SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

& COLLEGE OF COMPUTER SCIENCE INFORMATION TECHNOLOGY

COMPUTER SYSTEMS AND NETWORK DEPARTMENT

A PRIVACY PROTECTION APPLICATION

تطبيق حماية الخصوصية

THE DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT FOR THE REQUIREMENT OF BSC (HONOUR) DEGREE IN COMPUTER SYSTEMS AND NETWORKS

October 2017

بسم الله الرحمــن الرحيــم

SUDAN UNIVERSITY OF SCIENCE & TECHNOLOGY

COLLEGE OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

COMPUTER SYSTEMS AND NETWORK DEPARTMENT

Privacy protection application

Supervisor by:

T. reham lotfy

Students:

Hussam Haider yosif

Mohammed Azez Ali

الأيـــــة

قال تعالى:{ يرفع الله الذين امنوا منكم والذين أوتوا العلم درجات}

المجادلة[11]

صدق الله العظيم

الحمـــد

الحمد لله رب العالمين ،الحمد لله الذي أارد فقدر ،وملك فقهر ،وخلق فأمر وعبد فأثاب ،وشكر ،وعصي فعذب وغفر ،جعل مصير الذين كفرو إلي سقر ،والذين اتقوا ربهم إلي جنات ونهر ،ليجز الذين كفرو بما عملو ،والذين امنوا بالحسنى واشهد إن لا اله إلا الله ،وحده لا شريك له ،له الملك ،وله الحمد ،وهو علي كل شيء قدير يارب رضاك خير إلي من الدنيا وما فيها يا مالك النفس قاصيها ودانيها فنظرة منك يا سؤلي ويا أملى خير إلى من الدنيا وما فيها فليس للنفس أمآل تحققها سوى رضاك فذا أقصى أمانيها وأشهد أن سيدنا وحبيبنا وشفيعنا محمد عبد الله ورسوله عليه أفضل الصلاة واتم التسليم

ACKNOWLEDGEMENT

At the beginning and in the end all thanks belong to ALLAH. We are grateful to our supervisor T.Reham lotfy. Thanks for Sudan University of Science and Technology(SUST), Thanks to every teacher who have taught us,

DEDICATION

Mohmmed

To my dear father,

To my dear mother,

To all my brothers ,

To my family,

To my cousin,

To my self.

Hussam

To my dear father,

To my dear mother,

To my family,

To Sudan University of Science and Technology (SUST),

To all my friends and my colleagues,

To my teachers from basic school till now,

To my best teacher Dr. Reham lotfy .

Abstract

There is no doubt about the massive technology nowadays that affects our everyday life. One of these fronts is the rapid evolution within smart phones and their applications which simplifies our lives and our daily tasks.

By August 2013 there were 900,000 application in Google's Play Store for android devices, which shows the wide spread of smart phone apps. This wide spreading of apps leads to the creation of numerous stores that allow apps downloading, most of these stores don't have the systems to test these apps which results in stores being filled with malicious software.

We propose to solve this problem by developing an application, which is a system that works on the smart phones that use the Android operating system, the application examines the permission of applications installed in the phone and alerts the user and then classifies the applications into three values that show the status of the application (safe - medium - dangerous) Three colors are clear (green - yellow - red).

After the application of the system and testing it became possible for the user to download applications from all available stores and ensure the integrity of its powers after installation, and depending on the result extracted from the system, the user can decide whether to install the application or not install.

المستخلص

مما لا شك فيه و ما لا يخفى على احد منا التطور التكنولجى المتسارع الذى يشهده العالم اليوم فى شتى مناحى الحياة . و من هذه المناحى ، التطور السريع فى الهواتف الكيه و البرمجيات التى تسهل علينا انجاز مهامنا اليوميه .

حتى اغسطس 2013 كان هنالك حوالى 900,000 تطبيق فى متجر جوجل بلاى و هو متجر على الويب للبرامج تديره قوقل لاجهزه الاندرويد ،مما يعنى مدى انتشار تطبيقات الهواتف الذكيه .

هذا الازدياد المستمر فى عدد التطبيقات ادى الى ظهور العديد من المتاجر التى يمكن من خلالها تحميل التطبيقات مما يؤدى الى وجود العديد من التطبيقات الضاره بالمستخدمين .

اقترحنا لحل هذه المشكلة تطوير تطبيق وهو عباره عن نظام يعمل علي أجهزه الهواتف الزكية التي تسخدم نظام التشغيل اندرويد ، يقوم التطبيق بفحص صلاحيات التطبيقات المثبتة في الهاتف وتنبية المستخدم ومن ثم تصنيف التطبيقات الي ثلاثة قيم توضح حالة التطبيق (امنة - متوسطة - خطره) عن طريق ثلاثة الوان واضحة (أخضر - أصفر – احمر).

بعد تطبيق النظام وأختباره أصبحه من الممكن للمستخدم ان يقوم بتحميل التطبيقات من جميع المتاجر المتاحة والتاكد من سلامة صلاحياته بعد تثبيته ، وأعتماد علي النتيجه المستخرجه من النظام امكن للمستخدم باتخاز القرار بتثبيت التطبيق او عدمة.

LIST OF TERMS

Term	Descriptions
SQLite	Structured Query lite
DVM	Dalvik Virtual Machine
JVM	Java Virtual Machine
-APK	Android application package
IDE	integrated development environment
AndroidSAT	Security Analysis Tool for Android Applications
ADT	Android Development Tools
UML	Unified Modeling Language
OMG	Object Management Group
OS	Operating System

Table of Contents

Table of figures

Table of tables

Chapter 1

INTRODUCTION

1.1 Overview

With the development of the software industry, there are markets that promote effective software for users. The user downloads this software to benefit from it and does not know the validity of such applications, and in present time there are some applications appeared that penetrate the private information of users without the knowledge of them.

This research try to make users aware of what is happening at their phones, and protect the information that have been considered important to the user, which cannot be exposed to others.

1.2 Problem Statement

Users Lake of knowledge about application's permissions, the malicious software, and how it disrupts computers or cell phones and exposes user's private information to third party makes the need for software to notify the user when some information are send out of their devices.

1.3 Scope of Research

This Research covers seven of the user perspective aspects of security which are: Gallery, Camera, Phone Contact List, Wi-Fi connection, Data Connection, Bluetooth Connection and we have identified sever permissions and we identified these permissions after studying the impact on the phone and it turned out to be more impact on the phone than the rest of the permission (camera , wifi ,gallery , read content ,write content ,Bluetooth , internet) and on the basis of these permissions determine the degree of seriousness of application , and notifications the user when any application uses or sends private information to external entity using one of these seven permissions, and gives the user the ability to cancel such events. It also scan user device and gives a report about the installed applications and its degree of risk.

1.4 Objective of Research

This project ensures the privacy of the user, and secures the mobile from penetration.

1.5 Proposed Solution

This research builds android application that scans user's device, sends notification, searches for applications that are working to disrupt the cellphone, and gives the user the ability to uninstall it.

1.6 Expected results

Build a successful application that protects user privacy and saves your mobile from damage.

1.7 Thesis Layout

This research has the following Thesis Layout:-

Chapter 2: Explore the theoretical background of the android applications, and related studies.

Chapter 3: Discusses the techniques and tools that will be used to achieve the research objective.

Chapter 4: System Description and analysis.

Chapter 5: discusses the steps we took to create our project, code Implementation

Chapter 6: contains the results, conclusion and recommendations.

Chapter 2

EXPLORE THETHEORETICAL BACKGROUND AND RELATED STUDIES

2.1 Overview

This chapter will touch two major parts: the general description of android application and introduce the area of the research.

2.2Android

Android is an emerging platform with about 19 different versions till date Table 2.1shows different Android versions with their corresponding release date.

