Find Out of who Services Secure or not secured in Android Device

Bhavna Verma
M. Tech Scholar
(CSE) Patel College of Science and Technology,
Indore

Prof. Abhilasha Vyas
HOD
(CSE) Patel College of Science and Technology,
Indore

Abstract - Today, everyone necessarily requires a smart phone. There are tremendous amount of possibilities and numerous scopes witnessed in different areas of the mobile world. With this rapid growth in mobile services the researchers are alarmed about the security threats and are working upon it by securing the infrastructures which support the online-interface and other distributed services. In this paper, we discuss about secure android application on android phone. The objective is to provide such services which can secure customers' android devices. This data security methodology of cell phones is quite novel.

Keywords— Android smart phones, Security, Vulnerability, Privacy password lock and unlock.

I. INTRODUCTION

Man has embraced mobile phones like a best friend. Among 7 billion people worldwide, around 4 billion smart phones and millions of tablets are in use. A smart phone has various utilizations like taking pictures, watching movies, listening songs, surfing internet, making bank transactions, using social media, calling etc. A smart phone user keeps all his personal and professional data in his phone. These devices are valued immensely as we keep all our records in it. The personal data carried by such mobile devices are very important. These important and sensitive data (account numbers, policy numbers and others) can cause trouble if the device is lost. At present, there is a huge number of populations which recognize the security threat of smart devices and personal computers. Still many do not realize the everyday threat while accessing their smart devices. As a matter of fact, around 32 percent of the population thinks that they do not require any software for securing their smart phone devices. But this mindset and rising fame of smart mobile devices have attracted cyber crime up to an extreme extent. Over the period of months the attacks have increased and reached up to 37 percent. This has become the topic of discussion at this hour and several reports have been published for the same. These reports also talk about the susceptibility of the android smart devices. The susceptibility of smart phones is about 99 percent as mentioned by a survey. These personal phones and tablets are actually a type of mini-computers and they are more prone to vulnerability than desktops. Therefore, a mobile user must be very careful about his smart device and install a simple application and put a password to safeguard his data.

II. MOTIVATION AND OBJECTIVE

A. Motivation

Our mobile devices will have the next-gen operating system instead of desktops or mainframes. Along with the existing services available online, the current novel environments of developmental approach has opened space for newer applications and markets and have facilitated bigger integration. As the services of data supported smart phones have risen, the chances of susceptibility have increased. The next generation should provide the crucial wide-ranging security infrastructure. Android is the most widely used operating system of smart phones. Android has covered a huge amount of developers working for it and making applications and new products every day. The main reason behind its success is that Android lets its developers effortlessly extend phone services online. The chief impetus behind this project is to provide security and privacy to a smart phone user and it gets easier when Android gives easy access to developers. Majority number of smart phones and tablets are programmed to have only two control states; Lock and unlock. Accessibility of a smart device control is investigated by all-or nothing by a smart phone user. With user preferences we discovered all-or-nothing device access control to be extremely unfit. The participants demanded to keep their certain applications available even after locking and protecting the phone via authentication. Hence, to avoid any unwanted share of your crucial data and information one should be aware of the borrower. However, in the current scenario, only owner of the phone get an access to it because the OS which enables a single-user-mode. Thus, it was assumed that there is an urgent demand to prohibit the borrowers to access certain applications to install or uninstall. Below, there is a description about prototype of implementation of detailed model that expands the currently available Android app framework. The one propounded is one of the accessible methodologies which provide protection to user from the data personal.

B. Objectives

In this paper our main objective to provide who services secures or not secured in android device.

1. Root Access Detected.
2. No Hooking Framework Detected.
3. Device Lock Enabled.
4. App Data Backup Recovery Disabled.
5. Device Encryption Inactive.
6. Device OS Outdated.
7. Development Options Enabled.
8. Network Connection Enable.
9. No Emulator Access Detected.
10. No Debugger Detected.

