ANDROID APP TO MONITOR AND PROTECT FROM REMOTE ACCESS TROJAN

WRITTEN BY:
MUHAMMAD ALWI SHIHAB
201481096

INFORMATION TECHNOLOGY STUDY PROGRAM
FACULTY OF COMPUTER SCIENCE
ESA UNGGUL UNIVERSITY
JAKARTA
2018

ADVISOR : **LI QING (李青)**
MALABAY, S.Kom, M.Kom

ANDROID APP TO MONITOR AND PROTECT FROM REMOTE ACCESS TROJAN

To fulfil the requirement for attaining Bachelor Degree in Computing

WRITTEN BY:
MUHAMMAD ALWI SHIHAB
201481096

INFORMATION TECHNOLOGY STUDY PROGRAM
FACULTY OF COMPUTER SCIENCE
ESA UNGGUL UNIVERSITY
JAKARTA
2018

ADVISOR : **LI QING (李青)**
MALABAY, S.Kom, M.Kom

DECLARATION OF ORIGINALITY

I, the undersigned below:

Name	: Muhammad Alwi Shihab
Student ID No	: L14300102 / 201481096
Study Program	: Information Technology
Faculty	: Computer Science
Degree	: Bachelor Degree

Hereby declared that the thesis I wrote with the title:

ANDROID APP TO MONITOR AND PROTECT FROM REMOTE ACCESS TROJAN

1. Is truly a research written purely by myself, not copying from other published researches, and also not a result of plagiarism.
2. I will allow Nanjing Xiaozhuang University and Esa Unggul University to manage and keep the copy of this thesis, to be used as they deem necessary.

I made this statement of declaration with fully responsibility, and I'm willing to accept any consequences according to the rules and regulation should the statement above proved to be wrong in any way.

Nanjing, June 12, 2018

Muhammad Alwi Shihab

ACKNOWLEDGEMENTS

The first thanks to Allah SWT for all grace and guidance so I can do and finish this thesis with good spirit. I would like to acknowledge the assistance of the many people and source who generously help to make this project a reality. The title of this thesis is "Android App to Monitor and Project from Remote Access Trojan".

This thesis purpose is to complement, carry out academic curriculum, to make a reality of project the author wanted, to complete one of academic requirements for student at the Faculty of Computer Science both in Esa Unggul University Jakarta and Nanjing Xiaozhuang University to complete the Bachelor Degree program.

With all limitations, the author realizes also that this Thesis project will not be realized without the help, guidance, and encouragement of various parties. For that the author expressed his gratitude to:

1. My parents who always pray, give support, encouragement, love and believe to me to do all of this things.
2. My brother and sister who have been so great giving me support.
3. Rachma Giri Alifia, who gives me strength, believe me, and realizes that life is nice, beautiful.
4. Mrs. Sophie Mou as my mother in China, which always there to help, listen, support, give me spirit take care of me and advice until the end of my study. I'm very greatful and thank you for help me so I can continue my study to higher level.

5. Mr. Li Qing Laoshi and Malabay S.Kom, M.Kom. As my advisor. I am fortune to have two advisors in this thesis who always provide me guidance, support and patience for finishing this thesis.
6. To Esa Unggul University and Nanjing Xiaozhuang University who has given me the opportunity to study.
7. All classmate in Jakarta and Nanjing, and who help me to continue study in Nanjing and for supporting and provide advice to me.
8. Thank you to HIMASTIKA, FANGSHANTEAM and who cannot be mentioned one by one, who have helped and support me to finish this thesis.

Hopefully God always blessed those mentioned above and all their sacrifice becomes their merciful deeds to help their success future life. The author realizes that thesis is far from being perfect and there are shortcomings in the writing of this thesis due to time constraints. Nevertheless, the authors have attempted to create a final project that can be useful for writers and readers.

Nanjing May 16, 2018

APPROVAL PAGE

Student ID	: L14300102 / 201481096
Name	: Muhammad Alwi Shihab
Study Program	: Information Technology
Faculty	: Computer Science
Degree	: Bachelor Degree
Thesis title :	Android App to Monitor and Protect from Remote Access Trojan

This thesis has been examined by the panel of Faculty of Computer Science

Thesis examiners, Information Technology Study Program, and hereby declared as

PASSED.

Jakarta, August 14, 2018

Chief Examiner : **Malabay S.Kom, M.Kom** ________________

Examiner I : ________________

Examiner II : ________________

APPROVAL PAGE

Name	: Muhammad Alwi Shihab
Student ID	: L14300102 / 201481096
Study Program	: Software Engineering
Faculty	: School of Information Engineering
Thesis title	: Android App to Monitor and Protect from Remote Access Trojan

The Final Thesis above has been approved and accepted as one of the requirements to obtain a Bachelor of Engineering and Sarjana Komputer in the Software Engineering Study program, School of Information Engineering, Nanjing Xiaozhuang University on June 12th, 2018 and Informatics Engineering Study Program, Faculty of Computer Science, Esa Unggul University on August 2018.

Li Qing
Advisor I (China)

Malabay S.Kom, M.Kom
Advisor II (Indonesia)

Acknowledged,

Malabay, S.Kom, M.Kom
Head of IT Esa Unggul University

Professor Chen Wei Wei
Dean of School of Information Engineering NJXZU

Dr. Ir. Husni S. Sastramihardja, MT
Dean of Computer Science Faculty Esa Unggul University

ABSTRACT

The mobile phone, where people always use on daily life and total of phones is bigger than the people population itself. Internet has become an essential part of mobile phone for half of people population. The mobile and internet are together has many services and growing every second then more people are using these services. Internet banking and Social media are the examples of internet services using mobile phone. These services make people try to take advantage of legitimate users whenever money is involved. Malware is one of cybercrime software attacked on mobile phone, Remote Access Trojan included in variant of malware. Android operating system has the highest market share in the world, this fact makes android users target group criminals. Android's security mechanism is based on instrument that informs users about which permission the application used, this permission system may help gain awareness about the risks and which application ability in the system. Trojan is one of the most malware founded in android application, this Trojan allows attacker to go through the device system without notice to the owners. The user awareness is one of point that vulnerability, doesn't realize if users installed some malware and it steal anything. This paper presents a network activity, permission usage and encryption-based to detect and protect Android system. This work presents an innovative methodology that helps in the process of Remote Access Trojan detection for Android Operating system. Based on Network Activity, Permission usage and also encryption to help the Android user protect their device. This application use static analysis to characterize and classify Android applications as malicious.

Keywords: Android, Trojan Malware, Remote Access Trojan, User Awareness.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of the Study

These days a modern era, technology are growing so fast on every aspect, mobile phone is one of technology. Current mobile devices offer a large amount of service and applications than those offered by personal computers. At the same time, security threats increase and mobile devices became a target. The malicious users and hackers are taking advantage of both the limited capabilities of mobile devices and the lack of standard security mechanisms to design malicious application that can access sensitive data, stealing phone credit, spying users and crashing the device. Everyday malware attack increased in specially on Android platform.

Android's security model significantly relies on permission-based mechanisms. Permission are requested by applications and they are required to access application's interface/data that are defined in its manifest file. They are used to inform users about application's capabilities and the user does not realize/aware that Android security frame work does review applications permissions, but Android security framework does not review system though their network activity which to inform users which connection has connected to the device.

Kabakus et al. (2015) present a tool named *APK Auditor* that offers static analyses with permission-based Android malware assessment system.

Burguera et al. (2011) present *Crowdroid*, a framework to detect anomalous behavior of Android application. In order to categorized Android application as malicious, Crowdroid that collects Linux Karnel system calls containing device information.

Acosta-Guzman et al. (2015) present a Network Activity Monitoring Against Malware in Android Operating System as methodology to malware detection.

This application present a tool to against Malware as specific Malware is Remote Access Trojan based on Network Activity, Permission Usage and Encryption. There are variety of Trojan which has been created in this world for example Backdoor, Exploit, Ransom, Spy, etc. Remote Access Tool is one of the technology that is used to control operating system on devices and in this case Android system. However, there are a lot of illegal remote access tool one of which is known as Trojan horse malware on the internet to attack device and gain full control without any permission and user cognition. Malicious software or malware is used by Cybercriminals, Hacktivists, and Government to harm devices, steal personal information, spying users, etc. Malware contains an executable script or code used to bypass access control and disguise itself as a common software to deceive system and also antivirus.

1.2 Problem Identification

Based on the background, the author can identify the problems for the development of this application, are as follows:

1. How to identify Remote Access Tool Trojan via Network-Based?
2. How to identify Remote Access Tool Trojan via Permission-Based?
3. How to identify Remote Access Tool Trojan via Third-Party App Installed?
4. How to identify Remote Access Tool Trojan via Permission Usage?

1.3 Purpose of the study

The purpose of this system development are as follows:

1. This application is built to help, facilitate and increase user awareness from remote access Trojan.
2. This application is designed to monitoring and protect mobile phone from remote access Trojan.

1.4 Limitation of the study

Some limitations in this study are as follows:

1. This application is applied to the Android operating system version 5.0 (Lollipop) to 8.0 (Oreo).
2. This application cannot identify network status with application name.
3. This application has no uninstall feature.

1.5 Benefits of the study

The benefits of this research are as follows;

1. This application can help users to monitoring background connection between the device and other network.
2. This application can help users to finding a third-party application installed in device.
3. This application can help users to find which third-party application use specific permission.
4. This application can help users to finding location of network address which specify to Trojan server.
5. This application can help users to protect an important file.

1.6 Research Methodology

In this thesis, the methodology to be used are as follows:

1. Data Collection Method

   In this method, the authors utilize the resources of the library references such as journals, articles and the internet to get the supporting theories and references for the development of application.
2. Software Design Method

   The method used in the design of software is using extreme programming (Pressman, 2012). The process used in as follows:
   a. Analysis of needs

      Analysis of needs applied by identifying that needs to be obtained. The analytical method used by using the modelling language UML (Unified Modelling Language). Analysis of needs is done by identifying all of the software requirements based on the data obtained from the study of literature and then modelled into the use case diagram.

   b. Coding

      Coding is the stage for designing the application using Android Studio, and implement the UML model also Java Android.

   c. Design

      This stage emphasizes the design of simple applications. Each requirement that have been poured into the UML model is determined in the construction period.

   d. Testing

      This stage focuses on testing the feature and functionality of the application, whether it meets the requirements and solve the problem or not.
3. Documentation

   This stage is the preparation for the application documentation.

1.7 Writing Structure

The systematics of writing this thesis is divided into five chapters, as follows:

CHAPTER I INTRODUCTION

This chapter discuss the background of the study, problem identification, limitations of the study, benefits of the study and research methodology.

CHAPTER II THEORETICAL BASIS

This chapter discuss theoretical basis and literature review that related to this research.

CHAPTER III ANALYSIS AND DESIGN

This chapter discuss the analysis in building application as well as the design of the application.

CHAPTER IV IMPLEMENTATION AND TESTING

This chapter discuss the result of the analysis and design into code to build a working application, and testing to find out whether the application is working.

CHAPTER V CLOSING

This chapter discuss the conclusion of this thesis and suggestions for further research.

1.8 Schedule

Scheduling will be implemented in this thesis as follows:

No	Activity	Weeks								
		1	2	3	4	5	6	7	8	9
1	Literature review	■	■							
2	Analysis of needs/requirements			■	■					
3	Design				■					
4	Coding					■	■	■	■	
5	Testing					■	■	■	■	
6	Documentation									■

The process of arrange this thesis takes approximately 9 weeks, with the first week starting on March 29th, 2017 until May 31st, 2018

CHAPTER 2
THEORETICAL BASIS

2.1 Remote Access Trojan

A remote access Trojan (RAT) is a malware program that used for malicious purposes, this Trojan inspired by remote access tools. Remote access tool is a software used to remotely or control a computer with legitimately used by system administrator. RAT also known as a backdoor for administrative control over the target device. These day, a million of remote access Trojan has been created which they free or paid with multiple operating system as target. RATs are usually downloaded invisibly with a user-requested program such as common software, game or sent as an email attachment. Once the target system is infected, it will perform unauthorized operations and hide in the system. The attacker can remotely control the system to gaining the camera, microphone, key logs, screen capture, harm the device, etc.

Figure 2.1 Trojan Horse Illustration
(Source: gbhackers.com)

The most popular RATs, such as Back Orifice, SubSeven, DroidJack, SpyNote, AhMyth and TheFatRAT, are all-in-one intruder tools that do everything capture screen, sound, video content, key loggers, remote controllers, FTP servers, HTTP servers, Telnet servers, and password finders. Intruders can configure the IP port the RATs listen on, how the RATs execute. The more malicious RATs contain encrypted communications, and contain professional looking APIs so that other intruder developers can insert additional functionality. These RATs aggressive with large functionality has often size 200KB to 700K [27].

2.2 Android

Android is a operating system and software for mobile devices, based on the Linux kernel, and developed by Google. This operating system is open-source that mean allows developers to write managed code in the Java language, controlling the device via Google developed Java libraries or Android Studio. Android is a freely downloadable software and operating system stack for mobile, middleware and language applications based on Linux and Java. Google purchased the developer of Android in 2005, and Android was unveiled in 2007. Google released the Android code as open-source under the Apache License [24].