The Android framework is built over Linux kernel that controls and governs all the hardware drivers such as audio, camera and display drivers. It contains open source libraries such as SQLite, which is used for database purposes, and SSL library that is essential to use the Secure Sockets Layer protocol. The Android architecture contains Dalvik Virtual Machine (DVM), which works similar to the Java Virtual Machine (JVM). However, DVM executes (.dex) files whereas JVM executes .class files[15].

Every application runs in its own Dalvik virtual environment or sandbox in order to avoid possible interference between applications and every virtual environment running an application is assigned a unique User-ID(UID).

The application layer as consists of the software applications with which users interact. This layer communicates with the application framework to perform different activities. This application framework consists of different managers, which are used by an Android application. For example, if an application needs access to an incoming/outgoing phone call, it needs to access Telephony-Manager. Similarly, if an application needs to pop-up some notifications, it should interact with Notification Manager [15].

Table1.1 android version history[15]

Android Version	OS Name	Release Date
1.0	Alpha	09/2008
1.1	Beta	02/2009
1.5	Cupcake	04/2009
1.6	Donut	09/2009
2.0-2.1	Eclair	10/2009
2.2	Froyo	05/2010
2.3.x	Gingerbread	12/2011
3.1-3.2	Honeycomb	02/2011
4.0.3-4.0.4	Ice Cream Sandwich	10/2011
4.1.x-4.3	Jelly Bean	08/2012
4.4	KitKat	09/2013
5.1.1	Lollipop	12/2015
6.0	Marshmallow	08/2016
7.1.1	Nougat	12/2016

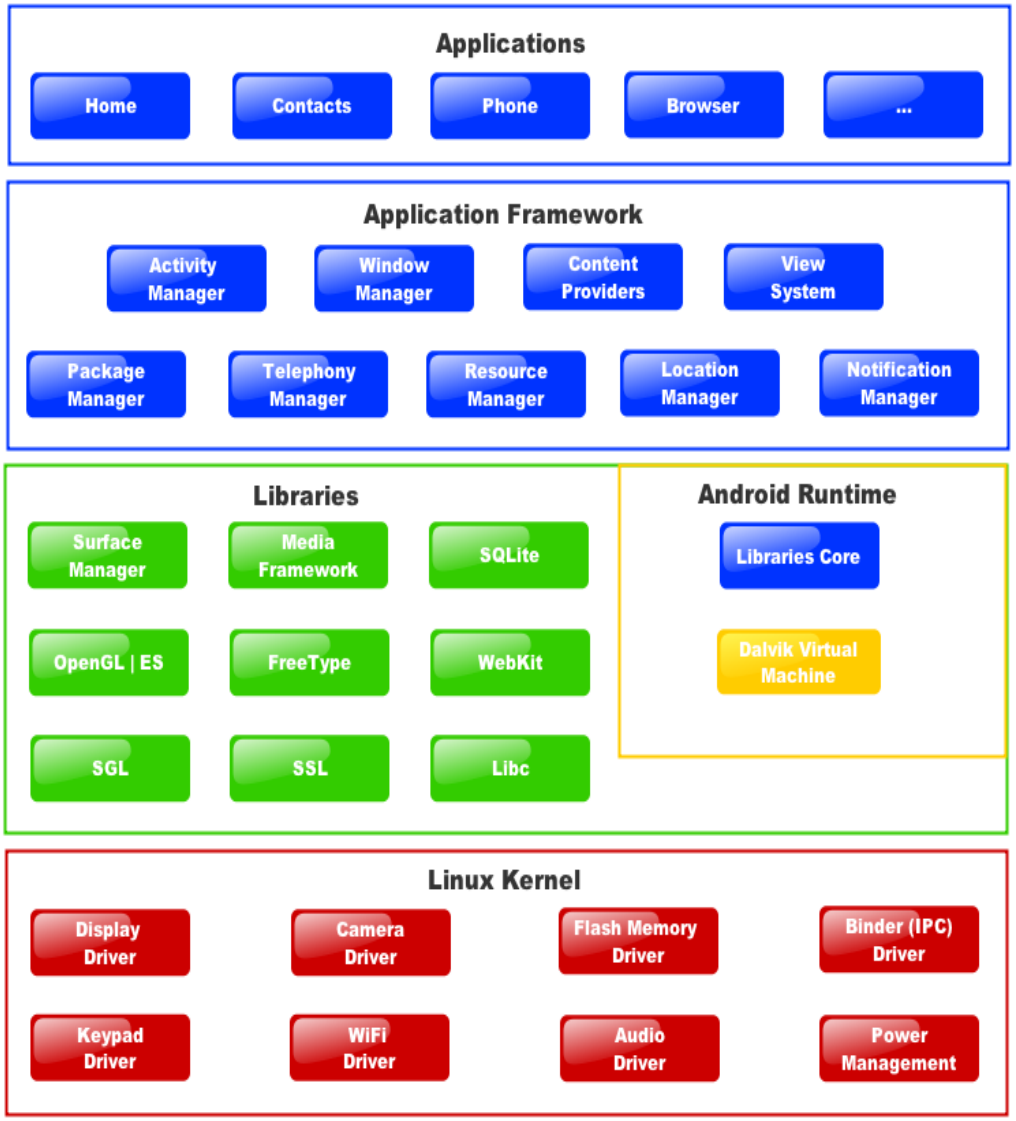

Figure 2-1 android architecture[15]

2.3 Android Security

Android devices come with built-in software called Verify Apps, which regularly checks to make sure all apps on your device are behaving. If a harmful app is detected, Verify Apps will display an alert, or block the app entirely[8].

2.4 Malicious software

Malicious software, commonly known as malware, is any software that brings harm to a computer system. Malware can be in the form of worms, viruses, Trojans, spyware, adware and root kits, which steal protected data, delete documents or add software not approved by a user[5].

2.5 Android Permissions

System permissions are divided into several protection levels. The most important protection levels to know about are normal and dangerous permissions:

Normal permissions cover areas where your app needs to access data or resources outside the app's sandbox, but where there's very little risk to the user's privacy or the operation of other apps.

Dangerous permissions cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps[7].

2.6 Android application package (APK)

Android application package (APK) is the package file format used to distribute and install application software and middleware onto Google's Android operating system. Certain other operating systems and devices, such as BlackBerry devices with the operating system version 10 or higher, also support APK packages.

APK files are analogous to other software packages such Dep packages in Debian-based operating systems like Ubuntu. To make an APK file, a program for Android is first compiled, and then all of its parts are packaged into one file.

An APK file contains all of that program's code (such as .dex files), resources, assets, certificates, and manifest file. As is the case with many file formats, APK files can have any name needed, provided that the file name ends in ".apk". APK files are a type of archive file, specifically in zip format packages based on the JAR file format, with “.apk” as the filename extension. The MIME type associated with APK files is application/vnd..android. package-archive[12] .

2.7 PREVIOUS STUDIES

2.7.1Detection of Malicious Android Mobile Applications Based on Aggregated System Call Events

This study presented techniques to effectively detect the malicious apps which are easy to install and use on its Android based commercial mobile device environment. Above all, it analyzed the access methods and research results on Crowdroid techniques collecting and analyzing the system call events occurring upon executing apps. It suggested techniques of discriminating the malicious apps based on this, implementing the extracting module of the system call events in Android based commercial mobile devices. It performed comparison analysis on characteristics of system call events occurring on normal and malicious apps using Strace module being able to collect the system call events in Android kernel. It also presented the algorithm to discriminate the malicious apps using the algorithm of frequency and similarity analysis of occurring events.

The use of techniques presented in this study made it possible to analyze the characteristics of system call events occurring upon executing malicious apps, and can be applied for a way to discriminate whether the arbitrary mobile apps are malicious or not through this[1].

