



FEBRUARY 1, 2016

BUILD A DYNAMIC MALWARE ANALYSIS SYSTEM FOR ANALYZING ANDROID
APPS TARGETING ON ANDROID'S NEW RUNTIME (ART)
Project proposal

R R HWAWASAM
SRI LANKA INSTITUTE OF INFORMATION TECHNOLOGY FACULTY OF POSTGRADUATE AND RESEARCH
M.SC. IN IT / IS / IM RESEARCH PROJECT



BUILD A DYNAMIC MALWARE ANALYSIS SYSTEM FOR ANALYZING ANDROID APPS
TARGETING ON ANDROID'S NEW RUNTIME (ART)

Project Proposal

Supervised by:

.....

Mr.Lakmal Rupasinghe

Abstract

Android users are free to download and install 3rd party applications. But just as the development of the PC brings computer viruses and results in a serious threat to Internet security. The rapid development of smart terminal also brings a variety of malicious software, mobile Internet is also facing a growing number of security issues. Android has more than 60% of the market share and it is in the initial stage.[\[1\]](#) The mobile phone security industry analysis reports in year 2012, that they have found 26850 of malware infections on Android.[\[1\]](#) Stealing confidential information (i.e IMEI numbers, mobile numbers or Contact details) and send SMS messages without user permission are the most damaging attacks done by the malware in the past. Now it is not like that, the smart terminals perform the same functionality like computers. Data transmission, online shopping using credit cards are few examples, today we perform using smart terminals. Therefore the cost of the vulnerability is very high than the past couple of years. Android device shipments are expected to top 1 billion this year and there are currently more than 1.100.000 apps available in the Play Store, which generated 50 billion downloads in 2013 alone, therefore, it is reasonable to assume that there is plenty of potential malware* and other security threats designed to take advantage of careless Android users.[\[6\]](#) Considering cost, there is a huge requirement to protect Android system from the malware infection. In this product I compatible an existing open source malware analysis tool to new Android ART (Android Runtime). DroidBox is very easy to use tool developed to offer dynamic analysis of Android applications. It is well known malware analysis sandbox and it will dynamically analyze the Android apps.

Contents

Abstract.....	2
Introduction: Context and Background	4
Problem definition	6
Aim and objectives.....	6
Proposed solution	6
Literature review	6
How Android Supports Security and Current Study	7
Research Methodology	19
Procedures	21
Project Deliverables in each sprint:	21
Budget Plan.....	22
Reference:.....	24

Introduction: Context and Background

Smart phones becoming more popular than the personal computers and laptops. In this device we are using open source operating system which has remarkable growth in a short period of time. This is mainly its openness and it is a free open source software. Android users are free to download and install 3rd party applications and also it freely promotes lots of developments on top of the Android operating systems. Because the character of open source Android is supported by a lot of companies, so it is developing very fast, by now it is the dominative smartphone operating system.

Although Android inherited from Linux kernel, and it introduces many other's security mechanism, such as application signature, use virtual machine to run application, authorization management, and its security is higher. But not enough, it has vulnerabilities. Malware is coming with some interesting applications and users are compelled to install them with taking a risks. After it is installed, it will steal user's personal information and leak it to remote destination for evil purpose, and some malware will start paid service while the user are not known. This is a very raw domain and people are not aware of the security threats this field poses.

In such situation identifying malware is a challenge because thousands of apps are developed daily and there are millions of apps in the app stores. Antivirus companies are failed to identify threats in new viruses because of this. And currently, detection on mobile malware is based on traditional computer virus detection method based on signature or behaviors. We can say that:

- static analysis
- dynamic analysis

Most of the intrusion detection tools are from either static or dynamic analysis techniques are used.

Static Analysis: it is a reverse technology, extract Android package and decompile DEX file to get smali file to analysis the malware and the potential threats by detecting the sensitive API in the source code, and determines whether there is a sensitive data leakage. And it is analyzing the malware without executing the program.

Dynamic analysis: use smali files generated in the static analysis to add the corresponding log generated code after sensitive API function, and then repackage application and signature, which will install repackaged application to the modified simulator and the API will analyze all generated sensitive log.

However, it is not every time easy to do static analysis because the source code has been obfuscated after complication, packaging, and signing. The obfuscation tool detects and deletes useless class, field, methods and attributes, and deletes useless annotation to make the byte code optimized. Besides, it makes use of meaningless name to rename class, fields, and methods, thus, the source code has poor readability and greatly increases the difficulty of code analysis. So we think that it is more efficient and effective if we use both these techniques to identify intrusions in the android systems. In this paper I will dynamic analysis open source tool to determine malware infection into the Android OS. The project idea is taken from HoneyNet Project.

The HoneyNet Project is a leading international, non-profit security research organization, dedicated to investigating the latest attacks and developing open source security tools to improve Internet security. There are project ideas for new researches and I have taken this idea from the HoneyNet project. (GSoC 2015 Project Ideas) [\[18\]](#)

DroidBox is very easy to use tool developed to offer dynamic analysis of Android applications. It is well known malware analysis sandbox and it will dynamically analyze the Android apps. DroidBox, authored by Patrick Lantz, is a sandbox for the Android platform. "It focuses on detecting information leaks that were derived from performing taint analysis for information-flow tracking on Android trojan applications. DroidBox is capable to identify information leaks of contacts, SMS data, IMEI, GPS coordinates, installed apps, phone numbers, and network traffic and file operations." DroidBox consists of an own system image and kernel meant to log one applications activities. Using adb logcat DroidBox will look for certain debug messages and collect anything related to the monitored app.[\[17\]](#) Nevertheless you can use DroidBox to get an overview of malicious activities triggered by the app. unfortunately this tool is built in Dalvik Runtime environment and currently it become obsolete since there is a new Android ART is in the market.