III. BACKGROUND ABOUT ANDROID OS

A. Application Behavior

The Android application runs by a UID as a different process in itself whereas unlike desktop which has a single UID upon which all the applications work. The unique quality of UID is to safeguard all the data which is stored in it. The programs in Android are unable to perform any particular operation on each other, that is to say, data reading or writing etc. The unambiguous data sharing should take place between the applications. Because of such characteristic, Buffer over Flow attack (a compromise) has been restricted to apps and their data. However, it must significantly be put into notice that any other program can be launched by an application which will work according to the launch of UID of application. It is a necessary requirement that a developer signs his or her application before running an application on an Android device. Later developers use these self-signed certificates for code-signing. To update own applications hassle-free, code-signing is done by a developer.

B. Files and preferences

UNIX-style file permissions are used by Android. Each app owns a file system exactly like a home directory which aids with user IDs. This characteristic feature serves only Internal-memory. Always one question s raised “How does other applications are provided with the exposure of private data i.e. Standard method of applications?" The answer is that the exposure of private depends upon the Content-Providers.

IV. EXISTING APPLICATIONS FOR SECURITY

In Android phones the Google play store has many apps which provide the necessity of password protection of videos and images. Every app in the store is built under an application layer and each of the app has pros and cons of its own. Some of the most frequently used Android security apps are listed below.

1. Root Access Detected.
2. No Hooking Framework Detected.
3. Device Lock Enabled.
4. App Data Backup Recovery Disabled.
5. Device Encryption Inactive.
6. Device OS Outdated.
7. Development Options Enabled.
8. Network Connection Enable.
9. No Emulator Access Detected.
10. No Debugger Detected.

A. Android File Protector

Advantages:

1. Safety: When one downloads this app he experiences the level of privacy and feels safe about it. No one other than the owner himself can use the phone or sees the files in it.
2. Easy To Use: It is very easy to operate. Anyone can download and use this app.
3. Guarantee: This product gives a whopping 30 days money-back
   Guarantee i.e. if a person is dissatisfied by the performance he can get his money returned.

Disadvantages:

1. Locks: This app slows down the speed while locking and unlocking. Accessing files will be less quick than normal as files which are locked take time to get unlocked.
2. Passwords: One must ensure oneself of remembering the password as forgetting it would be time consuming and contain a little hassle.

B. File Locker

Advantages

1. This app smoothly works in protecting and encrypting or your files on an Android smart device.
2. The unauthorized access is protected.
3. The encoding of file is done by this app which ultimately makes the files unreadable.

Disadvantages

1. Forgetting password will be troublesome. You might need to generate new password every time you lock a file.
2. The password that you set is not stored on the phone. While we unlock the phone a simple checksum is performed. So, it s advised that one must not slip away his password or he will be unable to open files.
3. When the file gets locked the owner receives a notification on the phone about the change of file name. It is suggested that the encrypted file name remains unchanged as the app is incapable to restore the original name and by the result of which the file remains locked for good.

C. Easy File Locker

Advantages

1. As the name suggests it is comparatively easier to utilize.
2. The remarkable quality of this is app is that it keeps on protecting the personal files even when one does not run the program.

Disadvantages

1. Encryption algorithm is not utilized by this app instead of which simple set of Cascade Protection Levels are being used. The CPL consists of permissions to write, access, provide visibility and erase.
2. Over all protection is not guaranteed.
3. Simple and plain interface is offered.

V. EXISTING METHODS

A. At System Level

Boot has been performed by the kernel which was attached. One must have to go to bootloader.com and needs to connect this to USB port of the computer system which has Android SDK installed in it. Requires to locate the fast-boot command in that particular folder. Fast-boot boot-z image and boot it. Kernel source has to be downloaded from the tcdev.com vivo-ics-crc-3.0.16 and source has to be extracted and moved to the directory. For kernel compilation arm-eabi-4.4.3 cross compiler has been used. For making boot.img Android kitchen has to be installed. The unused attribute bits has been used as flag to denote if the file is locked or unlocked. These bits are accessible in fat.h in the msdosinodeinfo structure. At low level, these unix file internals may not be similar as fat internals. But the one that is similar to unix is syscall which internally manages and controls vfat. So, the 'rm' command will call 'unlink' as syscall which Kernel detects as vfat. It will entually performs the operations of vfat. As we do not delete anything we are less likely to be bothered about the Free-List. Sometimes we might hide a data and if the inode which is hidden is put on the Free-List, the Kernel might assume it as a deleted file and could utilize it for the very next allocation of files. So, by that very way we can say that it is overwritten. Well, there is no different opening for fat. But there is namei(). In namei-vfat.c, namei gives out inode when the given path file works as parameter. And one can verify if that inode has some hidden bit-on. We will be able to return error on namei itself which will progressively return to that particular application where the file seems to be not existing. We have introduced a new operation ioctl in inode-operations. In such operations the bit can be changed on the basis of requirement. A 'C' file which contains an invocation of the ioctl call can easily be cross compiled for a device. We need to push it through the adb and then call it from the adb-shell. We must have to cross compile it and in order to complete this task we need to use arm-abi tool chain.