2.7.2AndroSAT: Security Analysis Tool for Android Applications

The increasing popularity of the Android operating system has led to sudden escalation in Android malware. This work developed a framework to analyze Android applications using static and dynamic analysis techniques (AndroSAT).

Static analysis techniques aim to analyze Android Apps without executing them. The objective of these techniques is to understand an application and predict what kind of operations and functionalities might be performed by it without executing it.

A dynamic analysis executes Android Apps in a controlled virtual environment, which logs low-level interactions with the operating system.

The effectiveness of AndroSAT was tested by analyzing a data set of1932 applications. The information obtained from the produced analysis reports proved to be very useful in many Android security related applications.

In particular, it used the data in these reports to perform three case studies: analyzing the frequency of use of different Android permissions and dynamic operations for both malicious and benign Apps, producing gcyber-intelligence information, and malware detection. The implemented prototype can be further extended to allow for more useful add-ons that can be used to provide further investigation of the security of Android applications .

The increasing popularity of the Android operating system has led to sudden escalation in Android malware. In this work, we developed a framework to analyze Android applications using static and dynamic analysis techniques (AndroSAT) [15].

Chapter 3

TOOLS AND TECHNIQUES

3.1Overview

This chapter discusses the techniques and tools that will be used to achieve the research objective.

3.2 Android Studio

Android Studio is the official integrated development environment (IDE) for the Android platform it was announced on May 16, 2013 at the Google I/O conference[14].

Based on JetBrains' IntelliJ IDEA software, Android Studio is designed specifically for Android development. It is available for download on Windows, mac OS and Linux, and replaced Eclipse Android Development Tools (ADT) as Google's primary IDE for native Android application development[13].

Some of the important features:

- Gradle-based build support.
- Android-specific refactoring and quick fixes.
- Lint tools to catch performance, usability, version compatibility and other problems.
- ProGuard integration and app-signing capabilities.
- Support for building Android Wear apps.
- Android Virtual Device (Emulator) to run and debug apps[14].

3.2.1Advantages

- Faster code/compile/run cycles (real time).
- Extensible (plug-in).

3.2.2 Disadvantages

- Pretty heavyweight.
- Requires JRE.

- Difficult to Learn [14].

3.3 Java

Java is a widely used programming language expressly designed for use in the distributed environment of the internet. It is the most popular programming language for Android smart phone applications and is among the most favored for edge device and internet.

Some of the important features:

- Simple.
- Object-Oriented.
- Portable.
- Platform independent.
- Secured.
- Architecture neutral.
- Dynamic.
- Interpreted.
- High Performance.
- Multithreaded.
- Distributed.[11]

3.4 Extensible Markup Language (XML)

Extensible Markup Language (XML) is used to describe data. The XML standard is a flexible way to create information formats and electronically share structured data via the public Internet, as well as via corporate networks.

Some of the important features:

- XML files are text files.
- XML is very simple.
- XML is extensible[2].

3.5 UML Diagrams

Most people refer to the Unified Modeling Language as UML. The UML is an international industry standard graphical notation for describing software analysis and designs. When a standardized notation is used, there is little room for misinterpretation and ambiguity. Therefore, standardization provides for efficient communication (a.k.a. “a picture is worth a thousand words”) and leads to fewer errors caused by misunderstanding.

The U in UML stands for unified because the UML is a unification and standardization of earlier modeling notations of Booch, Rumbaugh, Jacobson, Mellor, Shlaer, Coad, and Wirf-Brock, among others.

The UML most closely reflects the combined work of Rumbaugh, Jacobson, and Booch – sometimes called the three amigos. The UML has been accepted as a standard by the Object Management Group (OMG). The OMG is a non-profit organization with about 700 members that sets standards for distributed object oriented computing [2].

3.5.1Use Case Diagram

A use case is a list of actions or event steps, typically defining the interactions between a role and a system, to achieve a goal. The actor can be a human or other external system.

In systems engineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals[2].

3.5.2 Sequence diagram

A sequence diagram, in the context of UML, represents object collaboration and is used to define event sequences between objects for a certain outcome.

A sequence diagram is an essential component used in processes related to analysis, design and documentation[2].

3.5.3Deployment diagram

Deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes. In other words, deployment diagrams show the hardware for your system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another[2].

Chapter 4

SYSTEM ANALYSIS

4.1Introduction

This section illustrates the general description of the system and its functions. It's also clarified the software components, hardware, and the detailed analysis of the system operations using the schemes UML.

4.2System Description

This system detects malicious applications installed in the phone and also identifies applications that penetrate the privacy of the user by giving notice of penetration.

4.3 System Environment

The integrated development environment (Android Studio) was used to develop the application and a database was created to store the latest application status.

4.4System functions

This application does the following functions:

- The application scans all installed applications and then determines the permissions of each application
- Determines the degree of risk or validity for each application.
- Removes harmful applications and give the ability to disable some features of the application.

4.5Analysisusing UML Scheme:

4.5.1 Use Case Diagram

Figure (4.1): illustrates the use case diagram of the android application and processes that can be made