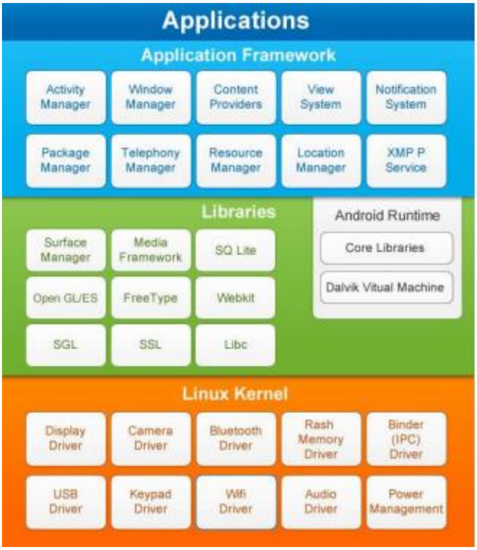


Figure 2.2 Android Architecture
(Source: Narmatha et al., 2016)

Android open-source project, this Android was founded on October 2003 described the Android project as “tremendous potential in developing smarter mobile devices that are more aware of its owner’s location and preferences.” Rubin said, as one of build Android Inc. Android’s kernel is based on the Linux kernel’s long-term support (LTS) branches.

2.2.1 Linux Kernel

The Lower layer of android operating system is Linux kernel. The Android does not really interact with the users and developers but with the Linux kernel system. The Linux Kernel provides a level of abstraction between the device hardware and the upper layers of the Android software stack. On 2017, Android device mainly based on Linux version 3.18 or 4.4 of the Linux kernel, the kernel provides preemptive multitasking, low level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers.

2.2.2 Libraries

On top of Linux kernel there is a set of c/c++ libraries used by the various components of the android system. Some of the core libraries are listed below:

a. System c library

   A BSD derived implementation of the standard and system library (libc) turned for embedded Linux-based devices.

b. SQLite

   It is used to access data published by content providers and includes SQLite database management classes.

c. SSL

   It is used to provide internet security.

d. SGL

   The underlying 2D graphics engine.

e. Libwebcore

A modern web browser engine which powers both the android browser and an embedded web view. OpenGL – It is used to provide Java interface to the OpenGL/ES 3D graphics rendering API.

f. Media Framework

It is used to provide different media code which allow the recording and playback of different media formats.

g. Web Kit

It is the browser engine used to display internet content or HTML content.

2.2.3 Android Runtime

Android Runtime (ART) is an application runtime environment used in Android operating system and its predecessor Dalvik Virtual machine and Core Java libraries that used by applications and some system services on Android. It is located on the same level as the library 20 layer. The Dalvik Virtual Machine enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Dalvik Virtual Machine allows multiple instance of Virtual machine to be created simultaneously providing security, isolation, memory management and threading support. Unlike Java Virtual Machine which is process-based, Dalvik Virtual Machine is register base. Dalvik Virtual Machine run .dex bytecode specification files which are created from .class file by dx tool. dx tool is included in Android Software Development Kit (SDK). Dalvik Virtual Machine is optimized for low processing power and low memory environments. DVM is developed by Dan Bornstein from Google. Android Runtime (ART) and Dalvik are compatible work when running together but some techniques that work on Dalvik do not work on ART.

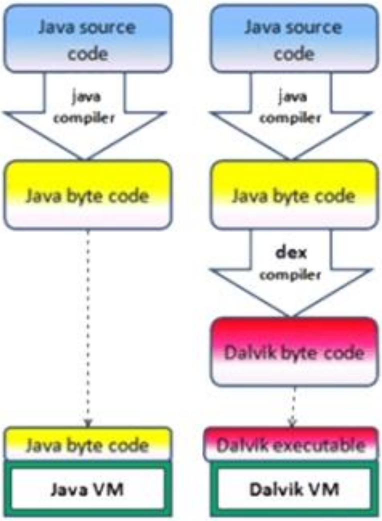


Figure 2.3 Dalvik Virtual Machine
(Source: Narmatha et al., 2016)

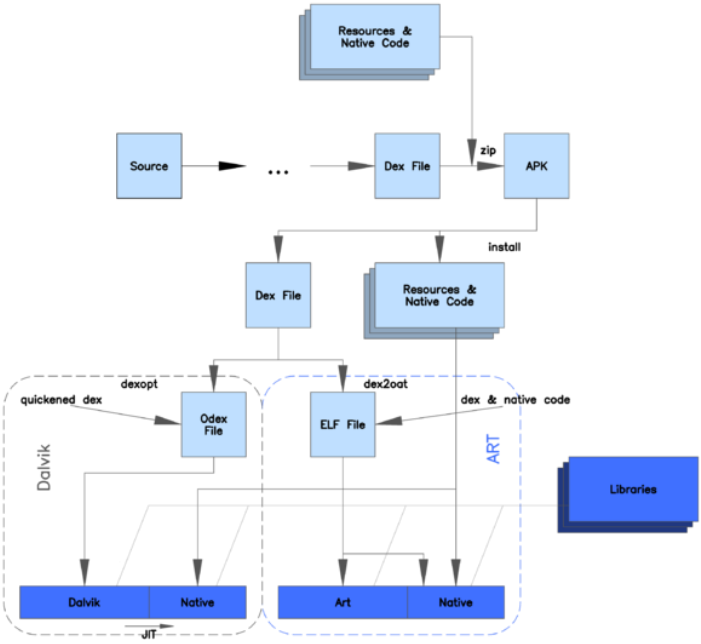


Figure 2.4 A comparison of Dalvik and ART architectures

2.2.4 Application Framework

The Application Framework layer provides many higher-level services to applications in the form of Java classes. The developers are allowed to make use of these services in application. This framework describes the concept that Android applications are constructed from reusable, interchangeable and replaceable components. The Android framework includes the following key services:

a. Activity Manager: Controls all aspects of the application lifecycle and activity.

b. Content Providers: Allows applications framework to publish and share data with other applications.

c. Resource Manager: Provides access to non-code embedded resources such as strings, colour settings and user interface layouts.

d. Notifications Manager: Provides and allows applications to display alerts and notifications to the user.

e. View System: An extensible set of views used to create application user interfaces.

f. Package Manager: The system by which applications are able to find out information about other applications currently installed on the device.

g. Telephony Manager: Provides the telephony services available on the device such as status and subscriber information.

h. Location Manager: The system provides access to the location services allowing an application to receive updates about location changes.

2.2.5 Application

The applications layer is the top layer of the Android stack. The user of the Android device would mostly interact with this layer. Android will ship with a set of applications including an email client, calendar, browser, maps, contacts, others. All applications are written using java programming language.

2.2.6 Android Version

Android is updating day by day since its release. These updates to the base operating system mainly focusing on fixing bugs as well as adding new features to provide more comfortable environment and fix security issues. The most recent released versions of Android are:

Table 2.1 Android Version

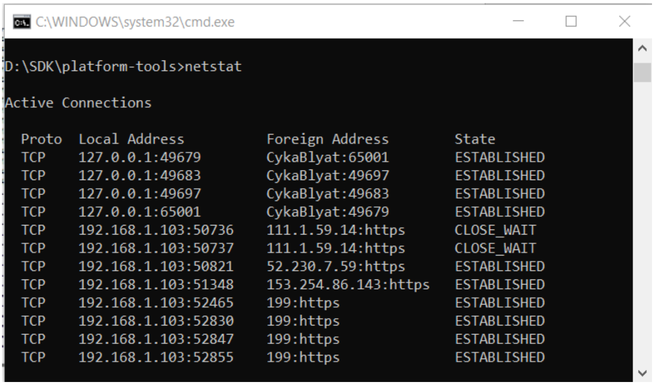
Code Name	Version Number	Release Date	API Level
No Name	1.0	September 23, 2008	1
Petit Four	1.1	February 9, 2009	2
Cupcake	1.5	April 27, 2009	3
Donut	1.6	September 15, 2009	4
Éclair	2.0 – 2.1	October 26, 2009	5 – 7
Froyo	2.2 – 2.2.3	May 20, 2010	8
Gingerbread	2.3 – 2.3.7	December 6, 2010	9 – 10
Honeycomb	3.0 – 3.2.6	February 22, 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	July 9, 2012	16 – 18

KitKat	4.4 – 4.4.4	October 31, 2013	19 – 20
Lollipop	5.0 – 5.1.1	November 12, 2014	21 – 22
Marshmallow	6.0 – 6.0.1	October 5, 2015	23
Nougat	7.0 – 7.1.2	August 22, 2016	24 – 25
Oreo	8.0 – 8.1	August 21, 2017	26-27
Android P	9	May 8, 2018	28

2.3 Network Activity Based

Network Activity Monitoring Based, according to Acosta-Guzman [11]. It is important to emphasize that the approach of this methodology consists in the development of the mechanism to keep track of the network activity not the mechanisms to evaluate and detect any suspicious behavior and nor to identify malware. However, methodology shows the implementation of a Network Activity Based for the application named Network Status related on Acosta-Guzman project, this Network Status is capable to running and update every user reopen this feature in order to identify new established connections and attempts of connection. This Network Status capable show internet protocol and listening open ports on it, lead the user to identifying network on device.

Considering the aforementioned Acosta-Guzman methodology, Network Activity Monitoring App works without the need of rooted device. The network activity itself cannot be used to identify malware but it could be used to identify all connection running on device which a Remote Access Trojan has server address used on target client. As a consequence, the user running this network status on its device may have an opportunity to kill the process and uninstall App that is generating new network activity. Alerting the user is the only reactive measure the network status can take without requiring root permissions.

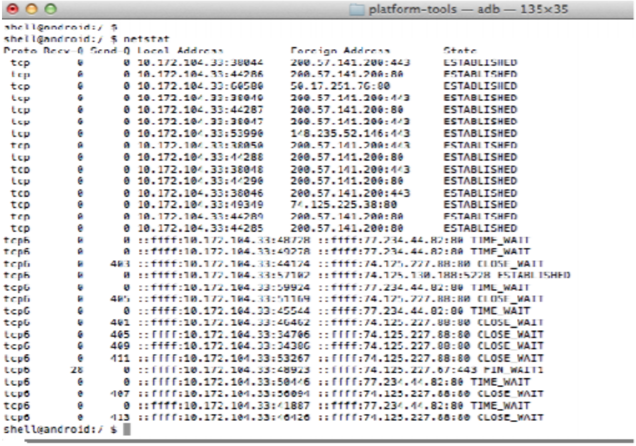


```
D:\SDK\platform-tools>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49679        CykaBlyat:65001        ESTABLISHED
  TCP    127.0.0.1:49683        CykaBlyat:49697        ESTABLISHED
  TCP    127.0.0.1:49697        CykaBlyat:49683        ESTABLISHED
  TCP    127.0.0.1:65001        CykaBlyat:49679        ESTABLISHED
  TCP    192.168.1.103:50736    111.1.59.14:https      CLOSE_WAIT
  TCP    192.168.1.103:50737    111.1.59.14:https      CLOSE_WAIT
  TCP    192.168.1.103:50821    52.230.7.59:https      ESTABLISHED
  TCP    192.168.1.103:51348    153.254.86.143:https   ESTABLISHED
  TCP    192.168.1.103:52465    199:https              ESTABLISHED
  TCP    192.168.1.103:52830    199:https              ESTABLISHED
  TCP    192.168.1.103:52847    199:https              ESTABLISHED
  TCP    192.168.1.103:52855    199:https              ESTABLISHED
```

Figure 2.5 Network Status result on terminal windows



```
shell@android:/ $
shell@android:/ $ netstat
Proto Recv-Q Send-Q Local Address          Foreign Address        State
 tcp       0      0 10.172.104.33:38044    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:44286    200.57.141.200:80      ESTABLISHED
 tcp       0      0 10.172.104.33:60580    50.17.251.76:80        ESTABLISHED
 tcp       0      0 10.172.104.33:38049    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:44287    200.57.141.200:80      ESTABLISHED
 tcp       0      0 10.172.104.33:38047    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:53990    148.235.52.146:443     ESTABLISHED
 tcp       0      0 10.172.104.33:38050    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:44288    200.57.141.200:80      ESTABLISHED
 tcp       0      0 10.172.104.33:38048    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:44290    200.57.141.200:80      ESTABLISHED
 tcp       0      0 10.172.104.33:38046    200.57.141.200:443     ESTABLISHED
 tcp       0      0 10.172.104.33:49349    74.125.225.38:80       ESTABLISHED
 tcp       0      0 10.172.104.33:44289    200.57.141.200:80      ESTABLISHED
 tcp       0      0 10.172.104.33:44285    200.57.141.200:80      ESTABLISHED
tcp6       0      0 ::ffff:10.172.104.33:48728 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0      0 ::ffff:10.172.104.33:49278 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0    403 ::ffff:10.172.104.33:44174 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0      0 ::ffff:10.172.104.33:57102 ::ffff:74.125.130.188:5228 ESTABLISHED
tcp6       0      0 ::ffff:10.172.104.33:59924 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0    405 ::ffff:10.172.104.33:51169 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0      0 ::ffff:10.172.104.33:45544 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0    401 ::ffff:10.172.104.33:46462 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0    405 ::ffff:10.172.104.33:34706 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0    409 ::ffff:10.172.104.33:34386 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0    411 ::ffff:10.172.104.33:53267 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6      28      0 ::ffff:10.172.104.33:48923 ::ffff:74.125.227.67:443 FIN_WAIT1
tcp6       0      0 ::ffff:10.172.104.33:50446 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0    407 ::ffff:10.172.104.33:56094 ::ffff:74.125.227.88:80 CLOSE_WAIT
tcp6       0      0 ::ffff:10.172.104.33:41887 ::ffff:77.234.44.82:80 TIME_WAIT
tcp6       0    413 ::ffff:10.172.104.33:46426 ::ffff:74.125.227.88:80 CLOSE_WAIT
shell@android:/ $
```

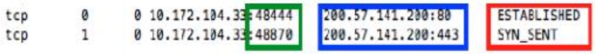
Figure 2.6 Netstat system on Android adb (Acosta-Guzman *et at.* 2015)

The Android operating system has terminal a call to the system invoking the command netstat directly from the kernel through the ”Process” and “Runtime” Java Classes and the method “getRuntime().exec()”, this netstat command provides an output showing all active sockets in a table containing important information such as protocol the local Internet protocol address, foreign address as destination IP address, (TCP, UDP, TCP6 or UDP6), and also the state of the socket port used on it [8].