B. At Application Level

The whole project is completed and constructed in Java. In order to develop the rear-end of the project utilization of the Eclipse IDE is done. It has all the technicalities to construct a java application. Many around the globe have recognized that this is the best tool present for developing Java. For Java Developers the Android Studio do provide superior editing along with validation, cross-referencing, code assist, incremental compilation; an XML editor.

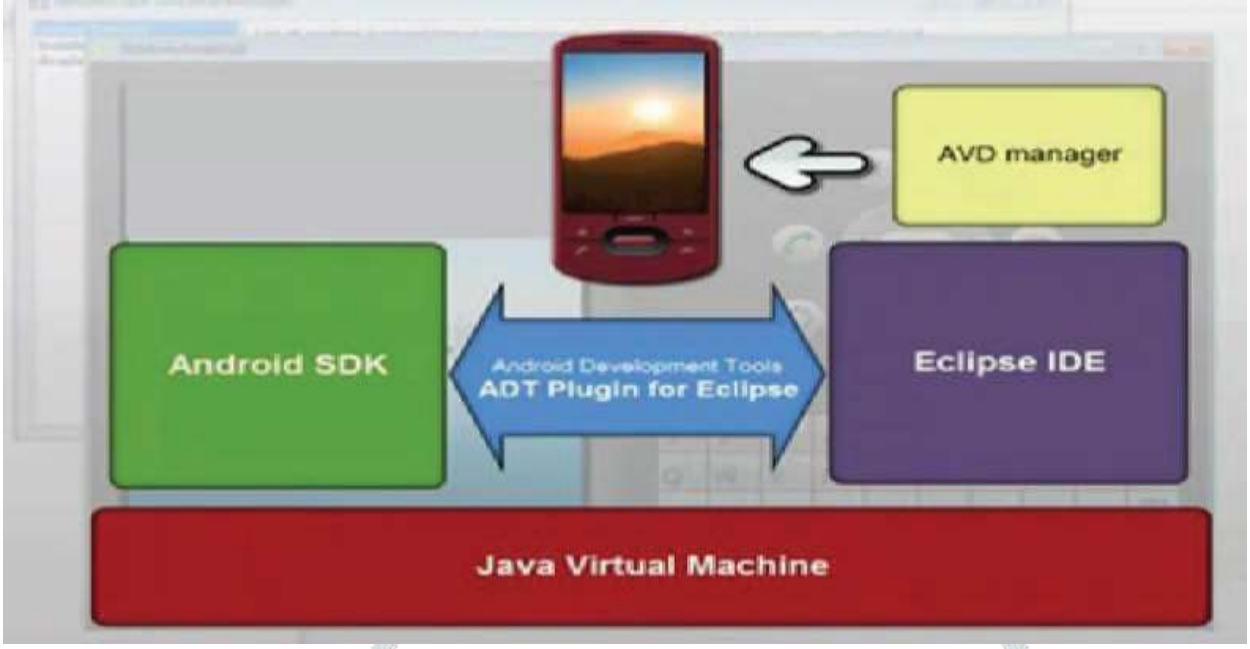


Figure 1: Android Application Development

The developing environment is set by the help of the Android-SDK. An all-inclusive set of tools like debuggers, libraries, various hand-set of emulators are included in the Android Software Development Kit (SDK). The necessary developer tools and API libraries are provided by it to test, debug and build apps for the Android. To rationalize the app development for Android the ADT Bundle consists of the important A-SDK constituent and edition of the Eclipse-IDE along with a built-in ADT. For our existing environment the SDK-manager tool aids to get updated tools and novel Android-platforms installed. SDK manager along with the help of A-SDK split platforms, tools and other various components into some packages that we can later download.