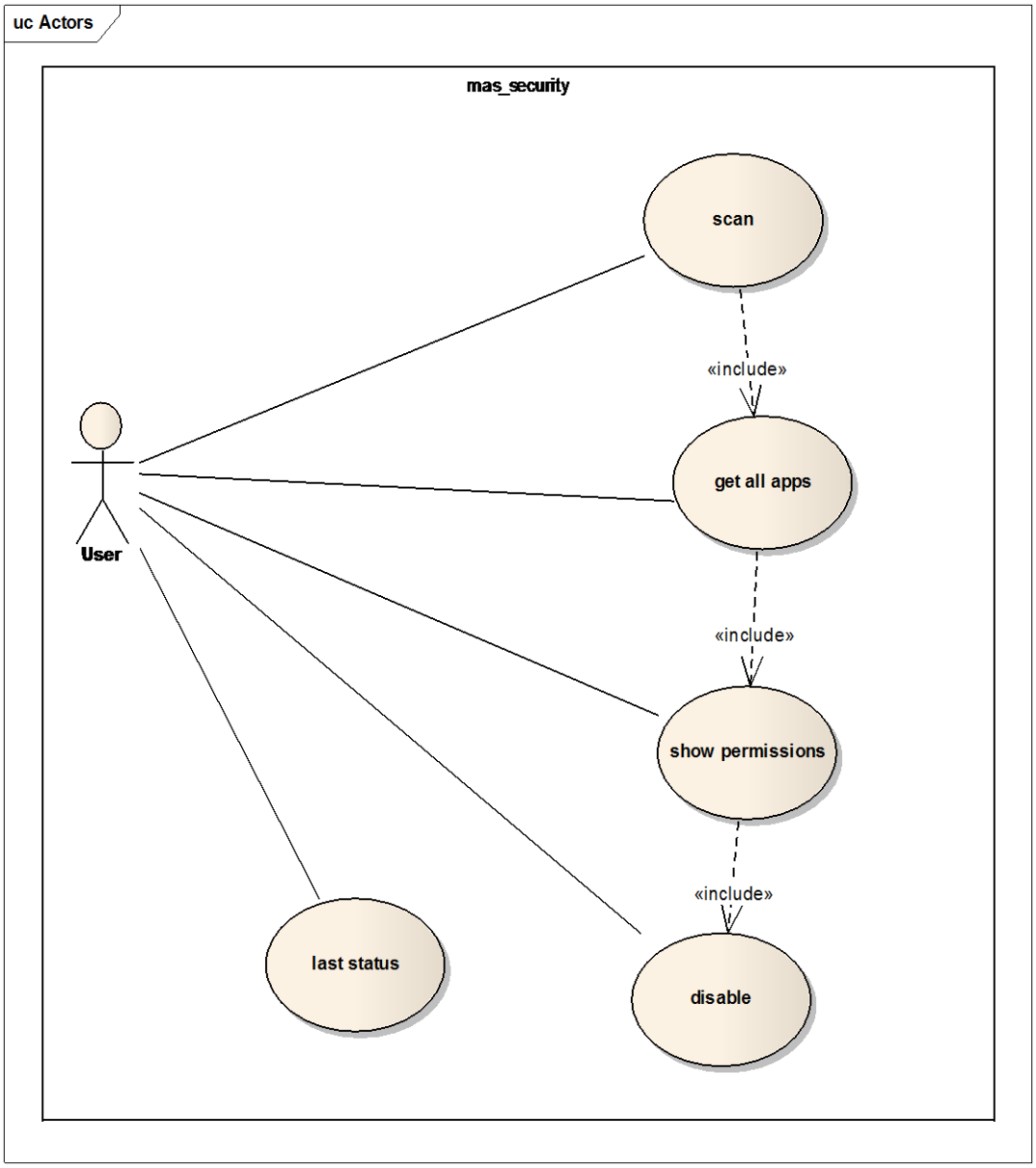

Figure 4-1use case diagram

4.5.2 Sequence Diagram

Figure (4.2): illustrates the sequence diagram that shows the system sequence of events .The user installs the application on the mobile phone to take advantage of it.

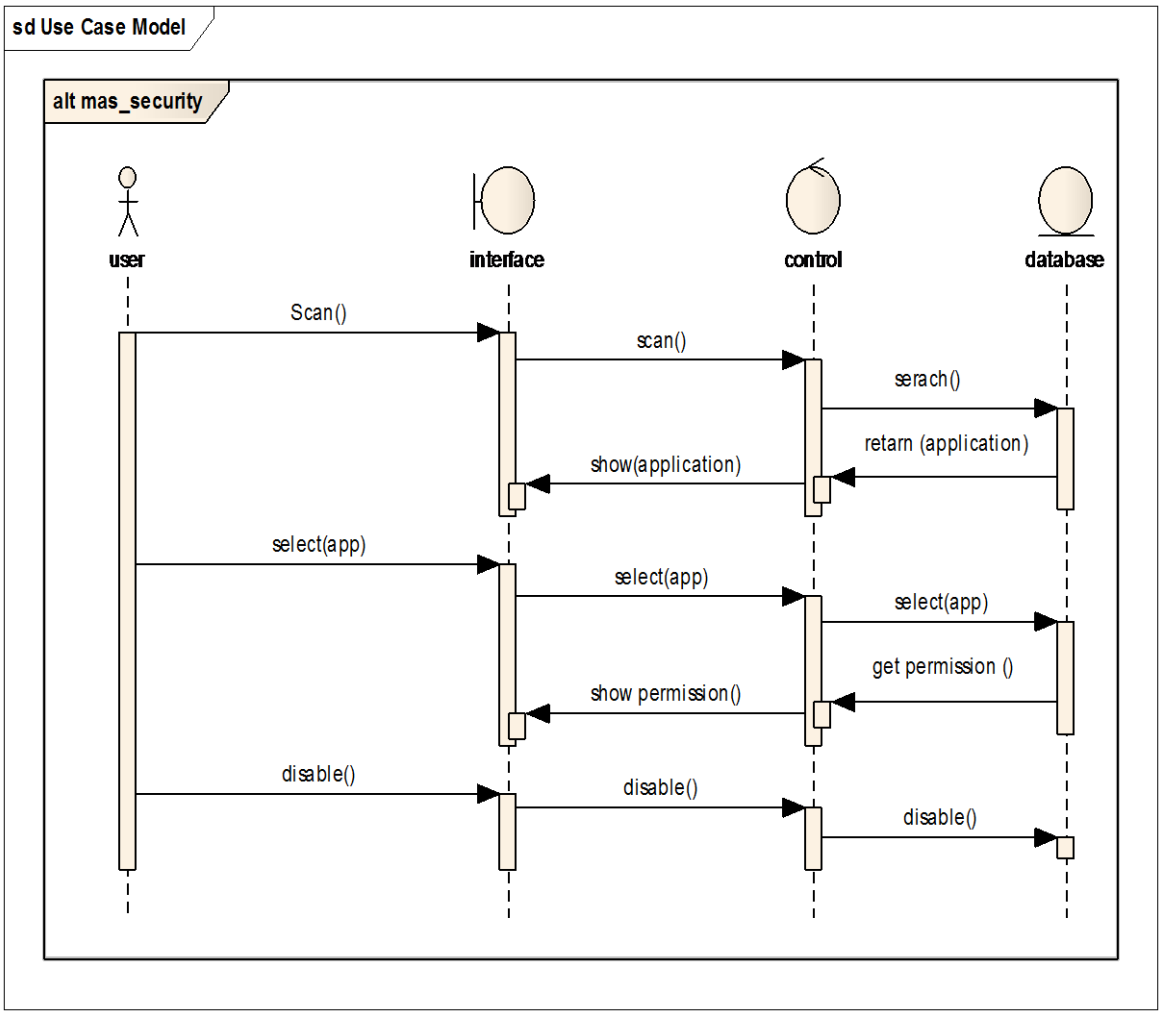

Figure 4-2 sequence diagram

Table 2: explain the function in sequence

Use Case Name	Monitor
Actor	User
Preconditions	No
Main Flow of Events	1. The user click button to performs a search for all applications status 2. A screen with all the applications in the phone displays. 3. If the user clicks on any of the applications in the list a screen that explaining the seriousness of the application is appeared. 4. An uninstall icon is also there if the user senses that the application is so dangerous.
Post Conditions	No

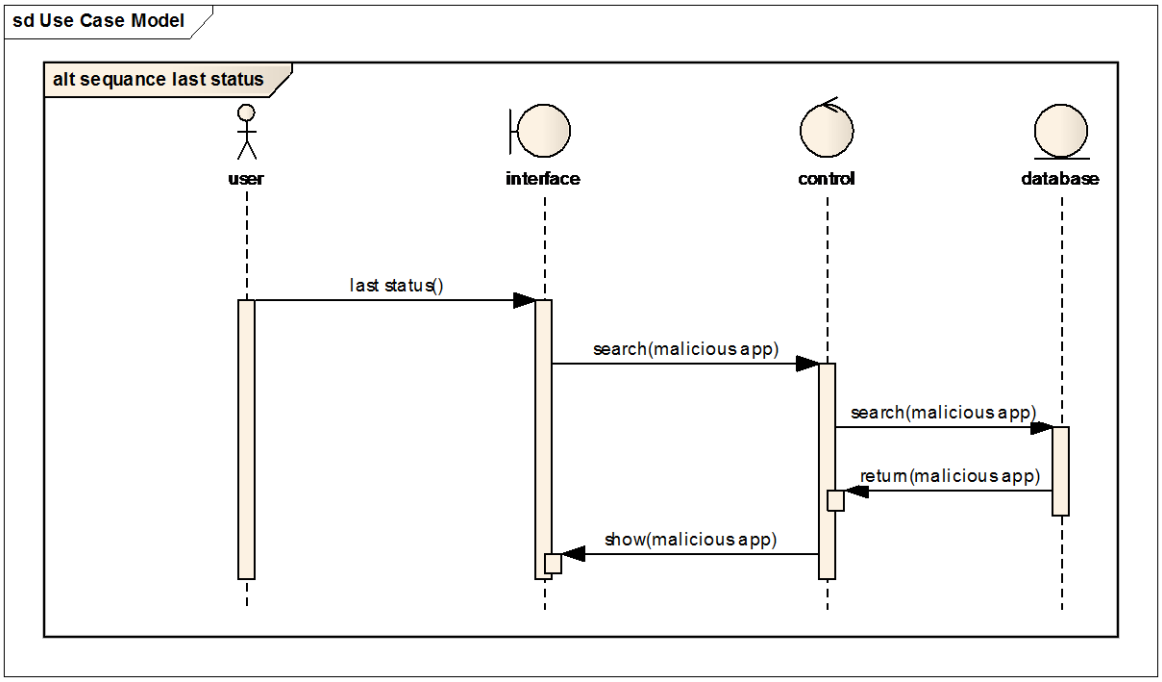

Figure 4-3 list status

Second: Determine the last state of the phone. The most dangerous application and how much the device is protected.

