	RQ_3	RO_3	To formulate the procedure of extracting the attack pattern (script).

1.5 Project Scopes

The scopes of this project are including:

1. Implemented only on specific type of malware attack which is AnserverBot malware.
2. Using a VMware Workstation 8 as platform
3. Using Windows 7 Ultimate as a based operating system.
4. Use a static analysis technique to get a malware attack pattern.

1.6 Research Contribution

Table above shows the contribution of this research.

Table 1.4: Research Contribution

RP	RQ	RO	RC	Research Contribution
RP_1	RQ_1	RO_1	RC_1	Classification of Android malware behavioral.
	RQ_2	RO_2	RC_2	Proposed the general Android malware's attack pattern.
	RQ_3	RO_3	RC_3	Proposed script/technique to extracting the malware attack

1.7 Expected Output

The goal of this project is to identify the normal or abnormal behaviour of the source code by producing a script. In this project analyst going to take you through the various phases so as to understand how and what are these malwares exactly made up of and how these malware's behave in an isolated environment.

1.8 Report Organization

This report will have 6 chapters that consist of:

Chapter 1: Introduction

- Chapter 1 tells a basic about this project. Describe the background, problem statement, research question, project objective, project scope, and the expected output of this project.

Chapter 2: Literature Review

- Chapter 2 will describe specifically what malware is, what is static analysis and environment use related to this project.

Chapter 3: Methodology

- Chapter 3 is an explanation of how to do the experimental using tools to analyse malware in operating system, network traffic, and so on.

Chapter 4: Design and Implementations

- Chapter 4 will show the design of the hardware and software requirement. Implementation will describe details on how the analysis or experimental works with sample output or result.

Chapter 5: Testing and Analysis

- Chapter 5 describe the method or step how to test and analyse in system development and comparative analysis of the result.

Chapter 6: Conclusion

- Chapter 6 is a conclusion of the overall project. Describe the limitation that faced in this project, contribution and future works.

1.9 Conclusion

This chapter manually describe about the objective doing this project is to overcome the problem faced. Scope and expected output has been stated as a basic guideline to successfully finish the analysis of Android malware.

After knowing the objective, the basic is to know what need to analyse, and using what tools or technique. What actually Android is. What is malware and others component or parameter that will use during this project analysis.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explained the detail about definition and what analyst should do for this research. This is an introduction between analyst and the project. Analyst need to know background of what to analyse. This chapter will specifically tell the detail about android, malware, and technique used to this project. Figure 2.1 shows the framework of literature review that will be guideline what analyst wants to explain in this chapter.