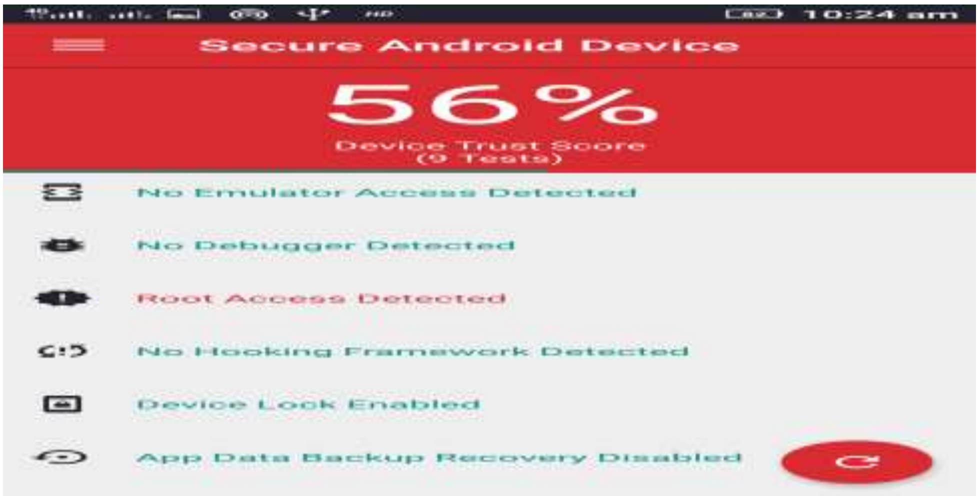
The SDK launching can be performed in one of the given ways: From Eclipse (with ADT), select Window – Android SDK Manager. On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory. The powerful android applications can be built by the help of ADT plug-ins. When we need to swiftly set-up new Android projects the Eclipse capabilities must be extended, UI has been constructed to run an app, by using A-SDK tools debug the application, and even APKs (signed or unsigned) are exported for distributing the application. The swiftest way to get started with is by using the Eclipse constructed applications which used ADT. When these projects have guided setup, they provide custom XML editors; debug output pane and tools integration. An incredible boost is received with the development of Android apps when ADT is involved. At last, to intermingle with the server the Android Virtual Device (emulator) works as the source. The figure shows The Android Application Development Architecture.

VI. PROGRESS

Some scenarios claim that the data in an Android smart phone is accessed by the external applications without the knowledge of the owner. The security issues are of the major concern in the Android phones. The main purpose is to cease the unintentional loss of data to external apps by providing security. The data is personal as well as very important to the owner of the phone; therefore, it must not be exposed at any cost. In order to achieve this task and safeguard the customer's data we are propounding novel and effective mechanisms.

VII. RESULTS

In this paper we get on the basis of 9 parameters we find how many percent secure data in android device, we represent in Figure 2



10:24 am

Secure Android Device

56%

Device Trust Score
(9 Tests)