The state result provides a way to differentiate connections that are already established (“ESTABLISHED”) from connections that are initiating (“SYN_SENT”), connections ending (“CLOSE_WAIT”, “TIME_WAIT”, “FIN_WAIT” and many more) and listening ports that are waiting for a connection from a different host in the network (“LISTEN”).

Once the active sockets are obtained it is necessary to determine the state for each element of the output just between three types: active connections, attempts of connections, and listening ports.

Besides the “State” it is essential to identify the value on the parameter “Proto” of the output for each element. This parameter can only take one of the following values: “TCP”, “TCP6”, “UDP” or “UDP6”. The following is to identify the destination IP address as well as the source and destination ports for each element of the netstat output with a state “ESTABLISHED” of “SYN_SENT; in other words, for active connections and connection attempts. From the elements with a “LISTEN” state it is only possible to obtain the number of the listening port.



```
tcp        0      0 10.172.104.33:48444   200.57.141.200:80    ESTABLISHED
tcp        1      0 10.172.104.33:48870   200.57.141.200:443   SYN_SENT
```

Figure 2.7 Source port, destination IP and state of the connection (Acosta-Guzman *et at.* 2015)

Identifying address and ports is fundamental not only because it is the connection’s detailed and important information by itself but also because it will lead to the identification of the Android and the App responsible for each socket. In a regular Linux Kernel or a Windows OS where the netstat command can also be found and figure on 2.5.

2.4 Permission Usage Based

Permission Usage Based, according to Abdullah Talha on *APK Auditor*(2014). Android's security mechanism is based on an instrument that informs users about which permissions the application needs to be granted before installing them. This permission system provides an overview of the application and may help gain awareness about the risks. However, user do not have enough information to conclude that standard users read or digital investigators understand these permissions and their implications. Digital investigators need to be on the alert for the presence of malware when examining Android devices, and can benefit from supporting tools that help them understand the capabilities of such malicious code. Android applications are packaged as Android application (APK) files which contain manifest file (AndroidManifest.xml), compiled Java classes and application resources. The system developed uses static analysis techniques based on permissions in order to characterize and extract profiles for Android applications [8].

Liang and Du (2014) With the increase use of Android mobile phones, more Android malwares are being developed. Android malware detection becomes a crucial task. A permission-combination-based scheme for Android malware detection. The Android malware detection scheme is based on permission combinations declared in the application manifest file. We obtain the permission combinations that are requested frequently by malwares but rarely by benign applications. We generate rule sets based on the permission combinations. [9]

2.5 AES Encryption

The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.

AES, or Advanced Encryption Standards, is a cryptographic cipher that is responsible for a large amount of the information security that you enjoy on a daily basis. Applied by everyone from the NSA to Microsoft to Apple, AES is one of the most important cryptographic algorithms being used in 2018. AES or Advanced Encryption Standards (also known as Rijndael) is one of the most widely used methods for encrypting and decrypting sensitive information in 2017 [28].

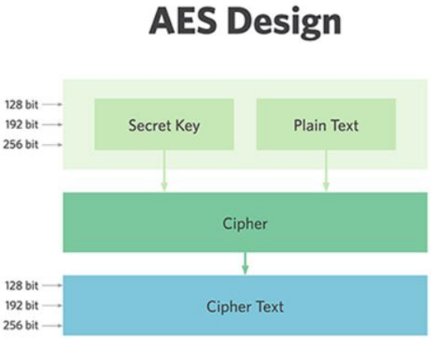


Figure 2.8 AES Design
(Source: thebestvpn.com)

AES instead Cipher on Android java class, this class provides the functionality of a cryptographic cipher for encryption and decryption. It forms the core of the Java Cryptographic Extension (JCE) framework. In order to create a Cipher object, the application calls the Cipher's getInstance method, and passes the name of the requested transformation to it. Optionally, the name of a provider may be specified. A transformation is a string that describes the operation (or set of operations) to be performed on the given input, to produce some output. A transformation always includes the name of a cryptographic algorithm (e.g., AES, DES), and may be followed by a feedback mode and padding scheme.

2.6 Android Studio

Android Studio is the official integrated development environment (IDE) for Android platform development and Android operating system by Google and JetBrains. It was announced on May 16, 2013 at the Google I/O conference. Android Studio is freely available under the Apache License 2.0. Android Studio was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. and now version 3.1.1 Based on Jet Brains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, Mac OS X and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development. The Android Studio feature are provide; Gradle-based, ProGuard integration, A rich layout, Android Virtual Device (Emulator) and so many other useful feature.

2.7 JSON

JSON is an acronym for JavaScript Object Notation an open-standard file format for human readable text to data objects, JSON used in this project to complete parsing IP Finding API information. JSON is developed to be a data interchange language which is easily read by humans and simple to use and parse by computers. JSON is “Selfdescribing” and simple to understand. In JSON, objects and arrays can be nested [25].

JSON is simple very lightweight object serialization technique or data format which is based on JavaScript Object initialization syntax, specifically array and object literals, it is essential to know the array and object literals particular JavaScript syntax. The JSON initialization code is assigned to a string and then is dealt with JavaScript eval() function or JSON parser [26].

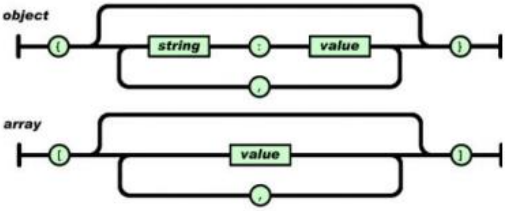
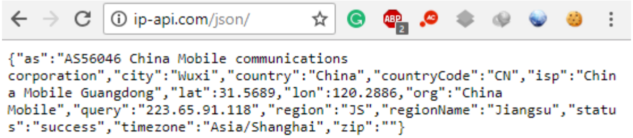


Figure 2.9 JSON Data Structure from json.org



```
{"as":"AS56046 China Mobile communications
corporation","city":"Wuxi","country":"China","countryCode":"CN","isp":"Chin
a Mobile Guangdong","lat":31.5689,"lon":120.2886,"org":"China
Mobile","query":"223.65.91.118","region":"JS","regionName":"Jiangsu","statu
s":"success","timezone":"Asia/Shanghai","zip":""}
```

Figure 2.10 An example of JSON output before parsing

2.8 UML

According to Fahad Alhumaidan, (2012) UML is a collection of diagrams for specifying various aspects such as requirements and design of software systems. It is a standard set of notations for visualizing and constructing artefacts of software systems as well as for business modelling and other non-software systems [29].

According to Sunguk Lee, (2012) Unified Modelling Language or UML is defined as a standardized general-purpose modelling language in the field of object-oriented software engineering. The UML combines techniques and processes from the data modelling (entity relationship diagrams), business modelling (work flows), object modelling, and component modelling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies [30].

2.9 Material Design

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices. The material theme provides a new style for an app, system widgets that let developers set their color palette, and default animations for touch feedback and activity transitions. To use material design in developer Android apps, Google has guidelines defined in the material design specification, support library and public library.

This application used several awesome lib which as *Android Constraint Layout*, *Custom Activity On Crash* library and *Material File Picker* dependencies and *Navigation Layout* Android.

Figure 2.11 Material Design
(Source: thenextweb.com)

2.10 Extreme Programming

Extreme Programming (XP) is a software engineering methodology, the most popular among several agile software development methodologies. XP prescribes a set of day-today practices for managers and developers [31].

Extreme Programming has the shortest iterative cycles among other Agile methodologies. Usually they last only one week. That is why XP developers have invented lots of new practices like pair programming and planning game to raise the productivity of their work [32].

There are the advantages and disadvantages of Extreme Programming:

a. Advantages

The greatest advantage of Extreme Programming is that this methodology allows software development companies to save costs and time required for project realization. Time savings are available because of the fact that XP focuses on timely delivery of final products. Extreme Programming teams save lots of money because they don’t use too much documentation. They usually solve problems through discussions inside of the team.

Simplicity is one more advantage of Extreme Programming projects. The developers who prefer to use this methodology create extremely simple code that can be improved at any moment.

b. Disadvantages

Additionally, in XP projects the defect documentation is not always good. Lack of defect documentation may lead to occurrence of similar bugs in the future. One more disadvantage of XP is that this methodology does not measure code quality assurance. It may cause defects in the initial code [32].

There are four basic activities that XP proposes for software development process:

a. Planning

The first phase of extreme programming life cycle is planning, where customers or users meet with the development team to create 'user stories' or requirements. The development team converts user stories into iterations that cover a small part of the functionality or features required. The planning team prepares the plan, time and costs of carrying out the iterations, and individual developer sign up for iterations.

b. Designing

XP's simplicity principle doesn't mean that it can exclude designing process. Without proper design in the long run system becomes too complex and projects could come to a halt. It is then important to create a design structure that organizes the logic in the system so too many dependencies in the system can be avoided.

c. Coding

In XP coding is considered the only important product of the system development process. XP programmers start to generate codes at the very beginning so "At the end of the day, there has to be a program."

d. Testing

XP emphasizes to always check if a function works is by testing it. XP uses Unit Tests which are automated tests, and the programmer will write tests as many as possible to try to break the break the code.

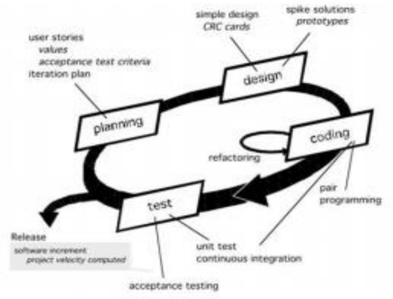


Figure 2.12 Extreme Programming Life Cycle
(Source: catatandestra.blogspot.com)

CHAPTER 3
ANALYSIS AND DESIGN

3.1 Analysis

3.1.1 Analysis of The Current System/ongoing

Based on the author experiences, experiment, observations to find any system has been use and created there are two system that used by people to against remote access Trojan, as follows:

1. Manual System
   The manual system for finding remote access Trojan by internet or article tutorial.
2. Application System
   The application system is people are using Anti-Virus.

3.1.2 Weakness of Current System

In the analysis of the current system above, the author found some problems encountered. The following is an analysis of the problems of the current System:

1. Manual System
   The manual system can take a long time and many step for finding, identifying and monitoring Android device from remote access Trojan, because it is possible that internet source tutorial different and

too much sources make a user confused. This manual system also possible to destroy the Android system, because there no experience on user.

2. Application System

The application system can take faster than manual, but antivirus system is decrease perform for some Android system. A lot of antivirus has no monitoring feature because they use automatically system and 2017 Google launch Google Play Protected as security system default of Android Play Store.

3.1.3 Literature Review

At this stage conducted literature review, which is studying reference similar research result that have been done, journal and article. It aims to obtain the theoretical foundation needed in the study.

3.1.4 Study of Literature

At this stage the author taken a data by comparing similar literature, either from literature or from the field in the form of similar applications that have been made before. The data generated at this stage are advantages and disadvantages of comparable objects.

3.1.5 Software Development Method

Software development method is a set of rules and guidelines used in the research process, planning, designing, developing, testing, setup and maintaining a software product.