DroidBox will dynamically analyze the Android apps and providing following information.

1. Hashes for the analyzed package
2. Incoming/outgoing network data
3. File read and write operations
4. Started services and loaded classes through DexClassLoader
5. Information leaks via the network, file and SMS
6. Circumvented permissions
7. Cryptographic operations performed using Android API
8. Listing broadcast receivers
9. Sent SMS and phone calls

With Android 4.4, a new Android runtime, **ART**. This runtime offers a number of new features that improve performance and smoothness of the Android platform and apps. Currently, ART is available on a number of Android 4.4 devices, such as the Nexus 4, Nexus 5, Nexus 7, and Google Play edition devices. At this time, all devices still use Dalvik as the default runtime. Android runtime (ART) is the managed runtime used by applications and some system services on Android. ART and its predecessor Dalvik were originally created specifically for the Android project. ART as the runtime executes the Dalvik Executable format and Dex bytecode specification. ART and Dalvik are compatible runtimes running Dex bytecode, so apps developed for Dalvik should work when running with ART. However, some techniques that work on Dalvik do not work on ART.

In this product I compatible an existing open source malware analysis tool to new Android ART (Android Runtime). Then I will analyze some most popular Android Operating systems and reviewed their inbuilt security features as well.

Problem definition

From Android 4.4, Android system has a new runtime called ART [\[15\]\[16\]](#) together with Dalvik. Most apps should just work when running with ART. However, some techniques that work on Dalvik do not work on ART. Since Android 5.0, Google totally abandoned Dalvik, so ART becomes the only runtime. Current dynamic analysis systems such as DroidBox, TaintDroid, DroidScope, etc., they are built on Dalvik VM, porting them to ART seems impossible since they depend on DVM heavily[\[15\]](#).

Aim and objectives

DroidBox is developed to offer dynamic analysis of Android applications.

Already developed open source project has to work on new Android ART (Android Runtime).

The goal of this project is to build a dynamic malware analysis system on ART, which allows users to monitor the execution of potentially malicious apps. This includes the following sub-goals:

- Monitoring function calls
- Modifying parameters/return value before/after function's execution
- Dumping objects' contents
- Reporting layer that is compatible with existing systems

Proposed solution

The solution of this project should guarantee two points: low performance overhead and easily maintainability of analysis environment for future new Android versions.

The open source DroidBox is enhanced to work on new Android versions, it requires skills of JAVA, C++, Linux, Android Systems, LLVM and ART.

Literature review

Android which is based on Linux and is open source, which is mainly used for smart mobile devices, such as smartphones and pads. Android is mainly applied in the field of personal smart mobile devices, and this kind of devices always stored personal information. If operating system is vulnerable then the personal information can be stolen by malware applications. In many researches there are mainly two kinds of Android malware detection technique used:

1. **signature-based (static)**
2. **behavior-based(dynamic) detection**

A research has being done which is called N-gram[5Ref5] and it is a static analysis, based malware prevention techniques for Android based mobile phones. N-gram is a software program that uses machine learning based algorithm to detect the that given application is malware or not. In the paper [7], authors have investigated a method and designed a program that uses machine learning algorithm to detect the given mobile application is having malware or not.

Another research [8] based on static analysis, they have proposed a method to detect malware based on the permissions and packages of the applications. Rather than only considering the permission information more information (i.e.,the package information in DEX files) is also measured since some studies have shown that package information contains useful information of Android applications.

Further in some researches which use extended mechanisms rather than above two direct techniques. Some of them are as follows.

Bose[3] proposed a behavior detection system on **Symbian**, training classification model using SVM; Schmidt et al. [4] made *classification detection on Symbian by statically extracting call functions and suggested the centroid Machine Learning algorithm*.

How Android Supports Security and Current Study

Android is a mobile operating system (OS) developed by Google, and it is used by several smartphones. Android phones typically come with several built-in applications and also support third-party programs. Developers can create programs for Android using the free Android SDK (Software Developer Kit). Android programs are written in Java and run through Google's "Dalvik" virtual machine, which is optimized for mobile devices. Users can download Android "apps" from the online Android Market.

Android Software Stack

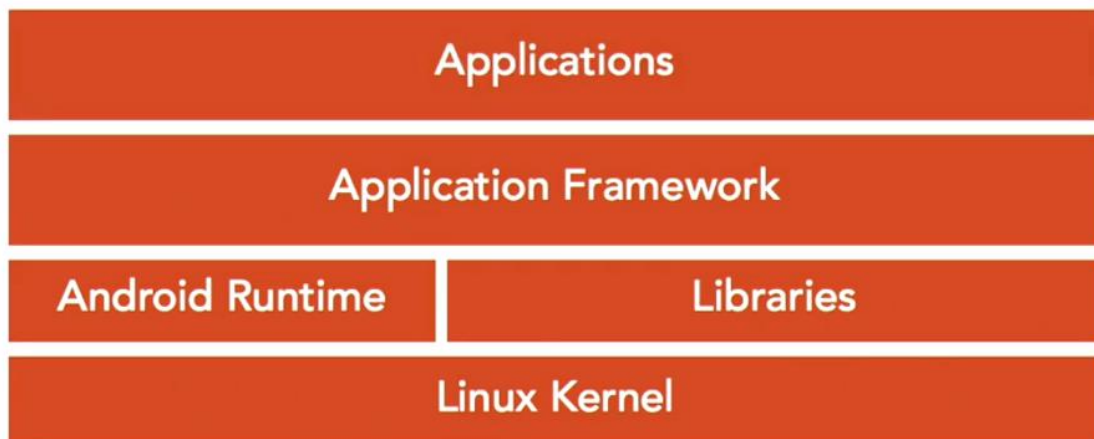


Figure 0:

Android is based on linux kernel, and it is highly optimized for mobile operating systems, and it made small as possible. On top of the kernel there is an Android runtime etc. It up to the Original equipment manufacturer (OEM) to customize these packages according to the their device. So Android is combination of google Android OS and customized drivers from OEM. These OEM like LG, Motorola, Nexus etc. Android Runtime include Core Libraries, and there are two separation for Android 4.4 and Android 5. That is Dalvik is in older versions (4.4 <) and ART is in Android 5 upwards. ART is introduced in KIT-KAT and completely replaced in Lollipop. ART uses a head of time compilation, that easy and faster to run apps, because they are compiled to machine code upon installation rather than application is initiated until feature is being called. To compile an app, you need to use compiler which include in Android SDK. The next level is application framework, it has module to controlling your application.

Application Components

These are like building blocks of an Android application. Each component is a different and basically there are four types of application components.

Activities - An application has several activities, a user can see an activity from a user interface, if there are many activities, and then one of them should be marked as default activity.

- Services - A service is running in the background which used by the application.
- Content Providers - It is data supply component.

- Broadcast receivers - Which are responding to other application when communication is happened.
- Android application package file (APK)

This is a package file which is used in distributing and installing application software and middleware on top of the Android OS.

i.e *.dex files

The file extension should be end as .apk

The Compilation Process

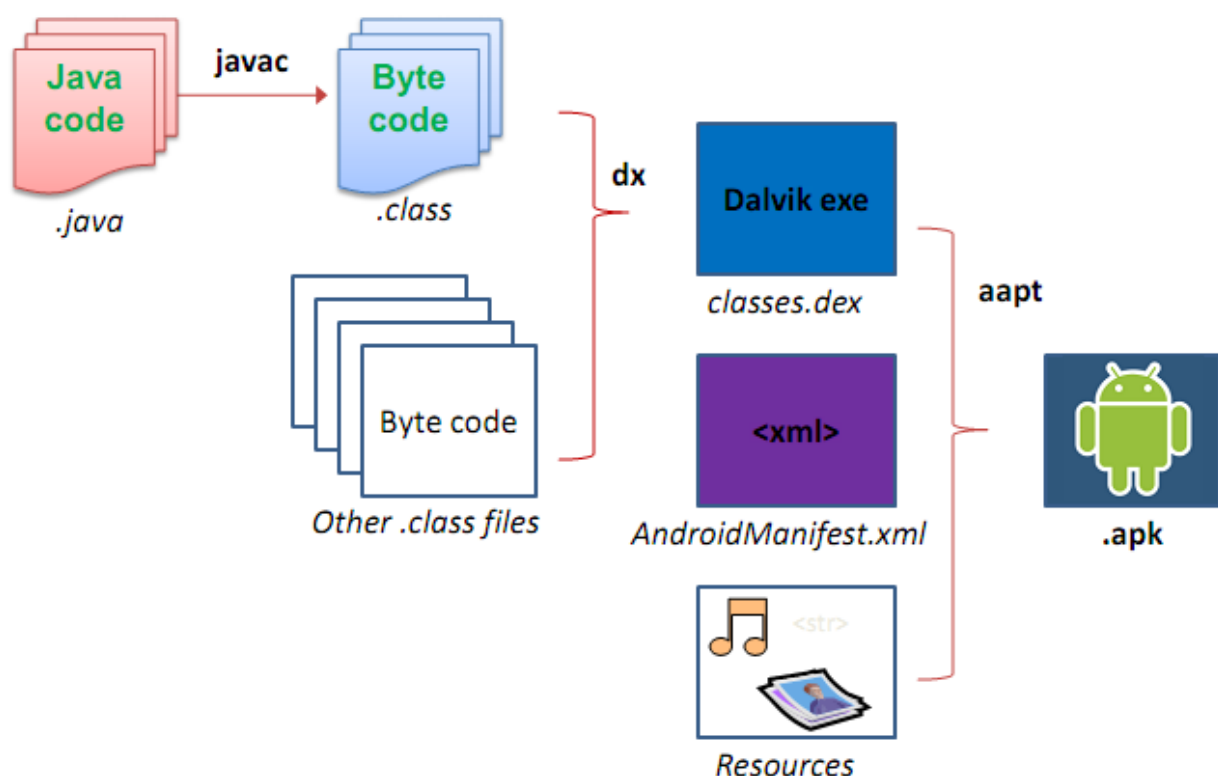


Figure: 1

The application source code is normal java code, it is a *.java, and it will be compiled from Oracle JDK. (javac). This will produced a class file, (byte code). There will be an optional third party tool called **ProGurd™**. This will be done for obfuscate the code, it will reduce the readability of the code, and so on. So decompilers cannot read the code correctly. Then regardless of the **ProGurd™**, then class file will be converted to *.DEX bytecode file. The dex file will be distributed via application package.

The Manifest File

A file which include all the component details of the application. The manifest does a number of things in addition to declaring the application's components, such as:

1. Identify any user permissions the application requires;
2. Declare the minimum application program interface (API) Level required by the application;
3. Declare hardware and software features used or required by the application;
4. API libraries the application needs to be linked against.

Targeting Android

Android is the target of 99 percent of the world's mobile malware, according to Cisco.[9] Cisco reported that the Android malwares are mainly spread as Trojanised applications designed to look like real, legitimate apps on third-party marketplaces.[9]

WHAT DOES IT DO?