Table 3: explain the status

Use Case Name	Monitor
Actor	User
Preconditions	No
Main Flow of Events	The information is saved after the Scan and returned as to the percentage of compliance and the most dangerous application.
Post Conditions	No

4.5.3 Deployment Diagram

Figure (4.3): illustrates deployment diagram illustrates hardware and software used in the system and how these components interact with each other.

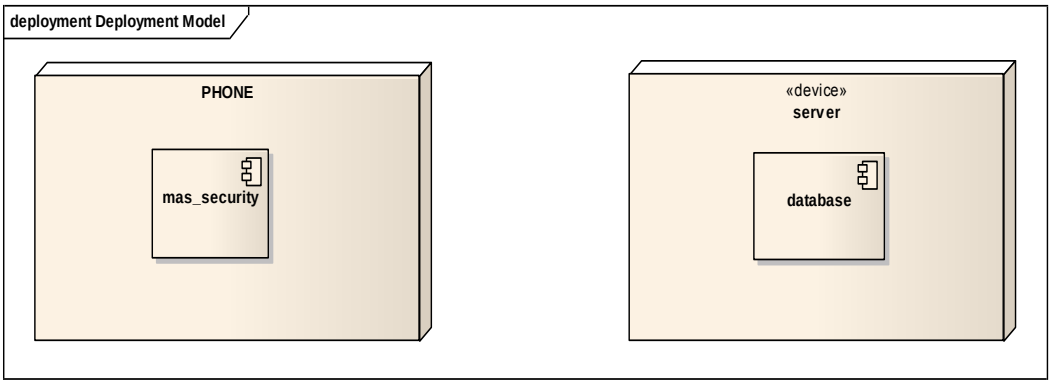

Figure 4-4deployment diagram

Specifications:

Table 4: Specifications

No.	Name	description
1	Ram	512
2	Hard disk	1G
3	Os	android
4	Android version	7
5	Database version	Sql lite

Chapter 5

IMPLIMINTATION SYSTEM

5.1 Introduction

In this chapter we will talk about how the user will interact with the software, and the Graphical interfaces of it.

5.2 Graphic User Interface

Figure 5.1 shows the first or splash screen that appears when the user clicks on the application icon.

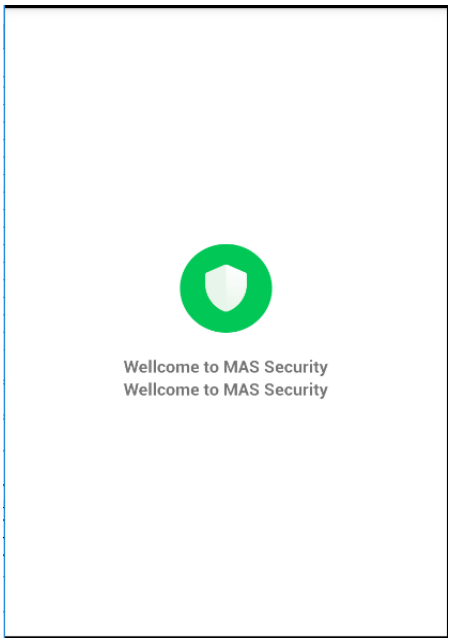

Figure 5-1Splash

5.3 Home screen

Figure 5.2 shows the Main Screen. In this screen, the user can scan the applications installed on the cell phone and know the status of the device if there is

a dangerous application, and it also gives the user information the profile of the application.

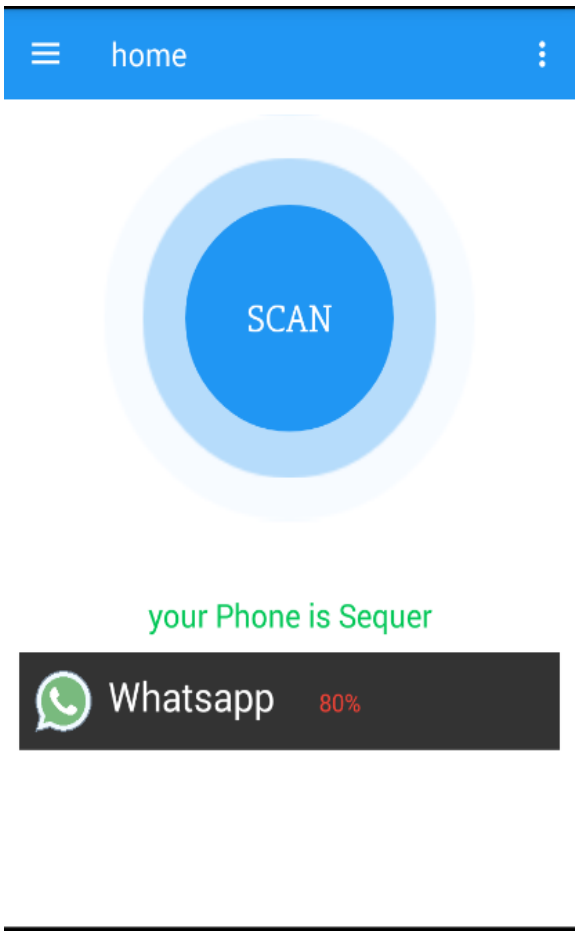

Figure 5-2 Main Screen

5.4 Navigation drawer

Figure 5.3 shows the navigation drawer. In this screen the user have information about the cell phone, how to use the application, and the safe or trusted applications at it phone.

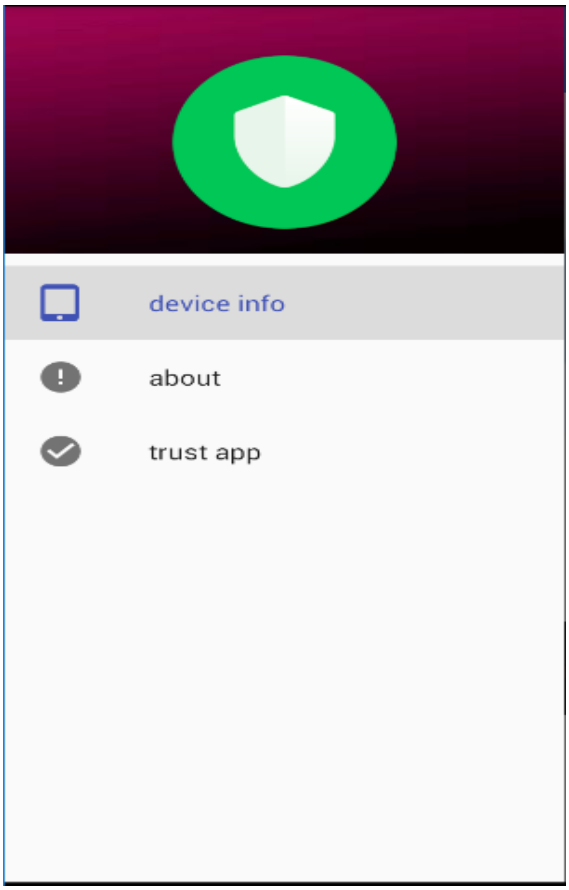

Figure 5-3 navigation drawer

5.5 Device info

Figure 5.4 shows the Device information screen. This screen displays information about user cell phone such as the version of the operating system, brand, id, and the host name.