There are many software development models and many organizations create and use their own model. Choosing the model has a high impact on testing. The most commonly used models are:

1. Waterfall;
2. V model;
3. Rapid Application Development;
4. Iterative model;
5. Spiral model;
6. Crystal model.

Each model has advantages and drawbacks. Agile methods are based on adaptive software development methods, while traditional SDLC models (waterfall model, for example) are based on a predictive approach. In traditional (Stoica *et al.*, 2013)

Table 3.1 Table Differences between traditional and agile development
(Source: Stoica *et al.*, 2013)

	Traditional Development	**Agile Development**
Fundamental hypothesis	Systems are fully specifiable, predictable and are developed through extended and detailed planning	High quality adaptive software is developed by small teams that use the principle of continuous improvement of design and testing based on fast feed-back and change
Management style	Command and control	Leadership and collaboration
Knowledge management	Explicit	Tacit
Communication	Formal	Informal

Development model	Life cycle model (waterfall, spiral or modified models)	*Evolutionary-delivery model*
Organizational structure	Mechanic (bureaucratic, high formalization), targeting large organization	Organic (flexible and participative, encourages social cooperation), targeting small and medium organizations
Quality control	Difficult planning and strict control. Difficult and late testing	Permanent control or requirements, design and solutions. Permanent testing
User requirements	Detailed and defined before coding/implementation	Interactive input
Cost of restart	High	Low
Development	Fixed	Easily changeable
Testing	After coding is completed	Evert iteration
Additional abilities	Nothing in particular	Interpersonal abilities and basic knowledge of the business
Client involvement	Low	High
Appropriate scale of the project	Large scale	Low and medium scale
Developers	Oriented on plan, with adequate abilities, access to external knowledge	Agile, with advanced knowledge, co-located and cooperative
Client	With access to knowledge cooperative, representative and empowered	Dedicated, knowledgeable, cooperative representative and empowered
Requirements	Very stable, know in advance	Emergent, with rapid changes

Architecture	Design for current and predictable requirements	Design for current requirements
Remodeling	Expensive	Not expensive
Size	Large team and projects	Small teams and project
Primary objective	High safety	Quick value

3.2 Conceptual Framework

This is the conceptual framework from WEDEFEND Android Application to Monitor and Protect from Remote Access Trojan.

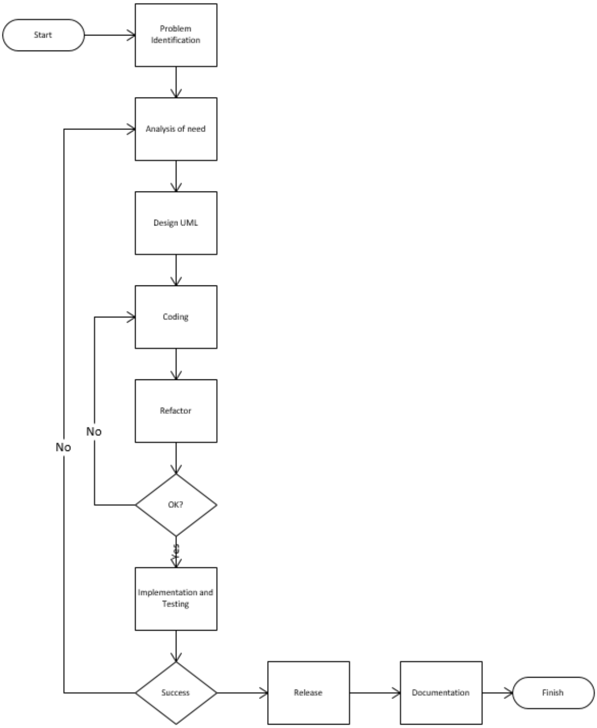


Figure 3.1 Conceptual Framework

3.3 Design

3.3.1 System Design

Based on the analysis of the problem above, system design will be illustrated in UML diagram, as follows:

1. Use Case Diagram
2. Class Diagram
3. Sequence Diagram
4. Activity Diagram

3.3.2 Use Case Diagram

Use Case diagram describe any activities by the system.

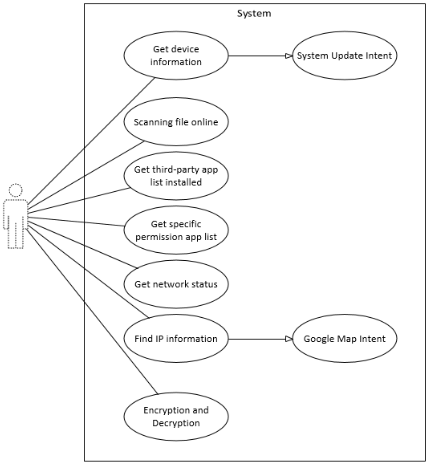


Figure 3.2 Use Case Diagram

Table 3.2 Table Use Case

No	Actor	Action	Description
1	User	Get device information	User will get device information details and update
2	User	Scanning file online	User can scan files online to detect is that malicious file
3	User	Get third-party app list installed	User can get third-party app that installed on the device
4	User	Get specific permission app list	User can get specific application list with specific permission that app use
5	User	Get network status	User can get network activity status between device and other system
6	User	Find IP Informaton	User can get information about internet protocol
7		Google map intent	After finding the ip information, system can show ip location on google map
8	User	Encryption and Decryption	User can encrypt an important file and decrypt an encrypted file

3.3.3 Class Diagram

Class diagram is part of the Entity Relationship Diagram (ERD).

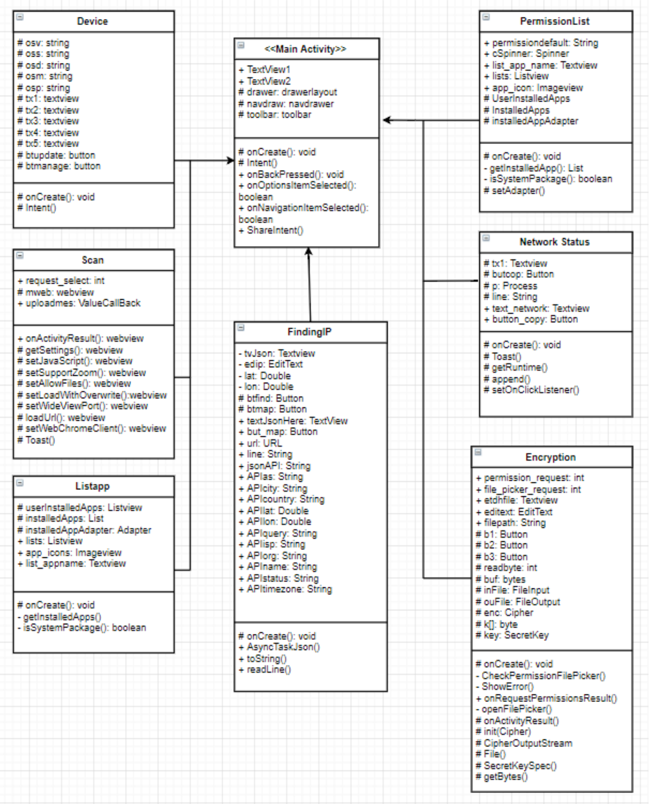


Figure 3.3 The Application Class Diagram

3.3.4 Sequence Diagram

Sequence diagram illustrates the interaction of objects arranged chronologically.

1. Sequence Diagram Get Device Information

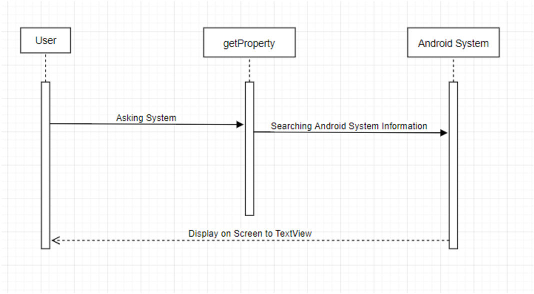

User
getProperty
Android System
Asking System
Searching Android System Information
Display on Screen to TextView

Figure 3.4 Sequence Diagram Get Device Information

2. Sequence Diagram Scan Online

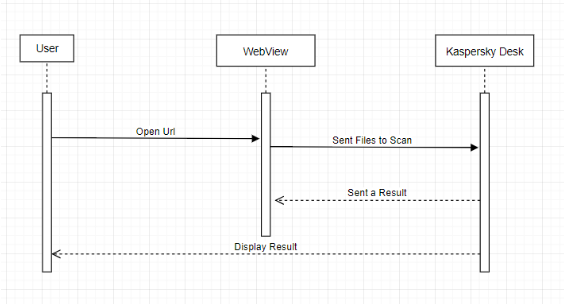


Figure 3.5 Sequence Diagram Scan File Online

3. Sequence Diagram Get App List

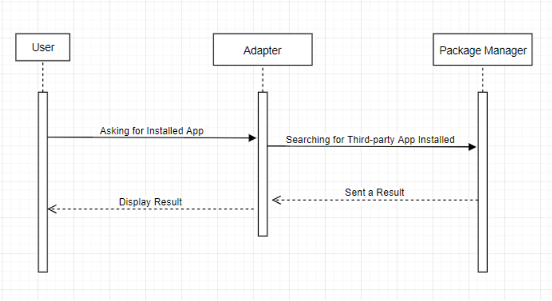


Figure 3.6 Sequence Diagram Get App List

4. Sequence Diagram Permission List

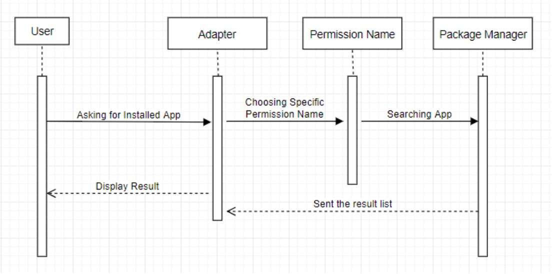


Figure 3.7 Sequence Diagram Permission List

5. Sequence Diagram Network Status

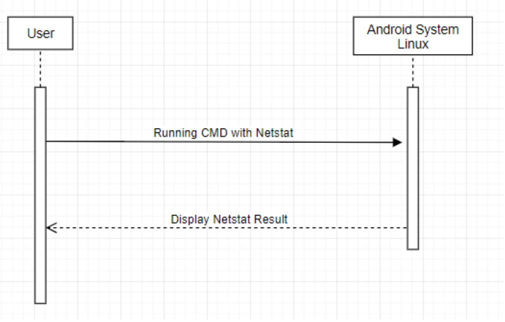


Figure 3.8 Sequence Diagram Network Status

6. Sequence Diagram Geolocation

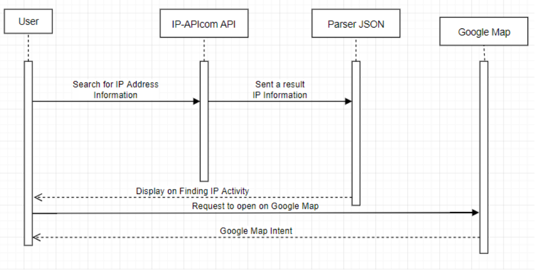

User
IP-APIcom API
Parser JSON
Google Map
Search for IP Address Information
Sent a result IP Information
Display on Finding IP Activity
Request to open on Google Map
Google Map Intent

Figure 3.9 Sequence Diagram Finding IP

7. Sequence Diagram Encryption

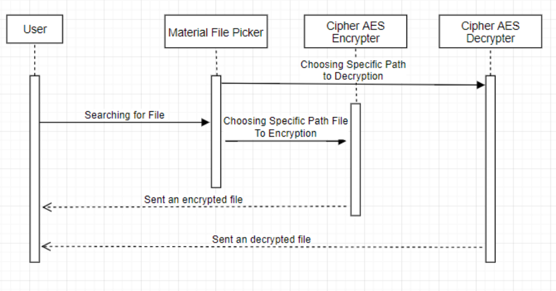


Figure 3.10 Sequence Diagram Encryption

3.3.5 Activity Diagram

Activity diagram is similar to flowchart, because it can model a workflow from one activity to another. Here is some activity diagram for this thesis:

1. Activity Diagram Main Page

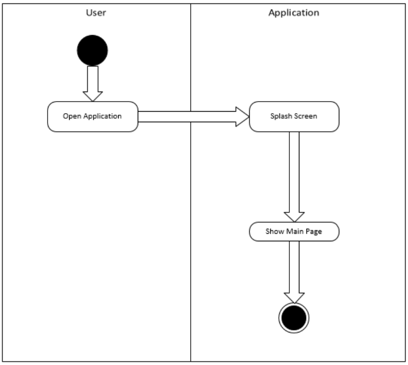


Figure 3.11 Activity Diagram Main Page

From the diagram above, the main activity starts with a splash screen and the application will show main page.

2. Activity Diagram My Device

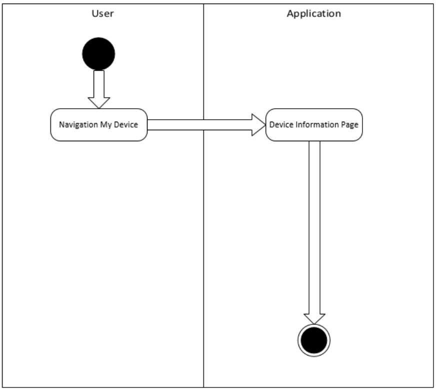


Figure 3.12 Activity Diagram My Device

When the user presses My Device navigation drawer, the application will show device information such as Operating System, Hardware, Model, Product and Serial. The application also has system update and unknown source button on setting for security measure.