Trojans are currently the most common type of mobile malware. Most of the Trojans we saw in Q1 2014 engaged in one (if not more) of the following activities:

SMS sending

Silently send SMS messages to premium-rate numbers or SMS-based subscription services.

Link clicking

Silently keep connecting to websites in order to inflate the site's visit counters.

File or app downloading

Download and install unsolicited files or apps onto the device.

Banking fraud

Silently monitor and divert banking-related SMS messages.

Location tracking

Silently track the device's GPS location and/or audio or video to monitor the user.

Data stealing

Steal personal material such as files, contacts, photos, and other private details.

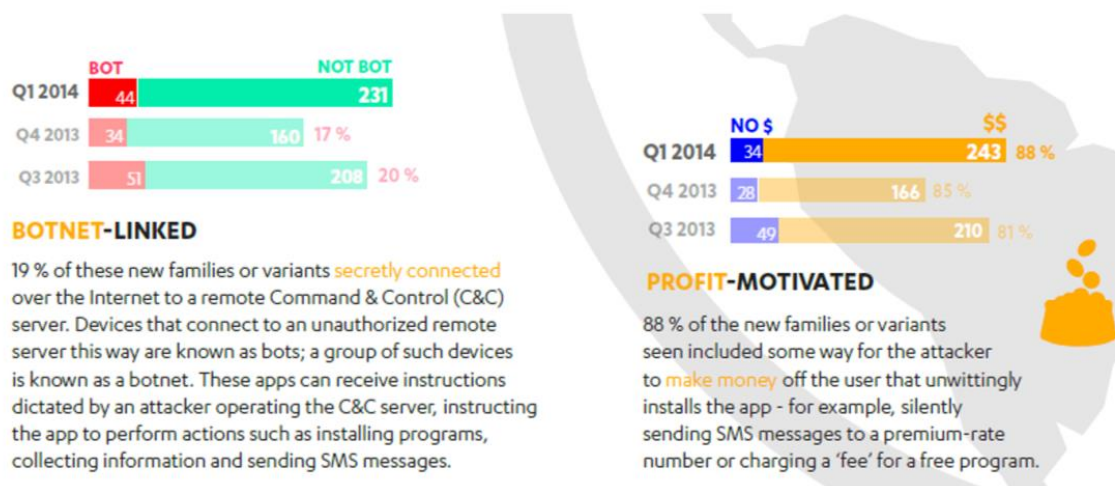
Fake app scanning

Pretend to be a mobile antivirus solution but has no

Fee charging

Charge a 'fee' for use/update/

Figure[9] : 2



Figure[9] : 3

What are the existing tools to catch Malware

The number of automated malware analysis tools can be found in the play.google.com for free as such as **Anubis** and **Mobile Sandbox**.

These are online tools which generates reports which are giving useful information such as permissions and URLs used by the application. Users can go through the reports and if there is a suspicion that an apk has been cloned and injected with code, then both apks (infected and uninfected apk) can be uploaded to a sandbox, and compared to discover differences in permissions, used features, URL's etc. So these tools give only a comparison only and further there are limitations in each tools.

Andrubis

Andrubis is the mobile user interface to the Andrubis analysis service. Andrubis executes Android apps in a sandbox and provides a detailed report about their behavior, including file access, network access, cryptographic operations, dynamic code loading and information leaks. In addition to dynamic analysis, Andrubis also performs static analysis, yielding information on e.g. the app's requested and used permissions, activities, services and external libraries. Based on the results of dynamic and static analysis Andrubis assesses the risk associated with an app in the form of a malice score between 0 to 10 (benign to malicious). This assessment is based on features learned from over 100,000 known benign and malicious applications and retrained on a regular basis.[10]

Note, that Andrubis only supports apps with a file size smaller than 8MB and an API level of up to 10.

Joe Sandbox Mobile

Joe Sandbox Mobile analyzes APKs in a controlled Android environment and monitors the runtime behavior of the APK for suspicious activities. All activities are compiled into comprehensive and detailed analysis reports.

Analysis reports, which contain key information about potential threats, enable cyber-security professionals to deploy, implement and develop appropriate defense and protections.

Joe Sandbox Mobile enables you to install and use Joe Sandbox in your lab. Currently Joe Sandbox Mobile analyzes any malware targeting Android-based operating systems. [\[11\]](#)