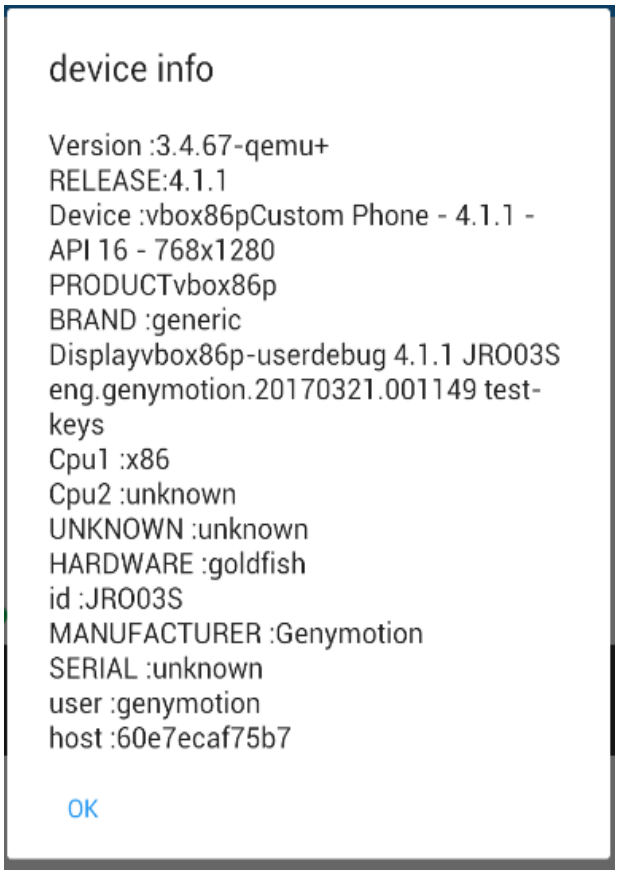

Figure 5-4 Device info

5.6 About

Figure 5.5 shows about the application screen. This screen displays information about the application such as application name, description of the application functions, sponsor, operating system, and RAM details.

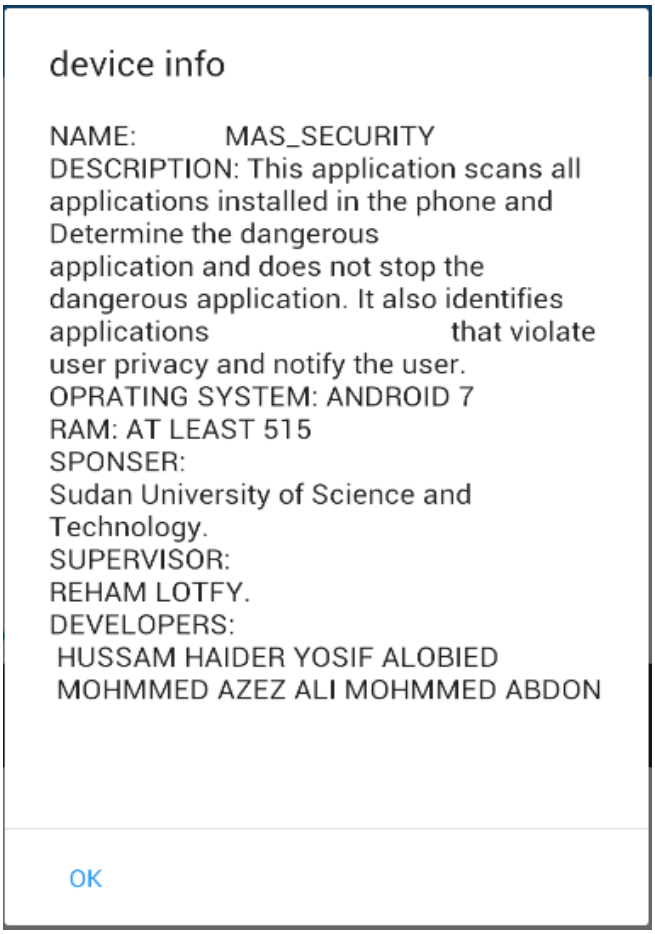

Figure 5-5 About

5.7 Trust app

Figure 5.6 shows list of cell phone trusted apps.

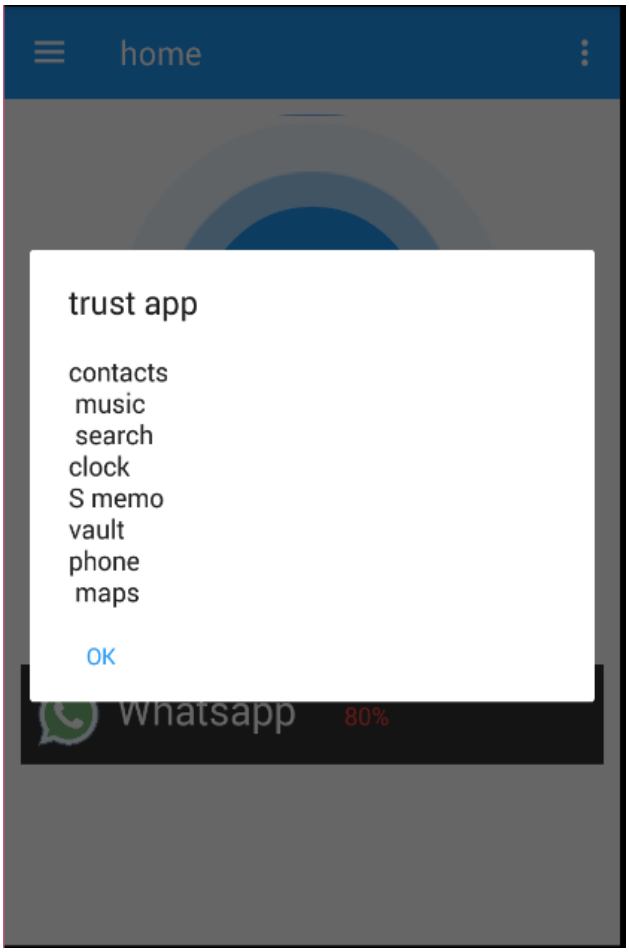

Figure 5-6 Trust app

5.8 List app

Figure 5.7 shows list of all scanned application. This screen displays if the user selects scan from the screen5.2.

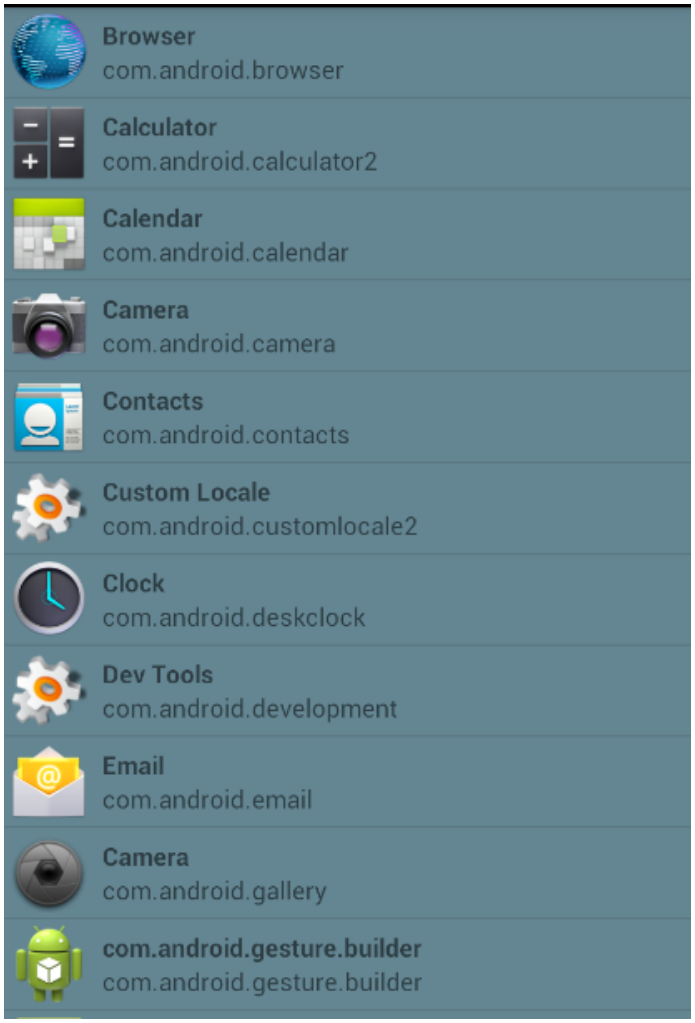

Figure 5-7 list app

5.9 Secure Screen

If the user clicks on any scanned application on figure 5.7 a detailed screen will appear as figure 5.8 shows. The screen contains the permissions and authorities this application has on the cell phone colored according to the risk level