3. Activity Diagram Scan File

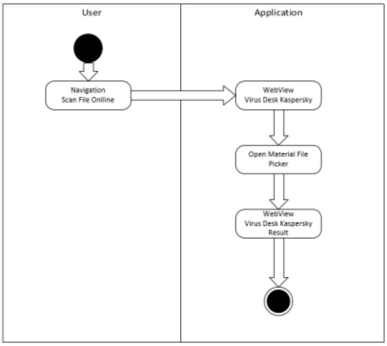

User
Application
Navigation Scan File Online
WebView Virus Desk Kaspersky
Open Material File Picker
WebView Virus Desk Kaspersky Result

Figure 3.13 Activity Diagram Scan File Online

When the user presses Scan File Online, the application will show Virus Desk Kaspersky use WebView function and if user want to scan file the application will open Material File Picker to choose a file.

4. Activity Diagram Check App

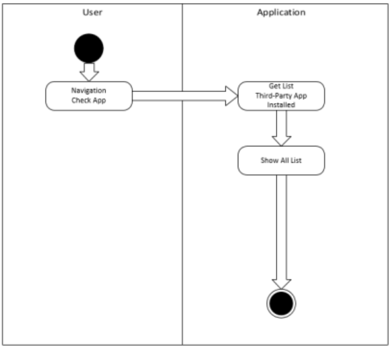


Figure 3.14 Activity Diagram Check App

User will get third-party app list that installed on Android device.

5. Activity Diagram Permission List

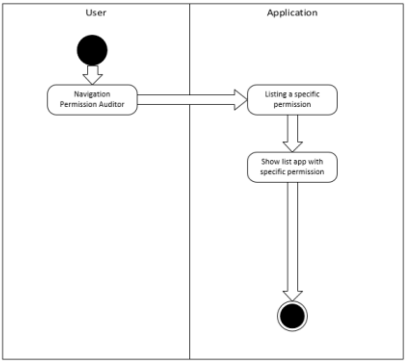

User
Application
Navigation Permission Auditor
Listing a specific permission
Show list app with specific permission

Figure 3.15 Activity Diagram Permission Auditor

User will get application list installed on Android that already categorize by specific permission.

6. Activity Diagram Network Status

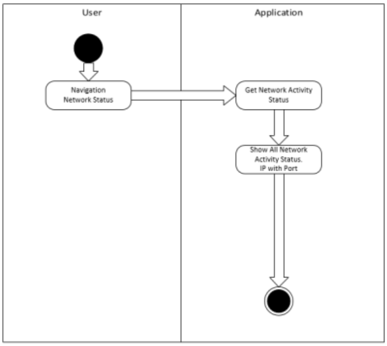

User
Application
Navigation Network Status
Get Network Activity Status
Show All Network Activity Status. IP with Port

Figure 3.16 Activity Diagram Network Status

When the user presses Network Status feature, the application will get all network activity in Android device and display on screen. The application gave a copy clipboard button to copy all network activity for next usage.

7. Activity Diagram Finding IP

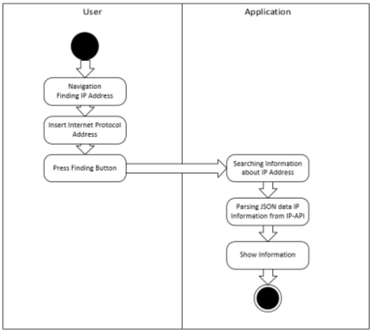

User
Application
Navigation Finding IP Address
Insert Internet Protocol Address
Press Finding Button
Searching Information about IP Address
Parsing JSON data IP Information from IP-API
Show Information

Figure 3.17 Activity Diagram Finding IP Address

The application will display a Finding IP Address page, and user insert some internet protocol address to get information about it. The application will search about IP Address, parsing it from JSON data and show all information to user.

8. Activity Diagram Encryption

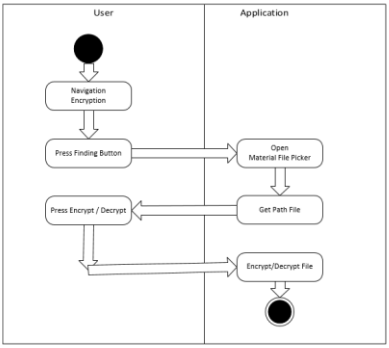


Figure 3.18 Activity Diagram Encryption

The application will display encryption feature, the user can encrypt a file to protect from stealer and decrypt an encrypted file.

9. Activity Diagram About

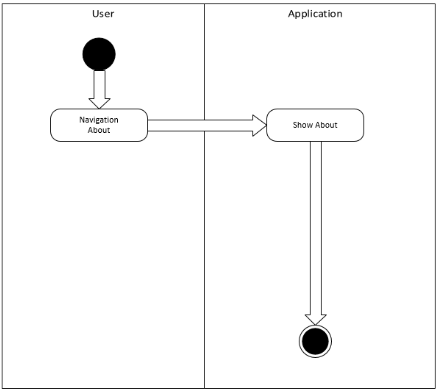


Figure 3.19 Activity Diagram About

User can read about WeDefend application information, creator.

10. Activity Diagram Share

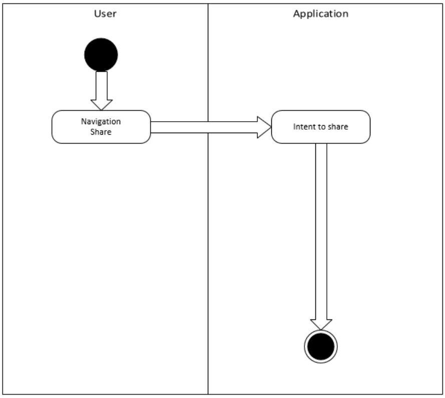


Figure 3.20 Activity Diagram Share

The user can share about the application using this navigation feature, the application will help the user to which application the user want to share.

3.4 Construction

After the Design of the application, the next step is to develop the application based on the design that has been created.

3.4.1 Programming

Software used to develop this application is Android Studio with Android Operating System users with Java programming language and XML. In addition, Android Navigation Drawer Layout, Material File Picker and Custom Activity on Crash also used for displaying some activity in this application.

3.4.2 Implementation

After the construction of the application is complete, then the next step is implementation. This step includes Application testing which is the last stage in XP (*Extreme Programming*) development method before release. At this stage, the system is ready to be tested. The test was performed using Android emulator and *Blackbox testing* method. The *Blackbox testing* method performs testing regardless of the program source code is run by the tester or user to observe the program whether it has received input, processed, and generated the appropriate output.

CHAPTER 4
IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Determine the Devices Used

For developing this application, the author uses some devices software and hardware. Software that used in the development of this application is as follows:

1. Android Development Tools	: Android Studio 3.0.1
2. Android Virtual Device	: Android 8.0
3. Memu Emulator	: Android 5.1

Android Studio is used for writing program code and to test program in Android Virtual Device before the program is tested on real devices.

4.1.2 Hardware environment specification

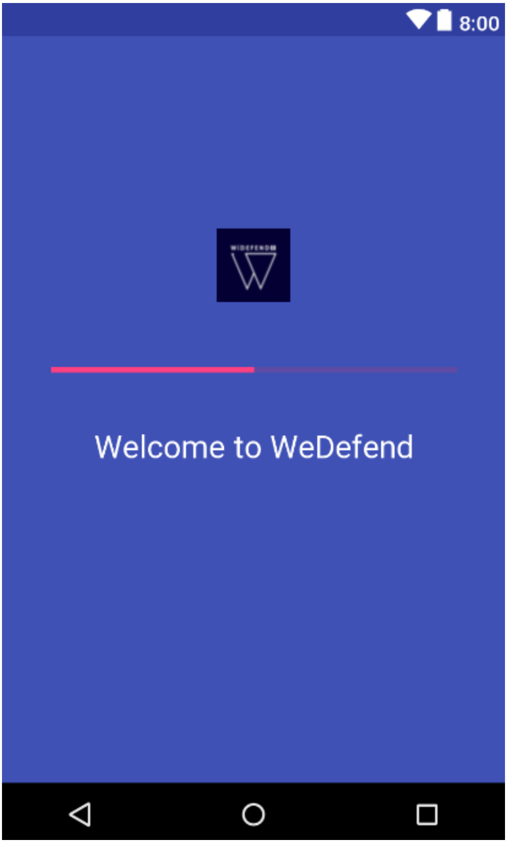
Hardware that used in the development of this application is as follows:

1. Notebook

Brand	: Lenovo Y700 15-ISK
Processor	: Intel® Core™ i5-6300HQ
CPU	: @2.30 GHz
RAM	: 8 GB
VGA	: Intel HD Graphic 530 & Nvidia GTX 960M 2GB
Hard Disk	: 500 GB
OS	: Windows 10 64bit

4.1.3 User Interface

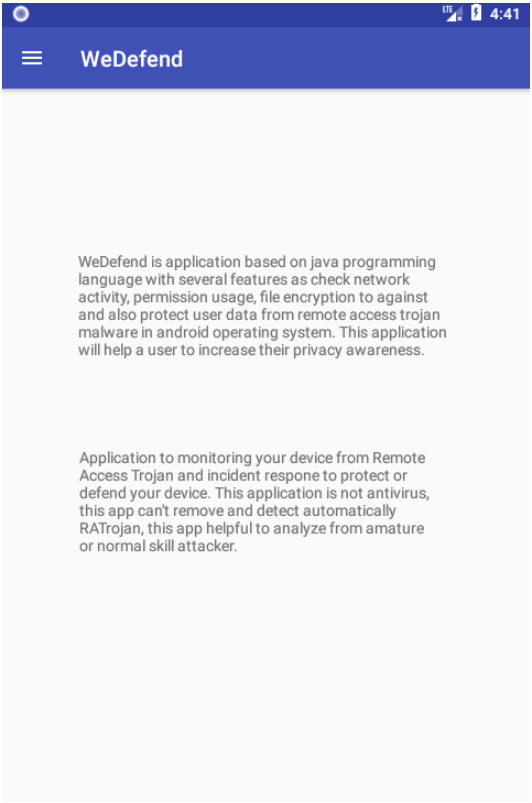
a. Splash Screen



Welcome to WeDefend

Figure 4.1 Splash Screen

b. Main Page

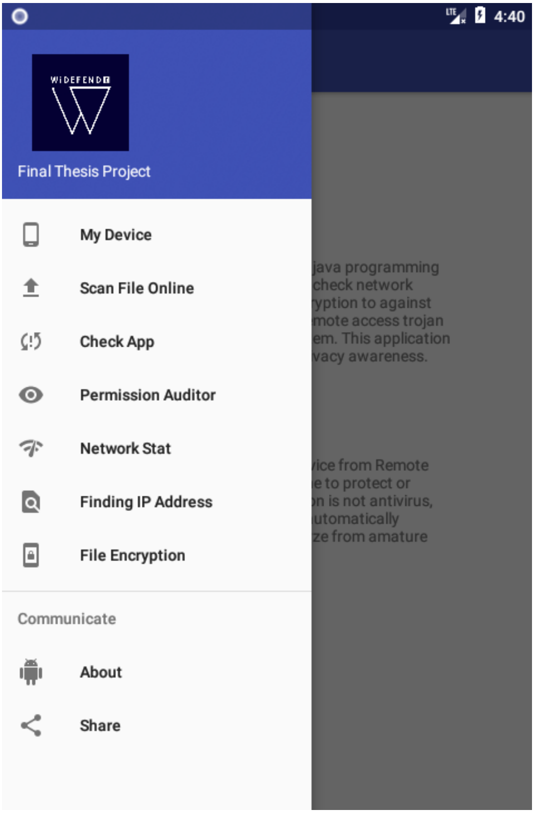


WeDefend is application based on java programming language with several features as check network activity, permission usage, file encryption to against and also protect user data from remote access trojan malware in android operating system. This application will help a user to increase their privacy awareness.

Application to monitoring your device from Remote Access Trojan and incident respone to protect or defend your device. This application is not antivirus, this app can't remove and detect automatically RATrojan, this app helpful to analyze from amature or normal skill attacker.