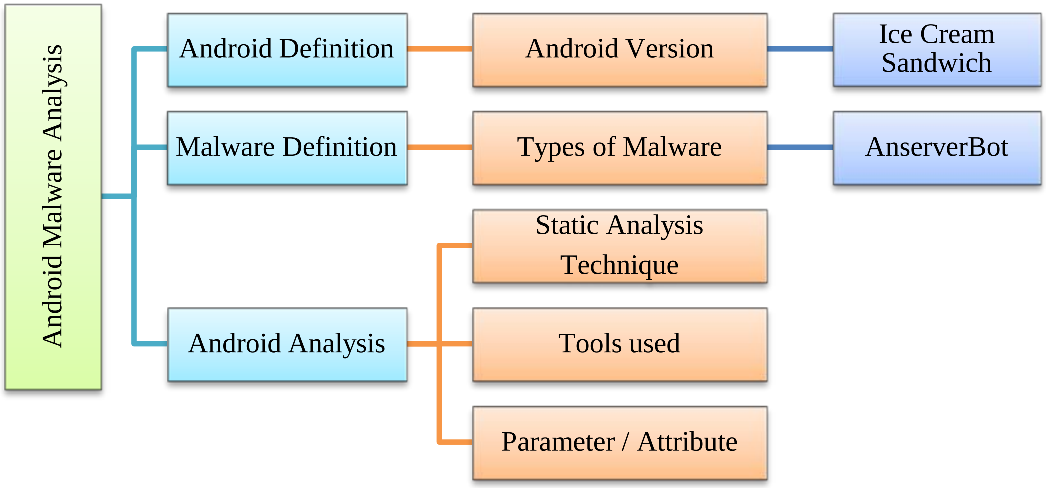

Figure 2.1: Framework of Literature Review

2.2 What is Android?

Android is an operating system that is designed for mobile device that are usually touch screen such as smartphones and tablets. It has been developed by Android Inc. in 2007 and the first android-powered phone was sold to Google in October 2008. Android is an open source and the code release is under the Apache Licence. This open and permissive licensing allows the software to be free-modified and distributed by device manufacture, wireless carriers and enthusiast developers.

Android also become a target of massive malware attack and these attack have been increasing wisely. It usually involves multi-functional types of malware that can steal contact information and personal information stored in user's smartphones. Otherwise, Android malware also can download packet of data from malicious servers that can cause user mobile a more threats in the future.

2.2.1 Android Package and Code files

The *.apk file* is an Android application package file. Each application is compiled and packaged in a single file that includes all of the application's code (*.dex* files), resources, assets, and manifest file. The application package file can have any name but must use the *.apk* extension. For example: *myExampleAppname.apk*. For convenience, an application package file is often referred to as an ".apk".

The *.dex* file is a compiled of an Android application code file. Android programs are compiled into *.dex* (Dalvik Executable) files, which are in turn zipped into a single *.apk* file on the device. *.dex* files can be created by automatically translating compiled applications written in the Java programming language.

2.2.2 Android Application

An Android Application (apps) is a software application that running on the Android platform (Michael, 2012). Usually, Android apps are written in the Java programming language and use Java core libraries. It first compiled to Dalvik executable to run on the Dalvik virtual machine, which is a virtual machine specially designed for mobile devices.

Usually, original apps which are not infected with Android malware are known as normal apps. This normal app only have application module, a few requested permission, request extra charges or don't have database name **anservera.db** and **anserverb.db**. But, the infected apps with Android malware adding two more modules plus it requesting many permissions. Appendix A will show the different between the Original Manifest and the Infected Manifest (by AnserverBot malware) requested permission. Figure 2.2 shows the original module name **com.apkbook.jinpmei** and **com.apkbook.meibai** in the apk files.