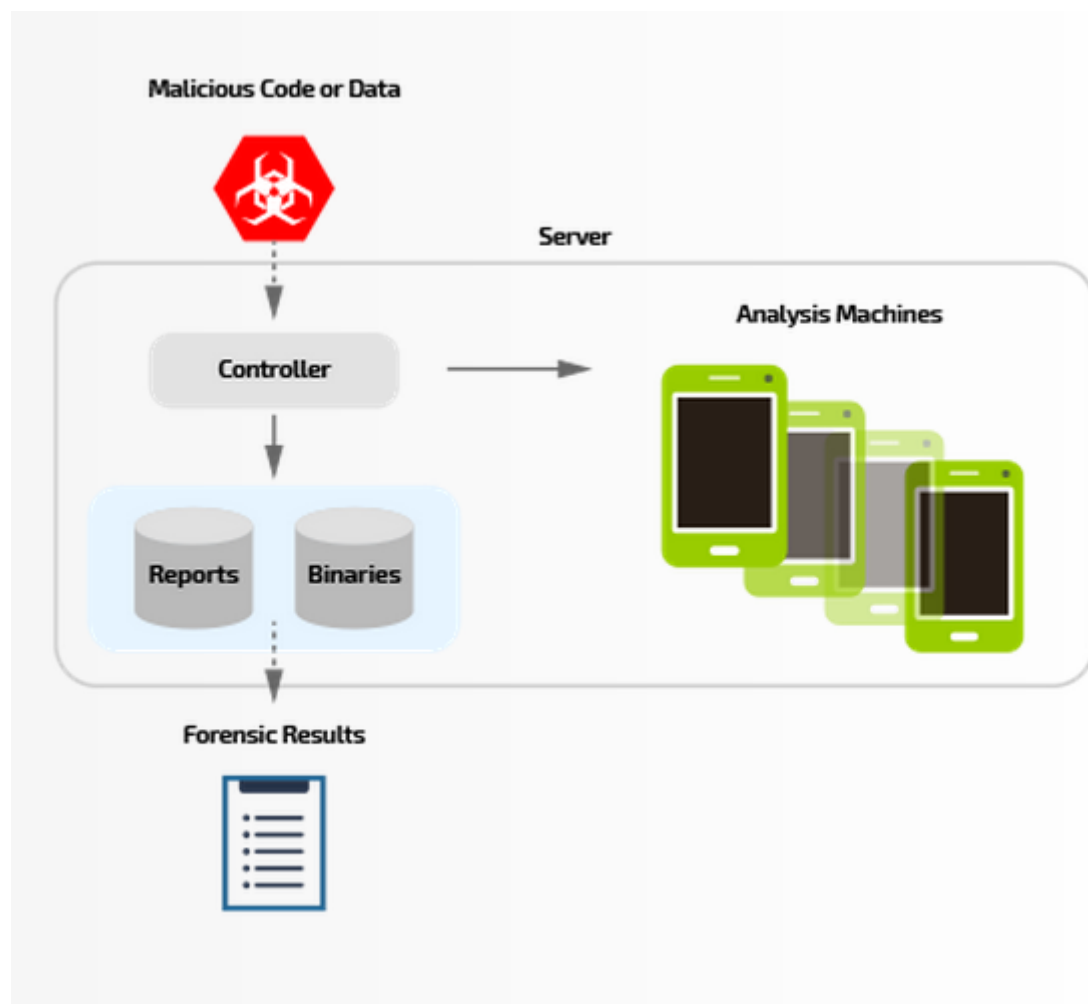


Figure 4:

Joe Sandbox Mobile's architecture is modular. It consists of at least one controller machine running Linux and multiple connected analysis machines (with Android installed) hosted by virtualization products such as VMware or VirtualBox. APKs are submitted from a user or submission scripts and sent via the Joe

Sandbox Mobile Web Interface to the controller's server. The Joe Sandbox Mobile server then stores the submissions in a local file database and forwards them to the connected analysis machines / phone, where the APK is installed and launched.[11]

Joe Sandbox Mobile's configurable and efficient instrumentation engine analyzes any activities during the APK execution and reports back behavior data instantly to the controller. Click to read more about Joe Security's unique technology called Hybrid Code Analysis (HCA) integrated into Joe Sandbox Mobile. [11]

Static and dynamic data is evaluated, and results, statistics, activities and code functions are compiled into a detailed and well-structured report.[11]

Inbuilt Android Security

The in-built security features of Android OS and the latest Android versions I have used here is:

- Android 4.4 KITKAT
- Android 5.0 Lollipop

It is should know that not any operating systems are 100% secure, but we can do so many things to protect our systems, but every protection techniques are not easy, it may giving so much inconvenience to users.

So considerations to be taken to protect your device is depending on following factors.

1. Is your device is valuable
2. Is your device has valuable data/information
3. Location of you is important

Passwords:

It is important to put a strong password to your device prior to taking any further security measurements. Develop a system to create a password and that should not easy to guess by others, the best practices to create strong password should be followed.

Dangerous software to steal password:

keystroke recorders.

Other type of attacks to steal password

- Brute force attack
- Phishing Scams
- Visual - video, etc.

Precautions:

1. Update your software: The bad guys are alway clever, and they finding security holes, but in the same time we can find fixes for those security holes, so it is necessary to update your OS with necessary updates. You must update both OS update as well as the updates to all the

applications you have installed. Fail to keep your software up to date is could leave you to open the security hole which are discovered and fixed by the software designers. Updating is one easy step to make your device is more secure.

2. Smart lock for trusted devices: Android lollipop version upward there is a feature called smart lock, which enables the Android device even there is a lock is configured when the phone is near the trusted device. So it is not very secure to have trusted device configured if it is not very trusted.
3. Smart lock for trusted places: This is also coming with Android lollipop version, which enables the device to users without putting the lock. So this is also not so secure feature when considering that there are intruders in every places.
4. Encrypting Android device: There is a configuration in Android OS to encrypt your device and the data when the phone is locked, this is a cool feature that will protect the data in the Android OS when the device is connected to the Computer systems.

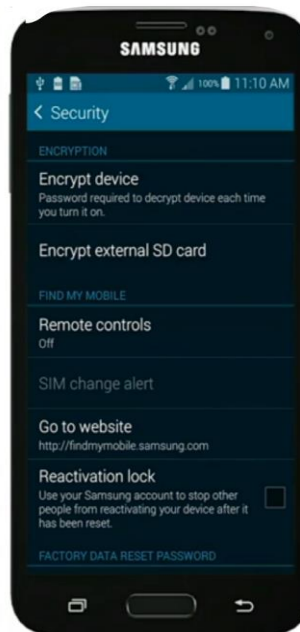


Figure 5:

- Enabling Android device manager: There are very handy two features giving by Android operating system that is to Remotely locate this device and Allow remote lock and erase options. Enabling these features are very handy when your device is lost or cannot find it. If your device is “On” and connected to the network, then it is easy to locate the device, putting a new pass codes and lock it, or wipe out all the data.