No Emulator Access Detected

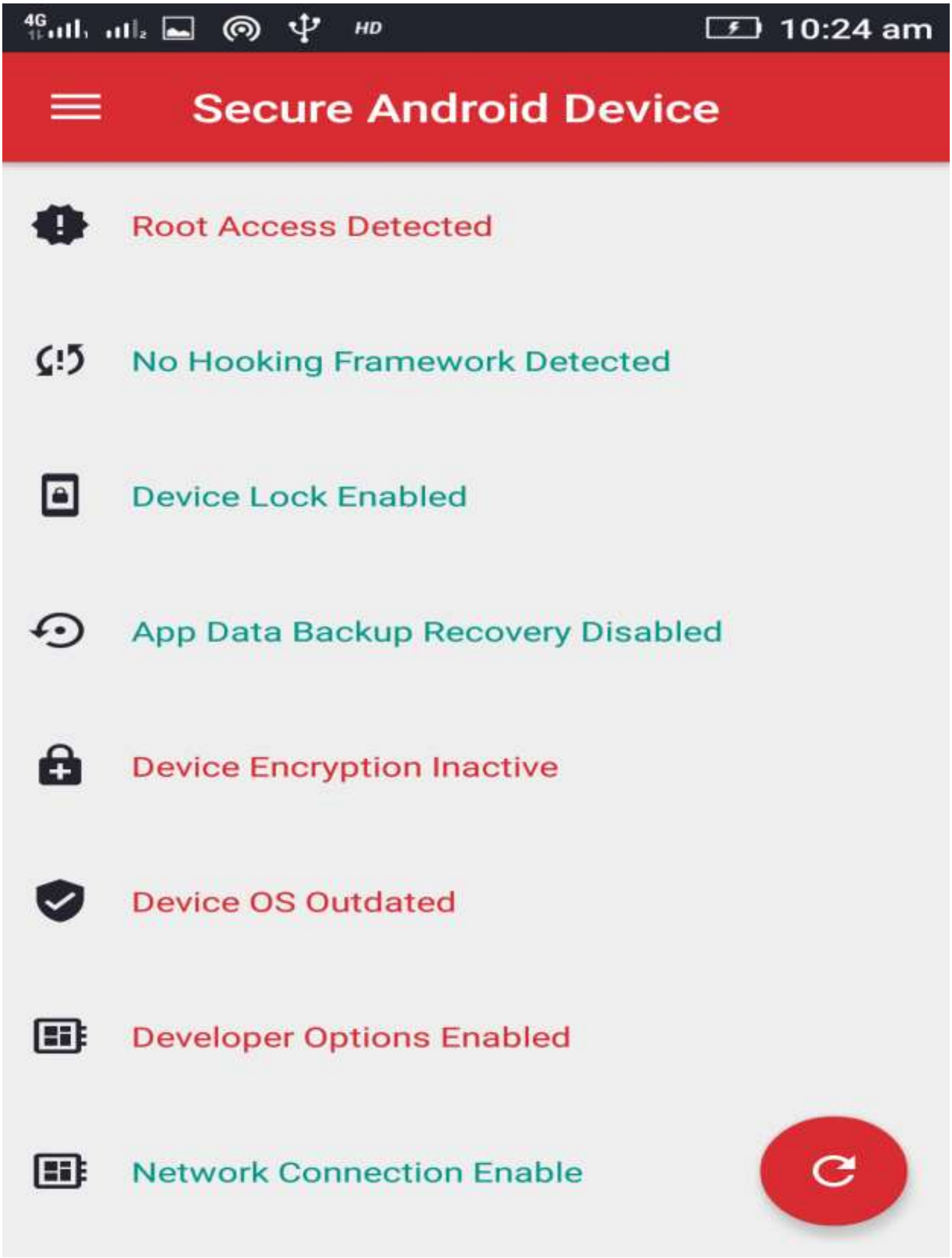
No Debugger Detected

Root Access Detected

No Hooking Framework Detected

Device Lock Enabled

App Data Backup Recovery Disabled



Secure Android Device

Root Access Detected

No Hooking Framework Detected

Device Lock Enabled

App Data Backup Recovery Disabled

Device Encryption Inactive

Device OS Outdated

Developer Options Enabled

Network Connection Enable

Figure 2: Check how many percent secure our android device.

Advantages:

1. Easy access is one of the main lookouts in any application. This app fulfills this demand very well indeed.
2. The GUI is very unique yet very simple.
3. The Open Source development of this app makes it free of charge to download and use.
4. Understanding this app is very precise and simple.

5. You can relieve yourself from worrying about securing your files and choose the files which one to show others and which ones not to. Figure 2 shows the how many secure our android device. If our phone is unsecure then will be apply immediately high level security.

VIII. FUTURE ENHANCEMENTS AND CONCLUSION

The enhancement would suggest that simply in place of 9 features any other features can be utilized at application level in case if security is to be offered. According to the some discussions earlier security can be provided at Kernel level as well. The key which is used for file is stored by each key which is maintained by a Key-File. An associative key mapping table has to be made for doing so; keeping in mind that one must keep the records of the "n" files which support only a single key.
Now-a-days, the omnipresence of smart phones is taking the world by storm. However, we understand the necessity of the security systems and the applications which are supported by them. The main area of concern is that the infrastructures for the security of these systems are still in need of major development. The phone apps security constraints now are augmented by the current version of Android OS with an objective of overcoming them. There is a need of safe and secure app interactions which can safeguard the personal chats, SMSs, video and audio data etc.