Figure 4.2 Main Page

c. Navigation Drawer

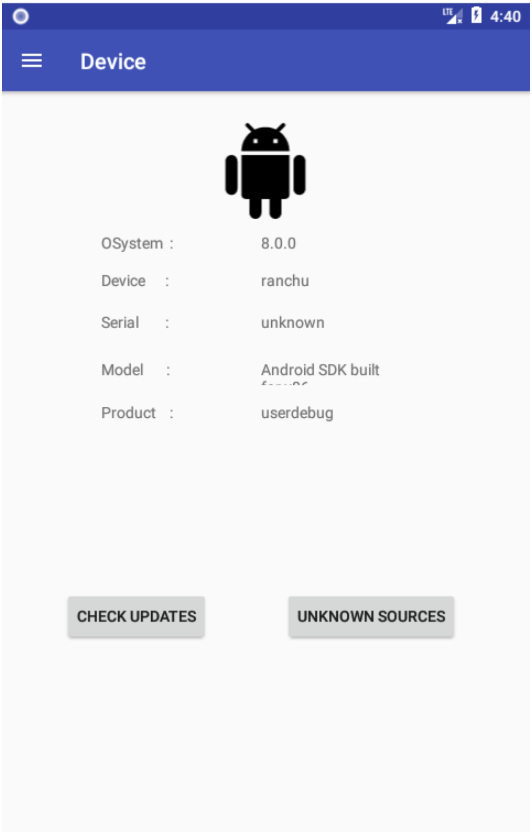


java programming
check network
yption to against
mote access trojan
em. This application
vacy awareness.

vice from Remote
e to protect or
n is not antivirus,
utomatically
ze from amature

Figure 4.3 Navigation Drawer

d. My Device Page



OSystem : 8.0.0

Device : ranchu

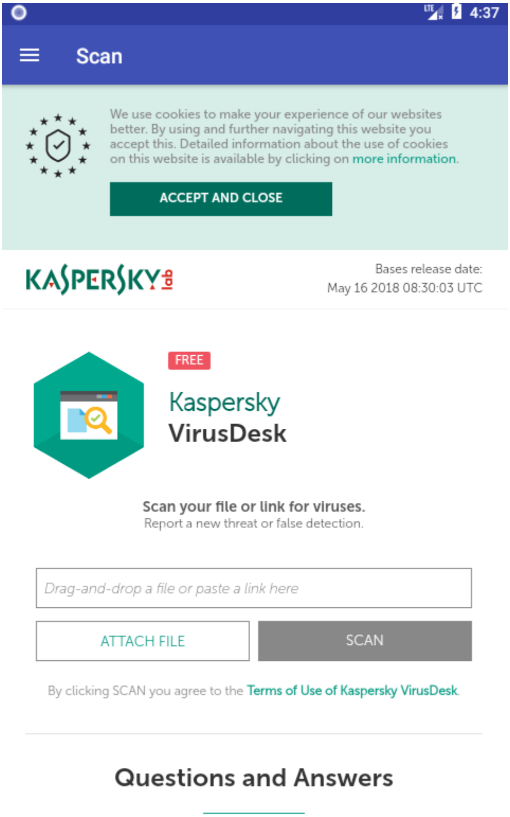
Serial : unknown

Model : Android SDK built

Product : userdebug

Figure 4.4 My Device

e. Scan File Online



We use cookies to make your experience of our websites better. By using and further navigating this website you accept this. Detailed information about the use of cookies on this website is available by clicking on more information.

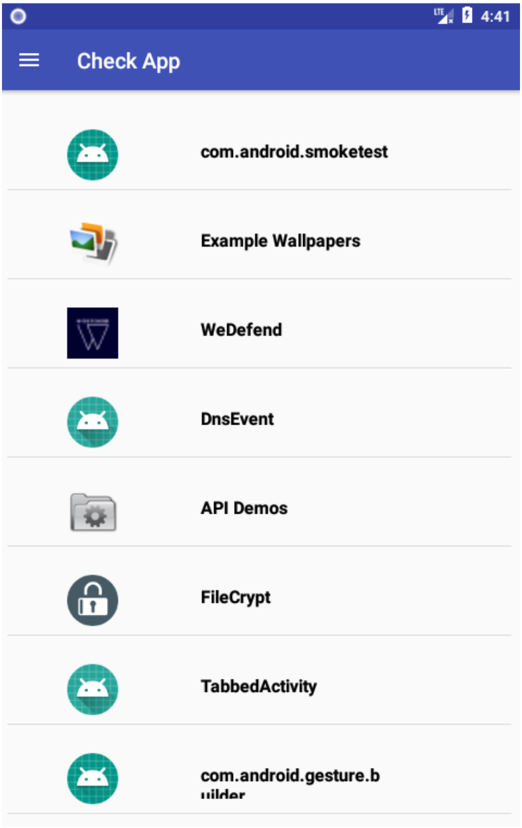
Bases release date:
May 16 2018 08:30:03 UTC

Scan your file or link for viruses.
Report a new threat or false detection.

By clicking SCAN you agree to the Terms of Use of Kaspersky VirusDesk.

Figure 4.5 Scan File Online

f. Check Third App



4:41

Check App

com.android.smoketest

Example Wallpapers

WeDefend

DnsEvent

API Demos

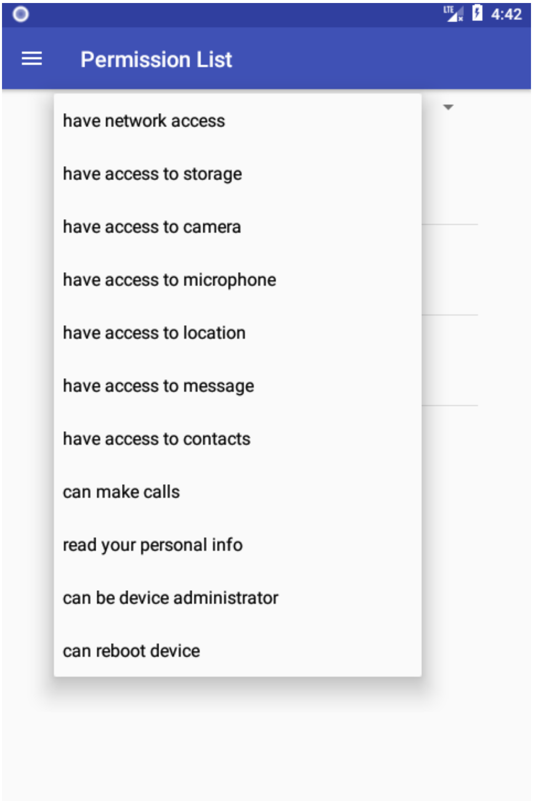
FileCrypt

TabbedActivity

com.android.gesture.builder

Figure 4.6 Check Third Party App

g. Permission Auditor



Permission List

have network access

have access to storage

have access to camera

have access to microphone

have access to location

have access to message

have access to contacts

can make calls

read your personal info

can be device administrator

can reboot device

Figure 4.7 Permission List

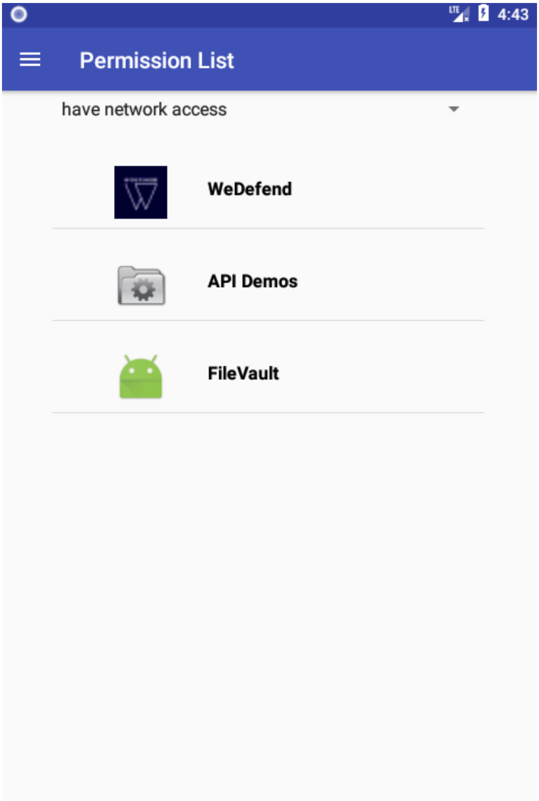
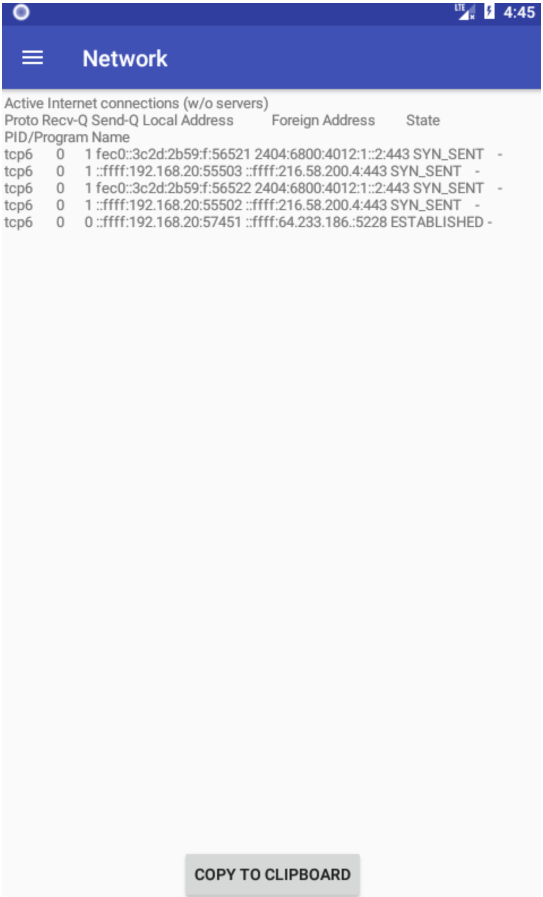


Figure 4.8 Permission Auditor

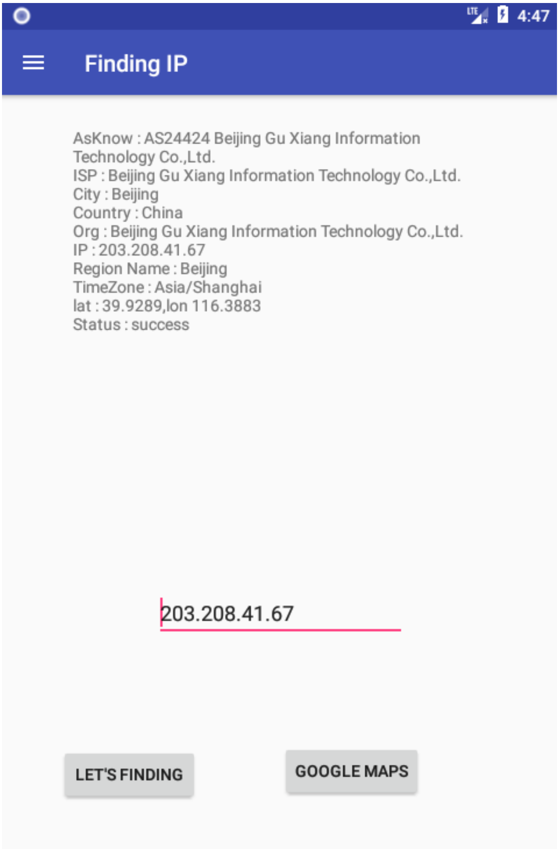
h. Network Stat



```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
PID/Program Name
tcp6       0      1 fec0::3c2d:2b59:f:56521 2404:6800:4012:1::2:443 SYN_SENT   -
tcp6       0      1 ::ffff:192.168.20:55503 ::ffff:216.58.200.4:443 SYN_SENT   -
tcp6       0      1 fec0::3c2d:2b59:f:56522 2404:6800:4012:1::2:443 SYN_SENT   -
tcp6       0      1 ::ffff:192.168.20:55502 ::ffff:216.58.200.4:443 SYN_SENT   -
tcp6       0      0 ::ffff:192.168.20:57451 ::ffff:64.233.186.:5228 ESTABLISHED -
```

Figure 4.9 Network Status

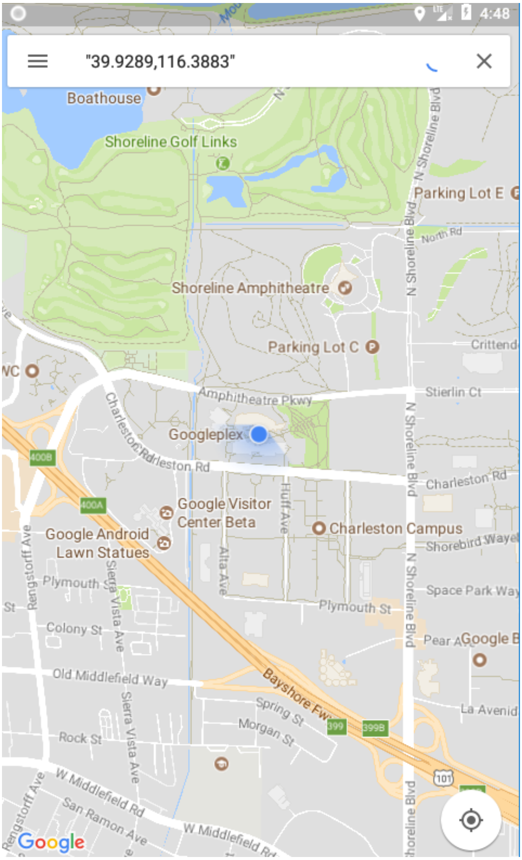
i. Finding Internet Protocol Address



AsKnow : AS24424 Beijing Gu Xiang Information Technology Co.,Ltd.
ISP : Beijing Gu Xiang Information Technology Co.,Ltd.
City : Beijing
Country : China
Org : Beijing Gu Xiang Information Technology Co.,Ltd.
IP : 203.208.41.67
Region Name : Beijing
TimeZone : Asia/Shanghai
lat : 39.9289,lon 116.3883
Status : success