How to do it?

You simply log in to the google play and go to settings to access device manager, in there you can find out your device and take necessary security actions to protect your device or data.

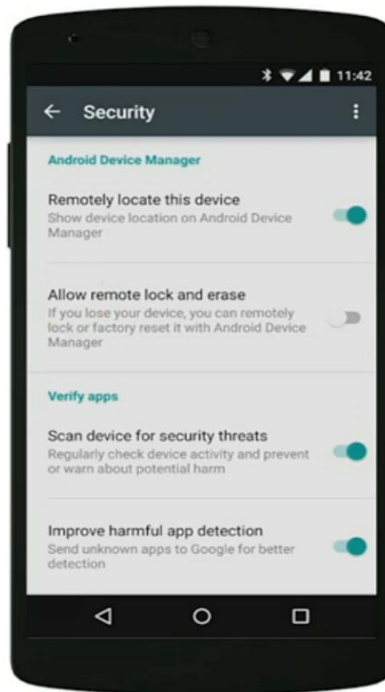


Figure 6:

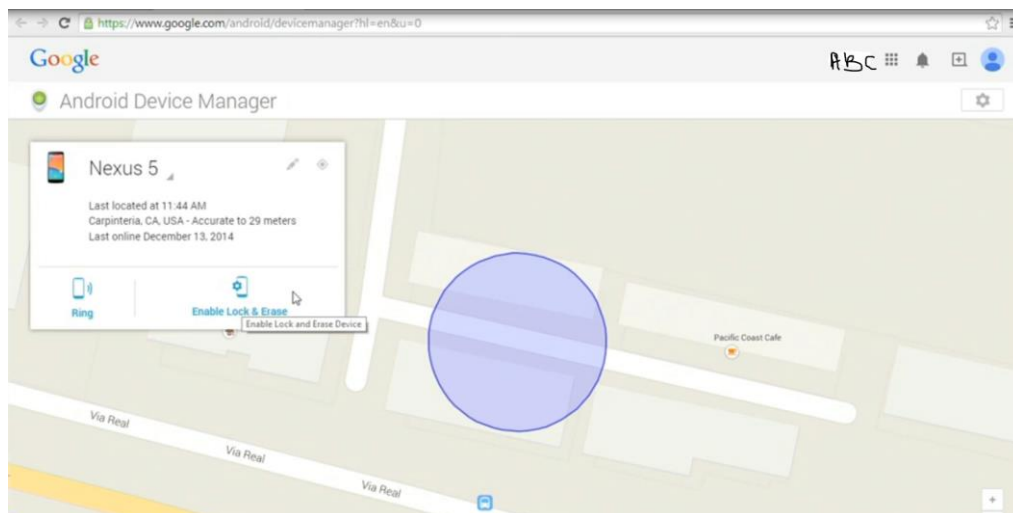


Figure 7:

- Restricting and get to know what app is accessing your personal data: Can you ever trust the developers who made these apps that they do not use your personal data for other purposes? i.e the facebook app can access your contact details etc. So what are the features

giving to restrict these access or see what type of data is accessed by the application. Android is giving these option to user. In the following example I show that facebook messenger works on the Android OS.

- When you installing facebook messenger you can see a user accept permission screen. In that you can see what are the required information to install this application.

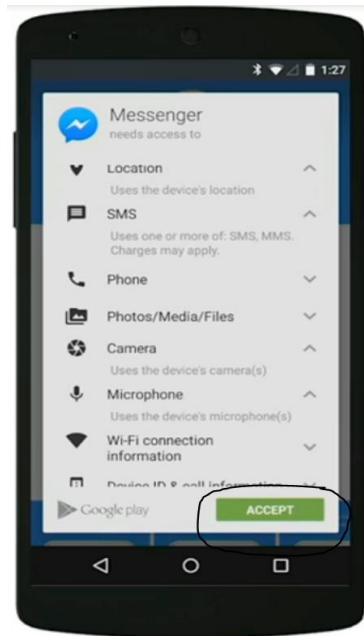


Figure 8:

Further you can confirm after installing the application whether that application has accessed to particular information only. For example in the facebook messenger, it has said that your location is being monitored by the application. So what you need to do it to go to settings > location > in that screen you can see all the list of applications has access to your location. Then you can find the fb messenger and find out what are the accessing information from this application.

The application not allowed to restrict each accessibility to information, so then you need to either uninstall the full application or allow it to access it. But the Android is giving option to turn off the location setting from the whole device, and this is a global setting. So in that you can restrict some personal information from the application.

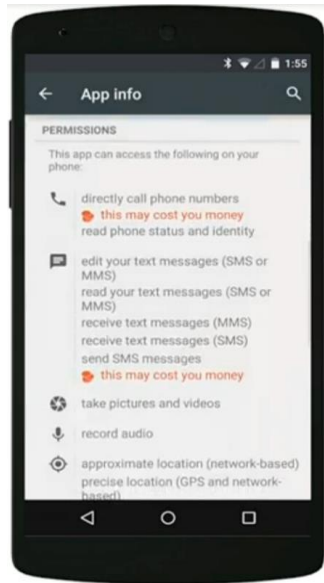


Figure 9 :

You should know that google is tracking your location through the phone, so anyone can able to access to your google account, then he can see what locations you have visited from the each day. Is this not worrying you? How many of us know this. And how many of us know how to stop this tracking.