At present, many of the smart phones are susceptible to various attacks by the architecture present in the security system and due to the user's role in installing a particular application on his phone. It is actually very difficult for a user to decide which app to install on the basis of a mere description. The security consciousness of the user is continuously being expected by the Android framework and ensures customers to believe that the app developers are not sinister at all. Due to the above mentioned scenario it could happen that the user unknowingly downloads the software which might contain a technical threat for him. The major task here is to develop a system that provides its customers ease to stay worriless with the issues that troubles them from installing an app that does not bring justice to their personal data from being disclosed. Thus, a built-in system allows its user to simply access and chooses the particular files that have to be safeguarded and hence, secure them from others. It has also crossed our attention that the security framework of an Android is limited to the storage of its media present on the phones.

REFERENCES

1 H. Jonsson, “Text Mining of Personal Communication,” IEE`E 2010.
2 Shabtai A., Fledel Y., Glezer C., “Google Android: A comprehensive Security Assessment,” IEEE Security and Privacy, 2010.
3 Deepali Mulani, “How smart is your Android smart Phone,” San Jose State University, May 2010.
4 W. Enck, M. Ontang and P. McDanial, “Understanding Android Security,” IEEE Security & Privacy Magazine, 7(1), 10-17, 2009
5 Avik Chaudhari, “Language-Based Security on Android,” ACM 2009.
6 Shabtai A., Fledel Y., Elovici Y., “Securing Android- Powered Mobile Devices Using SELinux”, Security & Privacy, IEEE, 2009.
7 Ongtang M., McLaughln S., Enck W. and McDaniel P., “Semantically Rich Application- Centric Security in Android”, Processings of Te 25th Annual Computer Security Applications Conference, Honolulu, Hawaii 2009.

8 Schmdt A. D.,”Enhancing Security of Linux-based Android Devices”, 15th International Linux Kongress, Germany 2008
9 Neal Leavitt, “Technology News”, Published by IEEE Computer Society June 2011.
10 Neal Leavitt, “Technology News”, Published by IEEE Computer Society June 2011.
11 Zemin Liu, Choon Sung Nam, Dong Ryeol Shin, UAMDroid, “ A user authority manager model for the Android platform” , (ACM), 2012
12 Woongryul icon .Jccycou Kim. Youngsook l, “A Practical Analysis of Smartphone Security”, (Springer Heidelberg), 2011.
13 William Kuck, “Defending users against smartphone apps: Techniques and future directions”, (Springer Heidelberg), 2011
14 Robert Love,Linux Kernel development
15 Daniel Bovet,Understanding Linux kernel
16 Eiji Hayashi,Oriana Riva, Karin Strauss, A.J. Bernheim Brush Stuart Schechter, “ Goldilocks and the Two Mobile Devices:Going Beyond All-Or-Nothing Access to a Devices Applications”, (Symposium On Usable Privacy and Security (SOUPS)), 2012
17 Burns J Developing secure mobile applications for Android, Available from: https://www.isecpartners.com/files/ iSEC-Securing-Android-Apps.pdf
18 Xudong Ni, Zhimin Yang, Xiaole Bai, Champion, Dong Xuan, DiffUser: Differentiated user access control on smartphones, Mobile Adhoc and Sensor Systems, (IEEE 6th International Conference), 2009
19 http://developer.android.com/index.html
20 http://www.android.com/market
21 http:// 7 Applications To Help You Hide Images And
22 Videos On Android Phones.com
23 http://Very-Android-File-Protector-Review.htm
24 http://Understanding-UNIX-Linux filesystem-Inodes.htm
25 http://[KITCHEN]Android-Kitchen-v0-223-[Linux-Mac- Windows]-xdadeveloper.html
26 http://jsnyder-arm-eabi-toolchain-GitHub.html
27 http://developer.android.com/guide/basics/what-isandroid.html, 2007