203.208.41.67

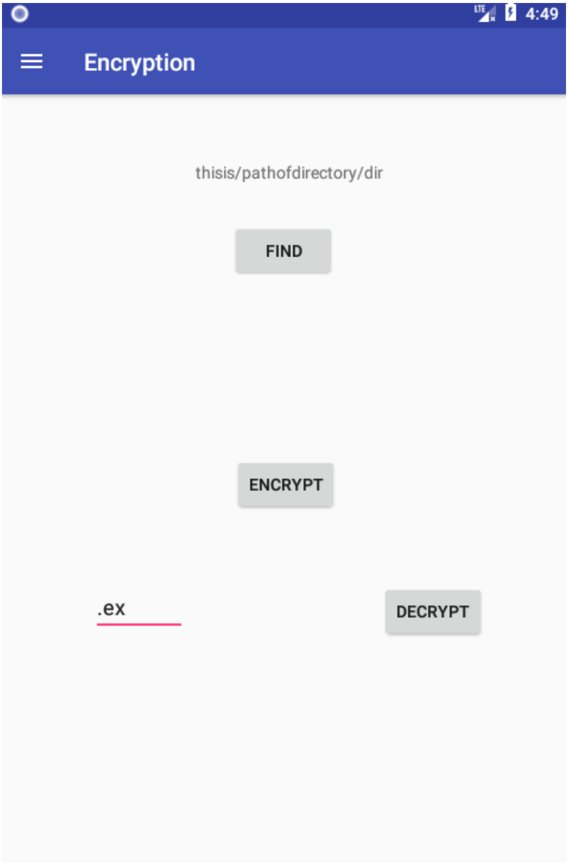
Figure 4.10 Find Internet Protocol Address



"39.9289,116.3883"

Figure 4.11 Google Map from Internet Protocol Address

j. File Encryption

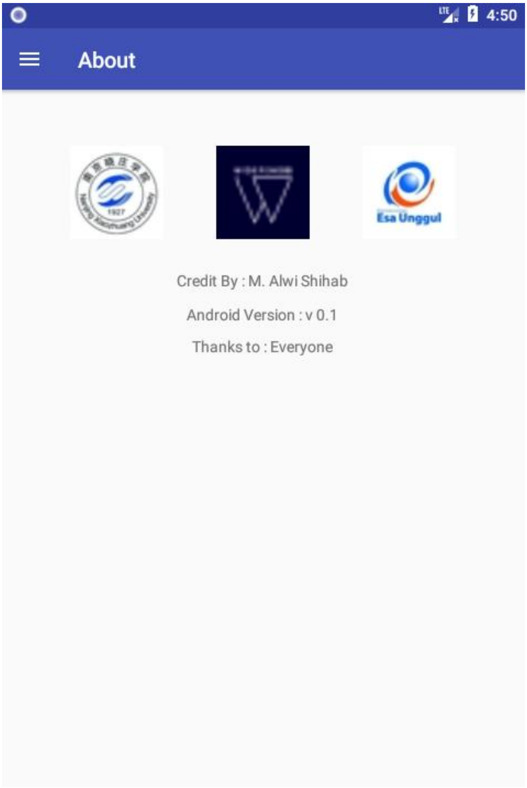


thisis/pathofdirectory/dir

.ex

Figure 4.12 File Encryption

k. About



About

Credit By : M. Alwi Shihab

Android Version : v 0.1

Thanks to : Everyone

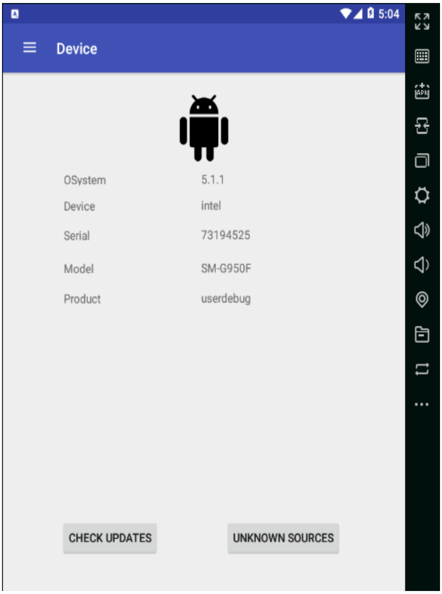
Figure 4.13 About

4.2 Testing

In this phase, the authors focus on testing application on Memu Emulator that already Infected by Remote Access Trojan. The goal is to identify and find out Trojan using all feature in the application. Such to scan file virus, find unknown application, permission usage on the application, network status, find information of address and open on google map and encrypt some important file.

4.2.1 My Device

The My Device will display a system information of device. Such Operation System, device, serial, model, product and button to check updates of system and also setting unknown source security install.

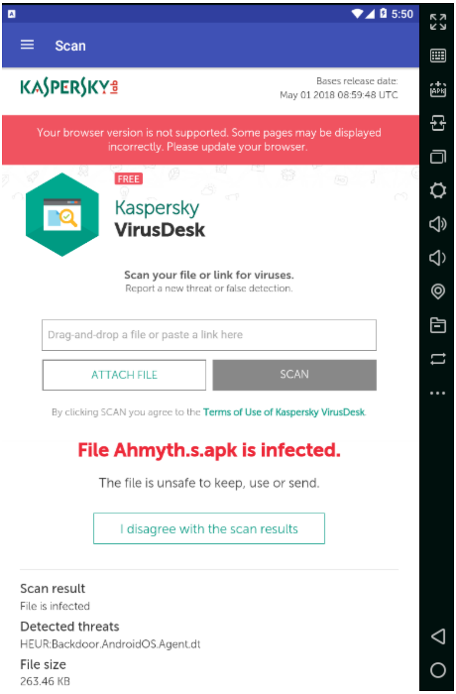


OSystem	5.1.1
Device	intel
Serial	73194525
Model	SM-G950F
Product	userdebug

Figure 4.14 Testing My Device

4.2.2 Scan File Online

The Scan File Online will lead user to open Kaspersky online scanner and scan file before use it.



Bases release date:
May 01 2018 08:59:48 UTC

Your browser version is not supported. Some pages may be displayed incorrectly. Please update your browser.

Scan your file or link for viruses.
Report a new threat or false detection.

Drag-and-drop a file or paste a link here

By clicking SCAN you agree to the Terms of Use of Kaspersky VirusDesk

File Ahmyth.s.apk is infected.

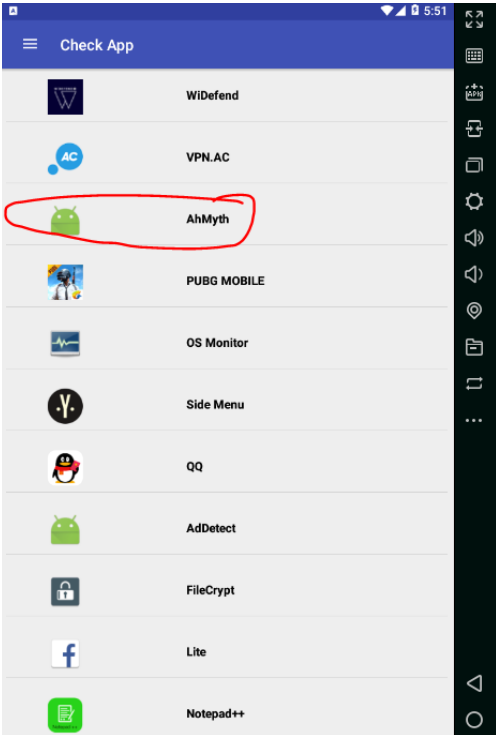
The file is unsafe to keep, use or send.

Scan result
File is infected
Detected threats
HEUR:Backdoor.AndroidOS.Agent.dt
File size
263.46 KB

Figure 4.15 Testing Scan Online

4.2.3 Check App

Showing all third-party app installed (nonsystem), to help user aware from unknown application. In this case, the author don’t know about AhMyth application and feel like never install it. The best solution is to uninstall this application.



5:51

Check App

WiDefend

VPN.AC

AhMyth

PUBG MOBILE

OS Monitor

Side Menu

QQ

AdDetect

FileCrypt

Lite

Notepad++

Figure 4.16 Testing Check App

4.2.4 Permission Auditor

Showing all third-party app which specific permission used on that app. In this case, the author know about AhMyth app but author though this app didn’t need to access camera: means over permission usage.

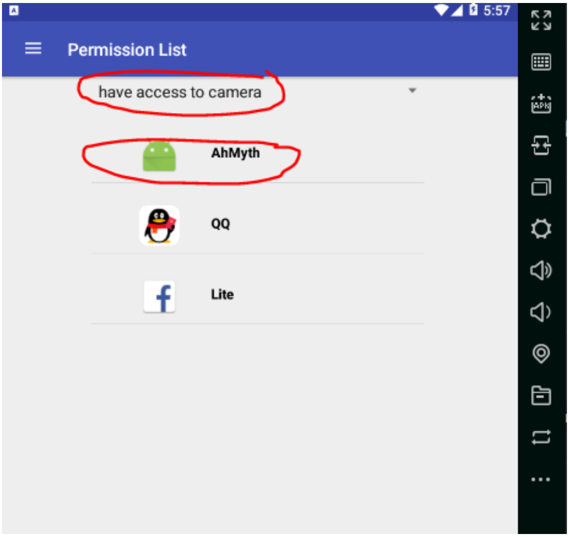
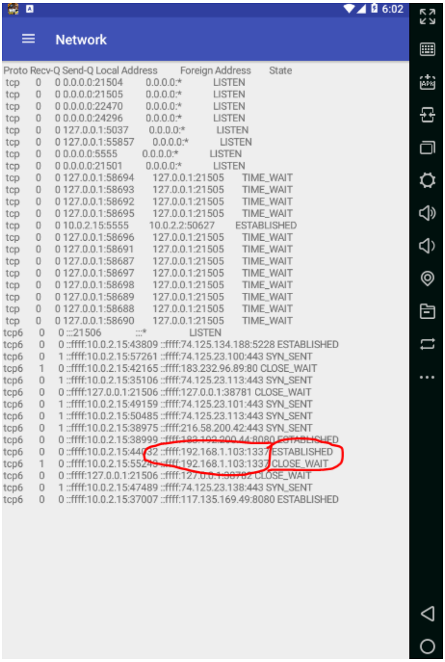


Figure 4.17 Testing Permission Auditor

4.2.5 Network Stat

This feature to show all connection or communication between android system and other source, with Internet Protocol Address and Port used. In this case example, the attacker Internet Protocol Address 192.168.1.103 and Port 1337 (it configure on attacker). Helping a user to aware from unknown Internet Protocol Address, Port. The user can copy to clipboard all network status output for next usage.