Google account synchronized automatically of your certain information (example location) and share with other applications.

Example:

When you do a google search then the suggestion are automatically comes 1st to related to your location.

When you search xyz certifications, then it will give suggestions to xyz certification to your home town. How can that know your location. Further google can give you the traffic report in the morning that suggested that your location is keep tracked by the google. This is a handy feature, but how do you know that google is keeping these information in safe? or else some other person can access to your google account and find out your location.

Do you need to disable this tracking? and delete past location history? So this is allowed from the Android OS, by you can go to "google settings" > account history > google location history > location history : there you can delete location history or view your location history. Same information can be found when log in to the google via web browser and and search maps.google.com/locationhistory will show your tracked information.

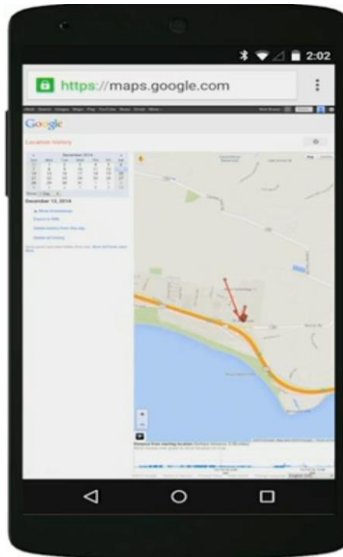


Figure 10:

That is only for location, so what else? yes it will track your browser history, youtube history etc.

Turning off browser history and search history : incognito mode

You can entirely switch off from google chrome account and stop syncing your browser history, else there is an option inbuilt in Android which is called Incognito Mode.

If you don't want Google Chrome to save a record of what you visit and download, you can browse the web in incognito mode.

Use incognito mode for private browsing

You can open an incognito window on your computer and mobile devices to prevent Chrome from saving your browsing history.

How to enable incognito mode for Android device?

- Open a Chrome window
- Touch the Chrome menu, which may look like either three dots Menu or a tab Menu.
- Touch new incognito tab.
- A new window will open with a gray incognito icon.
- To close incognito mode:
- If you have Android 5.0 (Lollipop) or later, swipe from the top and touch Chrome: Close all incognito windows.
- If you have an earlier version of Android, go to the corner of each of your incognito windows and touch the X.

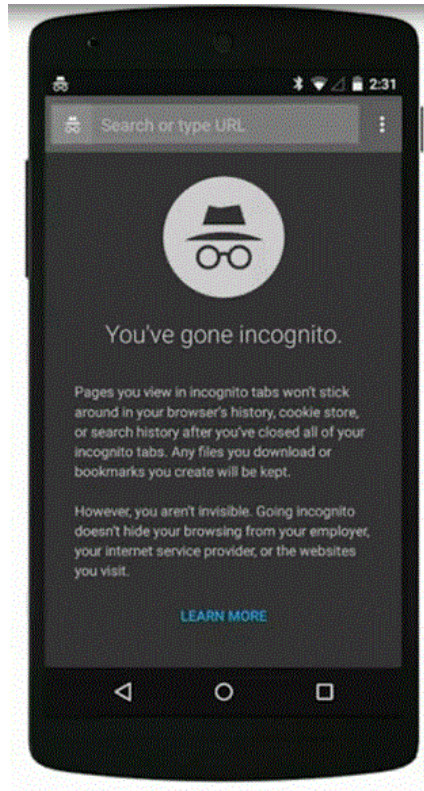


Figure 11:

Research Methodology

Android runtime (ART) is the managed runtime used by applications and some system services on Android. ART and its predecessor Dalvik were originally created specifically for the Android project. ART as the runtime executes the Dalvik Executable format and Dex bytecode specification. ART and Dalvik are compatible runtimes running Dex bytecode, so apps developed for Dalvik should work when running with ART. However, some techniques that work on Dalvik do not work on ART.

ART gives you as much context and detail as possible when runtime exceptions occur. ART provides expanded exception detail for `java.lang.ClassCastException`, `java.lang.ClassNotFoundException`, and `java.lang.NullPointerException`. (Later versions of Dalvik provided expanded exception detail for `java.lang.ArrayIndexOutOfBoundsException` and `java.lang.ArrayStoreException`, which now include the size of the array and the out-of-bounds offset, and ART does this as well.) For example, `java.lang.NullPointerException` now shows information about what the app was trying to do with the null pointer, such as the field the app was trying to write to, or the method it was trying to call. Here are some typical examples:

The Android Compatibility program defines technical details of the Android platform and provides tools for OEMs to ensure developer applications run on a variety of devices. The Android SDK provides built-in tools for developers to clearly state the device features required by their applications. Google Play shows applications only to those devices that can properly run those applications.

1st I will download the droidbox source code from the following repository:

Download necessary files and un compress it anywhere

```
wget http://droidbox.googlecode.com/files/DroidBox411RC.tar.gz
```

Setup a new AVD targeting Android 4.1.2 and choose Nexus 4 as device as well as ARM as CPU type by running following command.

```
android
```

Start the emulator with the new AVD:

```
./startemu.sh <AVD name>
```

When emulator has booted up, start analyzing samples

```
./droidbox.sh <file.apk> <duration in secs (optional)>
```

Getting test reports by analyzing samples

Download Android Studio

Android Studio provides everything you need to start developing apps for Android, including the Android Studio IDE and the Android SDK tools.