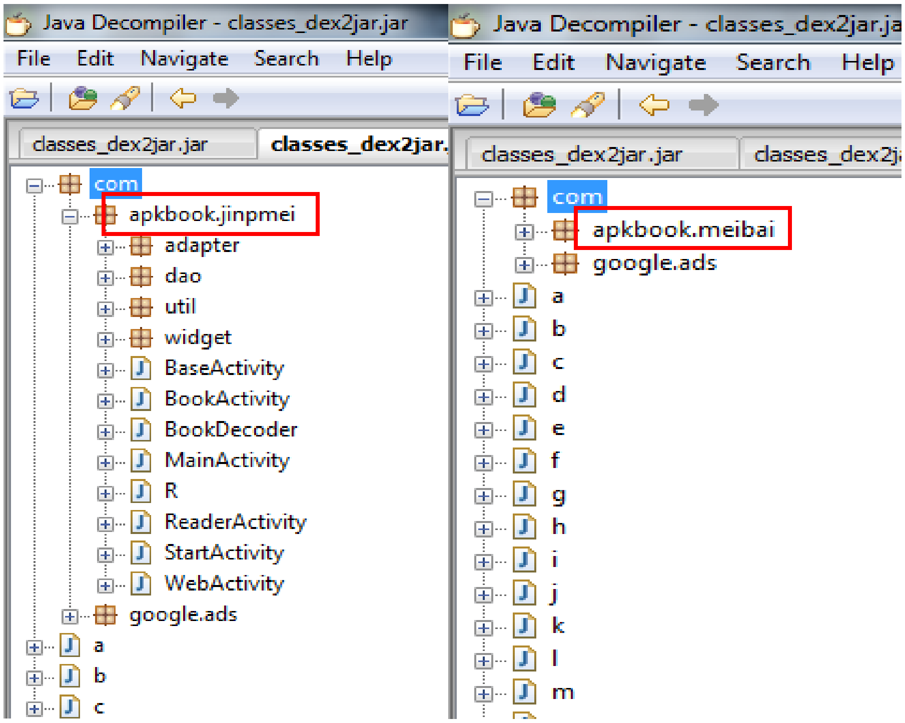

Figure 2.2: The original modules in normal apk files.

While in Figure 2.3 shows the two modules is added in the .apk file named

com.android.view.custom and **com.sec.android.providers.drm**.

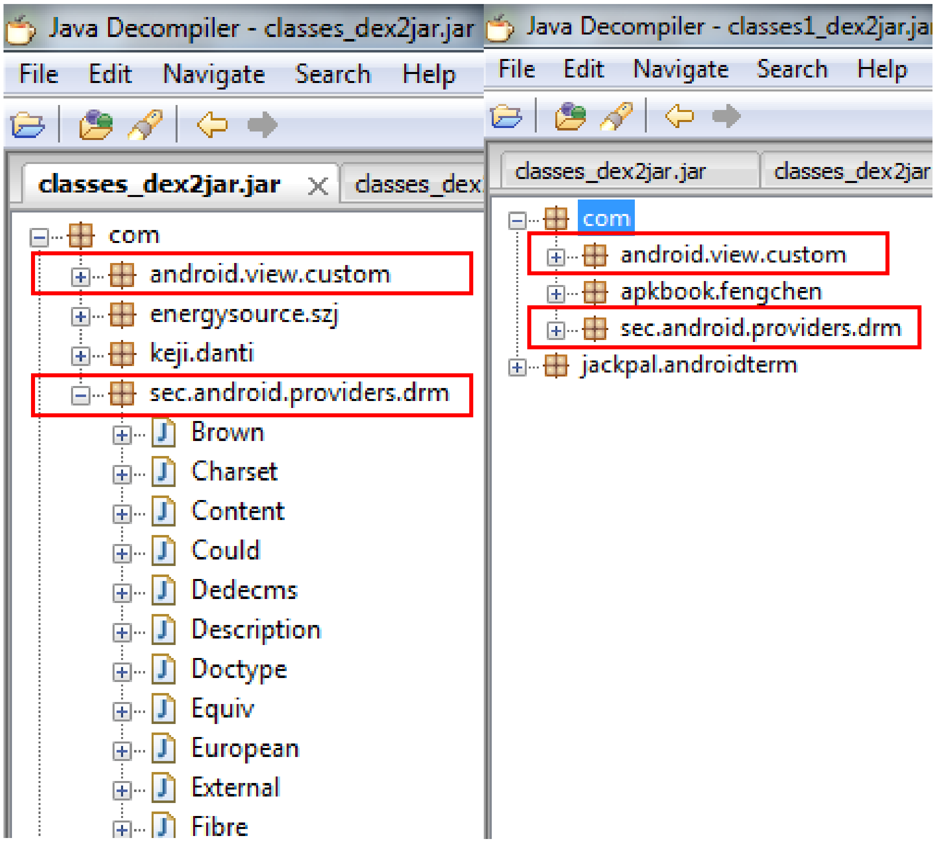

Figure 2.3: The infected apps have additional modules in the apk files.

These two additional modules have different function. The **com.android.view.custom** contains the bridge routines to access payload B and the module **com.sec.android.providers.drm** is essentially a bot client that connects to remote Command and Control (C&C) servers to download and upgrade payload B.

Another different between original apps and infected is an additional database under **assets/** directory. Figure 2.4 shows the apps with and without the malware. The **anservera.db** will be installed once the app runs and **anserverb.db** will be dynamically loaded to run without installation.