in green to state low risk level, yellow to state medium risk level, red to state high risk level.

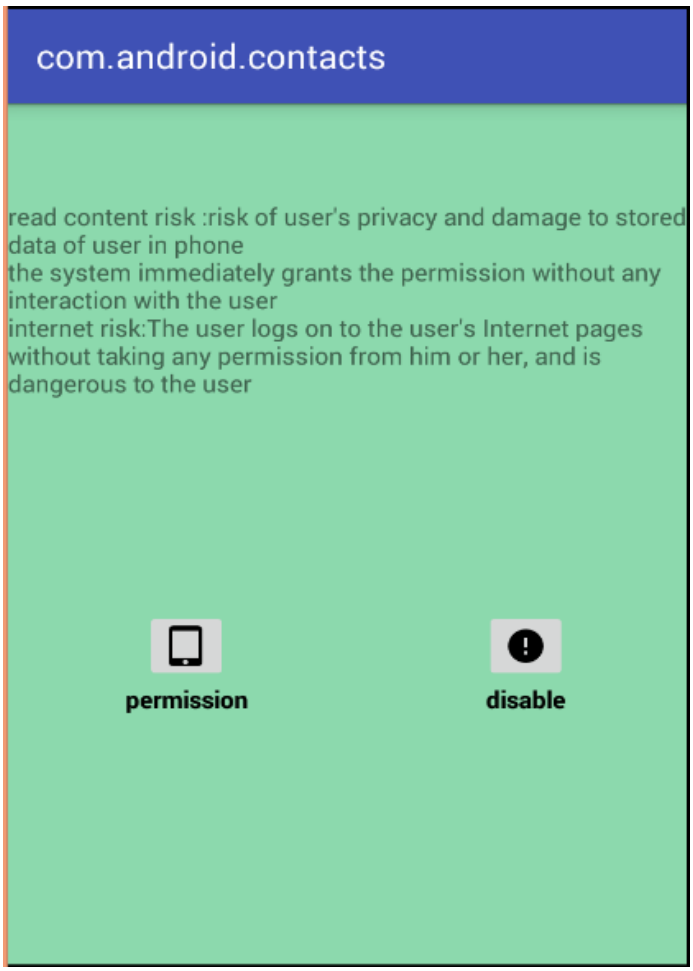

Figure 5-8 Secure Screen

5.10 Medium risk screen

Figure 5.9 shows an application with a medium risk level (uses five permissions).

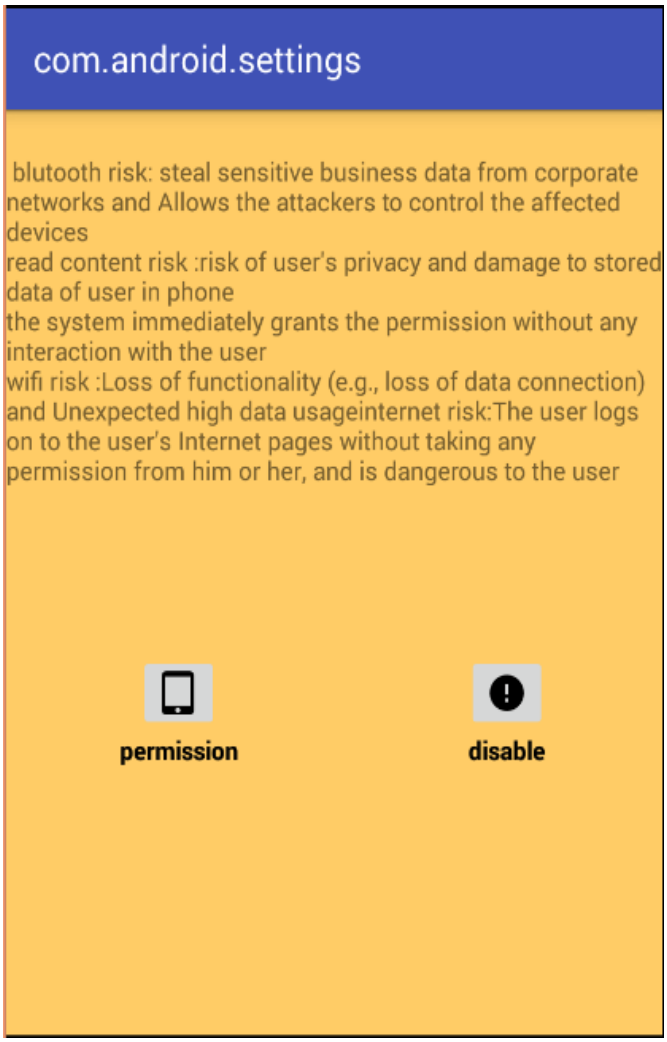

Figure 5-9 Medium risk screen

5.11 High risk screen

Figure 5.10 shows an application with a high risk level (uses all the seven permissions).

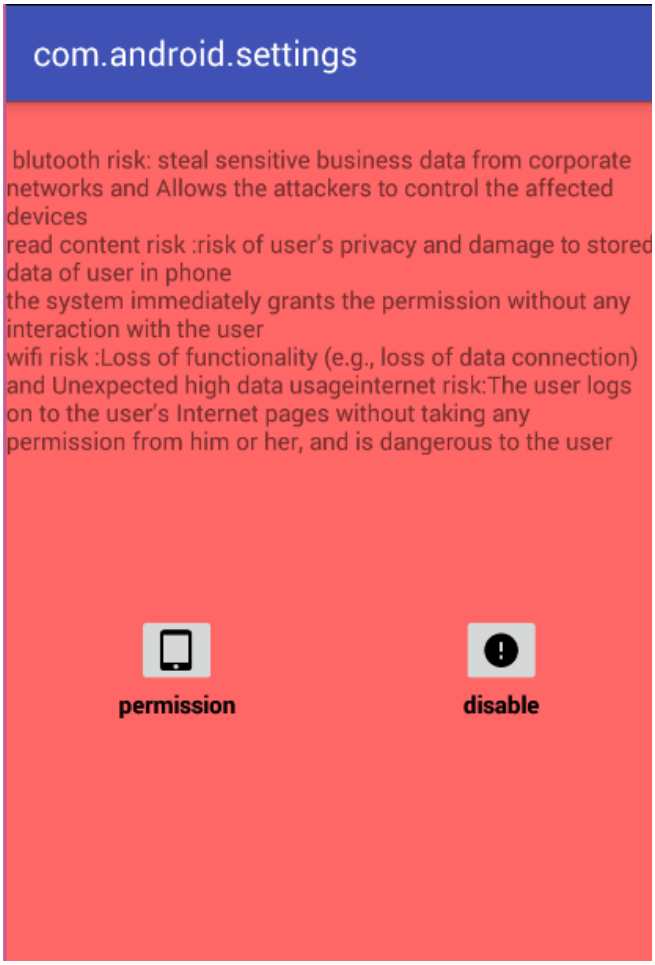

Figure 5-10High risk screen

5.12 Application uninstall screen

Figure 5.11 appears when the user click on disable button that illustrates on figure 5.10.