<http://developer.android.com/sdk/index.html>

Compile Droidbox in Android SDK and identify compilation errors

Fixing compilation errors

Install new DroidBox in the system and test sample applications

Cross check result with initial results to and find any run time problems

Correct runtime problems

Tools used in the research

Java

Android SDK

Linux Operating system

Nexus 4 as device as well as ARM as CPU type

Emulators (with the new AVD)

Procedures

It is important to choose the appropriate development lifecycle process to the project at hand because all other activities are derived from the process. We have chosen "Agile development process" into this project since it is an open source project and it needs better visibility to the stakeholders. Procedures and process are briefly described in this section.

In the project plan there are 10 Sprints included. Following table describes each and every sprint in detail.

Project Deliverables in each sprint:

- Sprint 1: Feasibility study
- Sprint 2: Project Proposal
- Sprint 3: Use case Designing
- Sprint 4: Implementing
- Sprint 7: Project Documentation 1
- Sprint 9: Stabilization
- Sprint 10: Presentation

Team Allocation: 1 members team is formed, 1hrs per day is allocated to the project from each member.

Project Proposal: This document is prepared. The purpose of this document is to give a brief idea for stakeholders to what is the project is all about, and why it is worth taking to the time to consider the project in the 1st place.

Use case designing: Explaining the main user interaction with the system, and depicting the specification of a use case. UML use cases are written, they are useful for presentations to management and/or project stakeholders, but for actual development you will find that use cases provide significantly more value because they describe "the meat" of the actual requirements.

Integration and implementation: Installation and Developing of the application in development environment. Compilation on ART environment is done in this stage, Then it will compile and fixing issues.

Project Documentation: Project documentation will be created during Sprint 8 and 9. This will completes the project documentation and presentation.

Stabilization and Verification: During Sprint 10 there is a contingency to fix issues. Also internal testing will be carried out after implementing the full system. All the reported bugs will be fixed after analyzing the bug reports. The completed system will be handed over to expert to get their expert feedback. This phase is call acceptance testing/user testing. The feedback of expert will be highly considered to improve the quality of the product. Prioritized issues and improvements are addressed immediately.

Budget Plan

Estimation Sheet

	Effort	Total
Project initiation		
sub total	5	15000.00
Customisation & Integration		
project proposal	4	12000.00
feasibility study	1	3000.00
Servers		
training	5	15000.00
development cost	20	60000.00
qa cost	10	30000.00
customer testing cost	5	15000.00
documentation cost	10	30000.00
Deployment		

	deployment cost	5	15000.00
	travelling		0.00
Project management			
	project management cost	15	45000.00
Hardware and software			
	hardware cost		100000.00
	electivity cost		5000.00
	software tools		0.00
	Total	0	345,000.0

Estimation Summary

Total effort estimation in man days 80

Rate per day : **3,000.0**
0

Reference:

- [1] Market Share Statistics for Internet Technologies. "operating-system-market-share". July, 2015. Netmarketshare - Market Share Statistics for Internet Technologies. 25th August, 2015. <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=9&qpcustomb=1>
- [2] The Mobile Phone Security Industry Analysis Re-port in the First Half of 2012. 25th August, 2015. http://www.anguanjia.com/?c=Notice&a=notice_view&id=299
- [3] Bose A, Hu X, Shin K G, et al. Behavioral detection of malware on mobile handsets[C]//Proceedings of the 6th international conference on Mobile systems, applications, and services. ACM, 2008: 225-238
- [4] Schmidt A D, Clausen J H, Camtepe A, et al. Detecting symbian os malware through static function call analysis[C]//Malicious and Unwanted Software (MALWARE), 2009 4th International Conference on. IEEE, 2009: 15-22.
- [5] Krishna Sandeep Reddy, Arun K Pujari. Springer-Verlag. 08th November 2006. Journal in Computer Virology. 25th August, 2015. <http://rd.springer.com/article/10.1007%2Fs11416-006-0027-8>.
- [6] Gizmo, Richards. "Gizmo's freeware." 19th April, 2015. www.techsupportalert.com. 25th August, 2015. <http://www.techsupportalert.com>.
- [7] R.Dhaya, M.Poongodj. Detecting Software Vulnerabilities in Android Using Static Analysis. Dept. of CSE, Velammal Engg. College, Chennai. 2014, International Conference on. IEEE
- [8] XIANGYU-JU. Android Malware Detection Through Permission And Package. Department of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China. Proceeding sof the 2014 International Conference on Wavelet Analysis and Pattern Recognition, Lanzhou, 13-16 July, 2014
- [9] Android users targeted by over 99 percent of mobile malware. June 2014. <http://www.v3.co.uk/v3-uk/news/2323418/android-and-java-top-security-targets-for-malware-and-hacks>
- [10] Anubis - Malware Analysis for Unknown Binaries. Spt 2014. <https://anubis.iseclab.org/>
- [11] Agile Sandbox for in-depth Malware Analysis on Mobile platforms. 2015. <http://www.joesecurity.org/joe-sandbox-mobile>
- [12] A tool for reverse engineering Android apk files 15 July 2015 <http://ibotpeaches.github.io/Apktool/>
- [13] Android package index. Android 6.0 r1 — 08 Oct 2015

<http://developer.android.com/reference/packages.html>

[14] Current Android Malware. 7th of October 2015

<http://forensics.spreitzenbarth.de/android-malware/>

[15] Android open source project, Dalvik and ART. <http://source.android.com/devices/tech/dalvik/>

[16] Android Cross Reference. (Lollipop 5.0.0_r2)

http://androidxref.com/5.0.0_r2/xref/art/

[17] Ref : <http://blog.dornea.nu/2014/08/05/android-dynamic-code-analysis-mastering-droidbox/>

[18] <https://www.honeynet.org/gsoc/ideas>

[19] <http://source.android.com/devices/tech/dalvik/index.html#features>