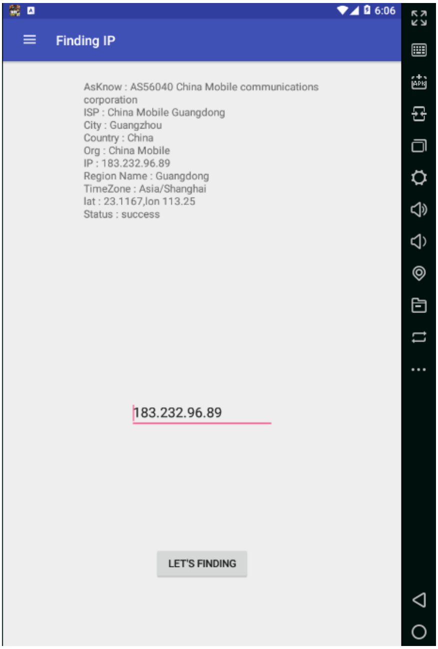


```
Proto Recv-Q Send-Q Local Address        Foreign Address       State
 tcp      0      0 0.0.0.0:21504        0.0.0.0:*            LISTEN
 tcp      0      0 0.0.0.0:21505        0.0.0.0:*            LISTEN
 tcp      0      0 0.0.0.0:22470        0.0.0.0:*            LISTEN
 tcp      0      0 0.0.0.0:24296        0.0.0.0:*            LISTEN
 tcp      0      0 127.0.0.1:5037       0.0.0.0:*            LISTEN
 tcp      0      0 127.0.0.1:55857      0.0.0.0:*            LISTEN
 tcp      0      0 0.0.0.0:5555         0.0.0.0:*            LISTEN
 tcp      0      0 0.0.0.0:21501        0.0.0.0:*            LISTEN
 tcp      0      0 127.0.0.1:58694      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58693      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58692      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58695      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 10.0.2.15:5555       10.0.2.2:50627       ESTABLISHED
 tcp      0      0 127.0.0.1:58696      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58691      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58687      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58697      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58698      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58689      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58688      127.0.0.1:21505      TIME_WAIT
 tcp      0      0 127.0.0.1:58690      127.0.0.1:21505      TIME_WAIT
tcp6      0      0 :::21506             :::*                 LISTEN
tcp6      0      0 ::ffff:10.0.2.15:43809 ::ffff:74.125.134.188:5228 ESTABLISHED
tcp6      0      1 ::ffff:10.0.2.15:57261 ::ffff:74.125.23.100:443 SYN_SENT
tcp6      1      0 ::ffff:10.0.2.15:42165 ::ffff:183.232.96.89:80 CLOSE_WAIT
tcp6      0      1 ::ffff:10.0.2.15:35106 ::ffff:74.125.23.113:443 SYN_SENT
tcp6      0      0 ::ffff:127.0.0.1:21506 ::ffff:127.0.0.1:38781 CLOSE_WAIT
tcp6      0      1 ::ffff:10.0.2.15:49159 ::ffff:74.125.23.101:443 SYN_SENT
tcp6      0      1 ::ffff:10.0.2.15:50485 ::ffff:74.125.23.113:443 SYN_SENT
tcp6      0      1 ::ffff:10.0.2.15:38975 ::ffff:216.58.200.42:443 SYN_SENT
tcp6      0      0 ::ffff:10.0.2.15:38999 ::ffff:183.192.200.44:8080 ESTABLISHED
tcp6      0      0 ::ffff:10.0.2.15:44032 ::ffff:192.168.1.103:1337 ESTABLISHED
tcp6      1      0 ::ffff:10.0.2.15:55249 ::ffff:192.168.1.103:1337 CLOSE_WAIT
tcp6      0      0 ::ffff:127.0.0.1:21506 ::ffff:127.0.0.1:38782 CLOSE_WAIT
tcp6      0      1 ::ffff:10.0.2.15:47489 ::ffff:74.125.23.138:443 SYN_SENT
tcp6      0      0 ::ffff:10.0.2.15:37007 ::ffff:117.135.169.49:8080 ESTABLISHED
```

Figure 4.18 Testing Network Status

4.2.6 Finding Internet Protocol Address

This feature to trace and find Internet Protocol information detail. In this case, I took a sample IP 183.232.96.89 from network status.



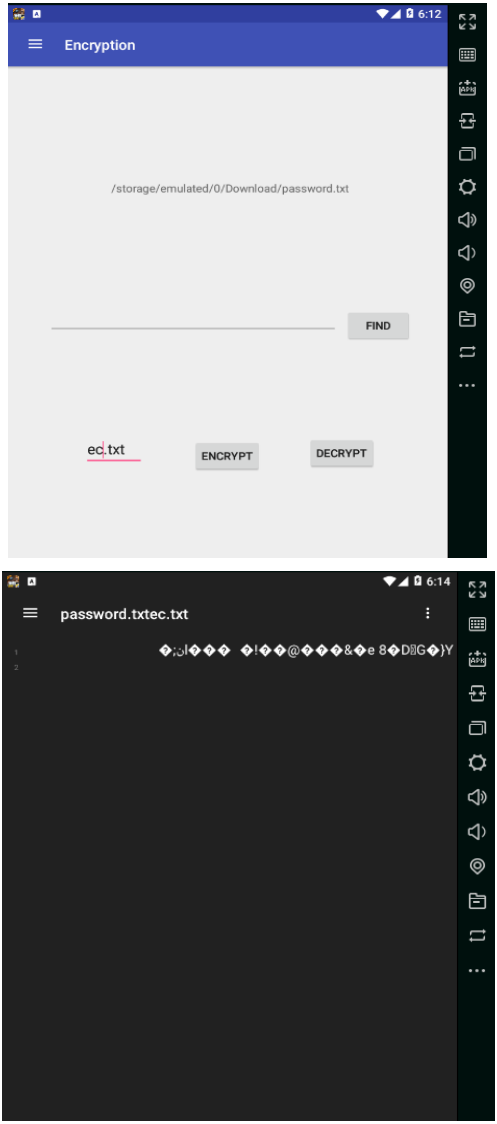
AsKnow : AS56040 China Mobile communications corporation
ISP : China Mobile Guangdong
City : Guangzhou
Country : China
Org : China Mobile
IP : 183.232.96.89
Region Name : Guangdong
TimeZone : Asia/Shanghai
lat : 23.1167,lon 113.25
Status : success

183.232.96.89

Figure 4.19 Testing Finding IP

4.2.7 File Encryption

Encrypting an important file, such as txt jpg and other to protect from unwanted user usage. In this case, I’ve file password.txt with text inside “*this is my hard password: 123securepassword*“ and try to encrypt it with same extension.



/storage/emulated/0/Download/password.txt

ec.txt

```
�;ناl���  �!��@����&�e 8�DG�}Y
```

Figure 4.20 Testing File Encryption

4.2.8 Black Box Testing

Table 4.1 Black Box Testing

No	Process Design	Expected Result	Test Result	Information
1	Click Icon Launcher	Displaying Splash Screen and Main Page	Success	Can be seen in **Figure 4.1 & 4.2**
2	Click My Device	Displaying Device Information	Success	Can be seen in **Figure 4.4**
3	Click Scan File Online	Displaying Result of Kaspersky Online Scanner	Success	Can be seen in **Figure 4.5**
4	Scanning File	Displaying Result of Kaspersky Scanner	Success	Can be seen in **Figure 4.15**
5	Check App	Displaying Third-Party Application List	Success	Can be seen in **Figure 4.6**
6	Permission Auditor	Displaying a Third-party Application List with Specific Permission	Success	Can be seen in **Figure 4.8**
7	Network Status	Displaying Network Activity	Success	Can be seen in **Figure 4.9**
8	Click Copy Clipboard Network Status Result	Copy Network Status Result to Clipboard	Success	
9	Click Finding IP Address	Found Result of IP Information	Success	Can be seen in **Figure 4.10**
10	Click Open to Google Map	Display IP Location on Google Map	Success	Can be seen in **Figure 4.11**
11	Encryption	A File Unreadable / Encrypted	Success	Can be seen in **Figure 4.20**
12	Decryption	A File Readable / Decrypting Encrypted File	Success	

CHAPTER 5
CONCLUSIONS AND SUGGESTION

5.1 Conclusion

This thesis aims to help people monitoring and protecting their Android device from remote access Trojan, the applications very useful for users whom want to identify and find manually if the system infected or not. This application can be concluded as follows:

1. Identify current system information and easy lead users to update manually
2. Scanning files before use it, is the first aid method to identify malware
3. Users can be easy to check list third-party app installed in device
4. Permission Auditor is the easy way to categorize third-party application on specific permission for users
5. Network Status is the best way to identify device infected by remote access Trojan
6. This application help the users to trace or find sources Internet Protocol Address detail information, very useful for identification
7. Encryption is very helpful for users who want protect their sensitive files
8. This application is not difficult to use for Non-experienced users

5.2 Suggestion

This application is still has a weakness and shortcomings, therefore this application still requires further development to improve the features in this application. Here are some suggestions for further development of the application:

1. This application only scan by online scanner. It is better if this application can scan offline for specific remote access Trojan virus.
2. This application only shows all the list third-party installed on device. It is better if the application show detail information of third-party app and can terminate process and uninstall the app.
3. This application network status feature only shows connection between device and other connection, it is better if this feature can tell which application use specific Internet Protocol address, port and categorize with whitelist address.
4. Encryption feature only support for a few extension and size files. It is better if this application support for all extension and all size and much better if this application has feature files vault to encrypt, save and open inside app.

REFERENCES

[1] I. Burguera, U. Zurutuza and S. Nadjm-Tehrani, "Crowdroid: Behavior-based Malware Detection System for Android," *ACM,* 2011.

[2] W. Te-En, M. Ching-Hao, A. B. Jeng, W. Horng-Tzer and W. Dong-Jie, "Android Malware Detection via a Latent Network Behavior Analysis," *IEEE,* 2012.

[3] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. G. Bringas and G. Alvarez, "PUMA: Permission Usage to Detect Malware in Android," *Springer-Verlag,* 2013.

[4] A. Arora, S. Grag and S. K. Peddoju, "Malware Detection Using Network Traffic Analysis in Android Based Mobile Devices," *IEEE,* 2014.

[5] D. Arp, H. Gascon and K. Rieck, "DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket," *Internet Society,* 2014.

[6] Z. Aung and W. Zaw, "Permission-Based Android Malware Detection," *International Journal of Scientific & Technology Research,* vol. 2, no. 3, 2013.

[7] T. Isohara, K. Takemori and A. Kubota, "Kernel-based Behavior Analysis for Android Malware Detection," *IEEE,* 2011.

[8] K. A. Talha, D. I. Alper and C. Aydin, "APK Auditor: Permission-based Android Malware Detection System," *Digital Investigation,* 2015.

[9] S. Liang and X. Du, "Permission-Combination-based Scheme for Android," *IEEE,* 2014.

[10] S. Y. Yerima, S. Sezer and I. Muttik, "High Accuracy Android Malware Detection using Ensemble Learning," *IET Information Security,* 2015.

[11] L. M. Acosta-Guzman, G. Aguilar-Torres and G. Gallegos-Garcia, "Network Activity Monitoring Against Malware in Android," *International Journal of Electrical and Computer Engineering,* 2016.

[12] H. A. Lashkari, A. F. Kadir, H. Gonzalez, K. F. Mbah and A. A. Ghorbani, "Towards a Network-Based Framework for Android Malware Detection and," *CIC,* 2014.

[13] Z. Liu, Y. Li, H. Yang and J. Qiu, "A Case Study on Key Technologies of Android Trojan," *IEEE,* 2014.

[14] A. Shabtai, U. Kanonov, Y. Elovici, C. Glezer and Y. Weiss, "Andromaly: A Behavioral Malware Detection Framework for Android Devices," *J Intell Information System,* 2012.

[15] W. Dong-Jie, M. Ching-Hao, W. Te-En, L. Hahn-Ming and W. Kuo-Ping, "DroidMat: Android Malware Detection through Manifest and API Calls Tracing," *IEEE,* 2012.

[16] Z. Yuan, Y. Lu, Z. Wang and Y. Xue, "Droid-Sec: Deep Learning in Android Malware Detection," *SIGCOMM,* 2014.

[17] A. Shabtai, L. Tenenboim-Chekina, D. Mimran, L. Rokach, B. Shapira and Y. Elovici, "Mobile Malware Detection through Analysis of Deviations in Application Network Behavior," *Elsevier,* 2014.

[18] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-an and H. Ye, "Significant Permission Identification for Machine Learning Based Android Malware Detection," *IEEE,* 2017.

[19] M. Grace, Y. Zhuo, Q. Zhang, S. Zuo and X. Jiang, "RiskRanker: Scalable and Accurate Zero-day Android Malware Detection," *ACM,* 2012.

[20] H. Gascon, F. Yamaguchi and D. Arp, "Structural Detection of Android Malware using Embedded Call Graphs," *ACM,* 2013.

[21] K. O. Elish, X. Shu, D. Yao, B. G. Ryder and X. Jiang, "Profiling User-Trigger Dependence for Android Malware Detection," *IEEE,* 2013.

[22] G. Dini, F. Martinelli, A. Saracino and D. Sgandurra, "MADAM: A Multi-Level Anomaly Detector for Android Malware," *IIT,* 2012.

[23] J. Sahs and L. Khan, "A Machine Learning Approach to Android Malware Detection," *IEEE,* 2012.

[24] S. Bhardwaj, P. Chauhan and R. Sharma, "Android Operating System," *International Journal of Engineering Technology and Management System,* 2013.

[25] B. N. Rupa, G. K. Mohan, J. S. Babu and T. H. Kim, "Test Report Generation Using JSON," *International Journal of Software Engineering and Its Applications,* 2015.

[26] Z. U. Haq, G. F. Khan and T. Hussain, "A Comprehesive Analysis of XML and JSON Web Technologies," *New Dovelopments in Circuits, System, Signal Processing, Communications and Computers,* 2012.

[27] R. A. Grimes, "TechNet," Microsoft, 2002. [Online]. Available: https://technet.microsoft.com/en-us/library/dd632947.aspx.

[28] J. Mason, "Advanced Encryption Standard (AES)," thebestvpn, 2017. [Online]. Available: https://thebestvpn.com/advanced-encryption-standard-aes/.

[29] F. Alhumaidan, "A Critical Analysis and Treatment of Important UML Diagram Enchancing Modeling Power," *Intelligent Information Management,* 2012.

[30] S. Lee, "Unified Modeling Language (UML) for Database System and Computer Applications," *International Journal of Database Theory and Application,* 2012.

[31] P. Kolte, T. Bhujbale and A. Chaware, "Web Portal Development using Programming Practices," 2012.

[32] W. Pierce, "Disadvantages and Advantages of Extreme Programming," 2016. [Online]. Available: https://atlaz.io/blog/disadvantages-and-advantages-of-extreme-programming/.