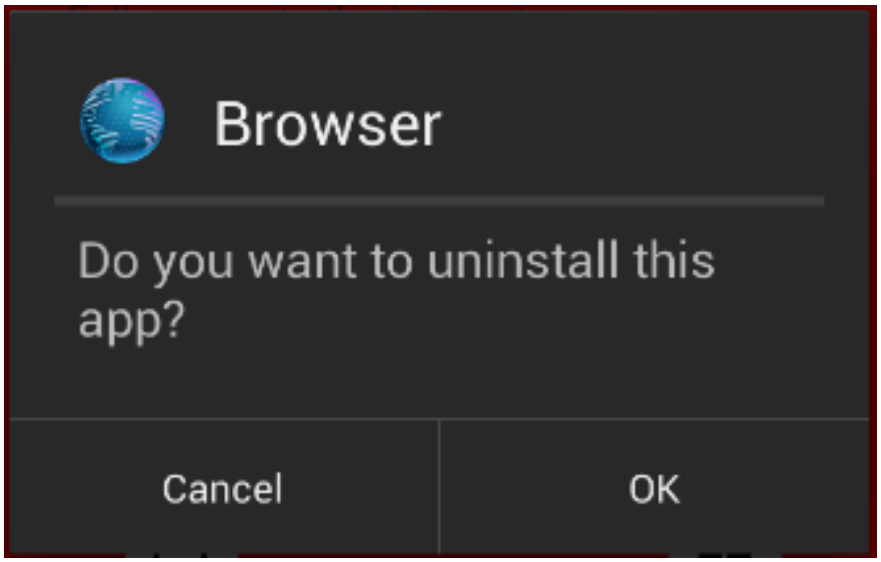

Figure 5-11 Application uninstall screen

5.13Notification

Figure 5.12 shows the notification that appears to user when the cell phone at risk.

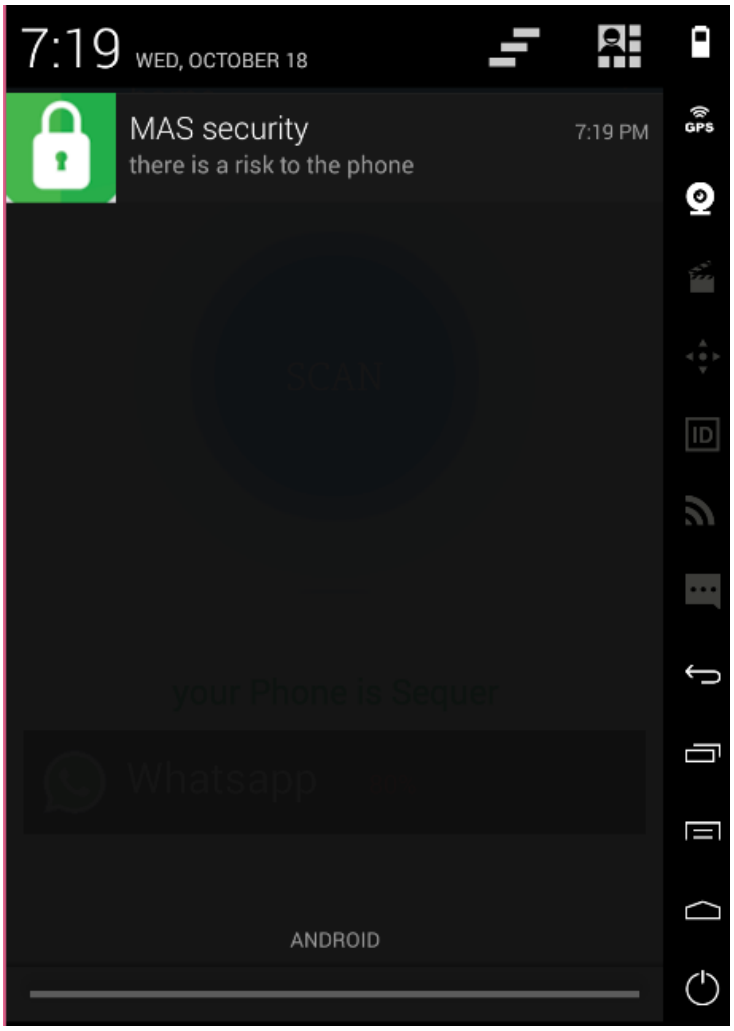

Figure 5-13Notification

Chapter 6

RESULT AND RECOMMUNDATION

6.1 Introduction

This section contains the research results and the recommendations for future work.

6.2 Conclusion

An android application has been established and implemented that provides help to normal users to protected their private data and know the risk levels of installed apps on their cell phones.

6.3 Results

The android application achieved the goals of the research successfully by performing the following functions:

- Scan all installed applications in the cellphone and identify the most dangerous application based on the seven permissions (Bluetooth , Wi-Fi, Camera, Read Contact List, Gallery, Internet access, and SMS).
- Sends alerts to the user when critical risk happen.
- Allows the user to uninstall the application that considered dangerous.
- Lock the camera from all applications unless the user enables it.

6.4 Recommendation

- Determine the risk level of the application during the installation process and alert the user.
- Allow automatic deletion of cash.
- Lock all applications with a particular pattern created by the user.

REFERENCES:

[1] Ham, You Joung, and Hyung-Woo Lee. "Detection of malicious android mobile applications based on aggregated system call events." International Journal of Computer and Communication Engineering 3.2 (2014): 149.

[2]OMG, OMG. "Unified Modeling Language: Superstructure 2.1.2."Object Management Group (November 2007) (2007).

time 5:00 pm

data 11/9

[3]https://www.csee.umbc.edu/courses/undergraduate/CMSC341/Lectures/Ecli pse/intro-to-eclipse.pdf access at 17/8/2016 02:00 p

[4]Google play android market. [Online]. Available:

https://play.google.com/store/apps

[5] More than 700,000 malicious Android apps wreak havoc on the web.

[Online]. Available:

http://www.neowin.net/news/more-than-700000-malicious-apps-wrea

k-havoc-in-the-play-store

[6] Android (operating system).[Online]. Available:

http://en.wikipedia.org/wiki/Android_(operating_system)

[7] Uncovering android master key that makes 99% of devices vulnerable.

[Online]. Available:

http://bluebox.com/corporate-blog/bluebox-uncovers-android-masterkey/

[8] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, “A study of

android application security,” in Proc. the 20th USENIX Security

Symposium, 2011.

[9] Y. J. Zhou and X. X. Jiang, “Dissecting Android malware:

characterization and evolution,” in Proc. the 33rd IEEE Symposium on

Security and Privacy, 2012, pp. 95-109.

[10] I. Burquera, U. Zurutuza, and S. Nadjm-Tehrani, “Crowdroid: behavior-based malware detection system for Android,” in Proc. the 1st ACM workshop on Security and privacy in smartphones and mobile devices, 2011, pp.15-26.

[11] S. Brahler, “Analysis of the android architecture,” Karlsruhe institute for technology, 2010.

[12] National Security Agency, “SELinux,” January 2009. https://www.nsa.gov/research/selinux/.

[13]The Android mobile platform by Benjamin SpeckmannA Review Paper Submitted to the Eastern Michigan University Department of Computer Science.

[14]Developers, Android. "What is android." (2011).

[15] Oberoi, Saurabh. Androsat: Security analysis tool for android applications. Diss. Concordia University, 2014.