The Eighth International Conference on Business Information Security (BISEC-2016), 15th October 2016, Belgrade, Serbia

ANDROID FORENSIC USING SOME OPEN SOURCE TOOLS

ISAK MRKAIĆ
University of Donja Gorica, Humanistic Studies, isak.mrkaic@gmail.com

Abstract: *In recent years Android operating system, being installed on huge numbers of smartphones, tablets and other devices, had a breakthrough on the market. Following that success, the need to recover and analyze data from Android OS, became important part of mobile forensics. Consequently, many commercial and open-source mobile forensic tools became available for forensics investigators. The subject of this paper is to present open source-free tools and to illustrate how to forensically recover data from Android based devices.*

Keywords: *Android OS, forensics, data acquisition, open source forensic tools.*

1. INTRODUCTION

Advancement of mobile phones came along with the technological expansion of the 21st century. There has been a substantial increase of production, development and use of smartphone devices in recent years [1]. Consequently, Google's Android operating system is in the lead when compared to the competition: in 2015 Android operating system's market share has risen to 81.2%, while the competing systems like IOs, Windows and others took up 15,8%, 2,2% and 0.8% respectively [2]. Given that smartphones are used to exchange messages, emails, photos, etc. and that the considerable amount of data stored on them could be acquired and analyzed, forensics of mobile devices became a substantial segment of digital forensics in general.
Mobile phones are frequently used in criminal activities, so the law enforcement services consider them to be a significant source of evidence. The Boston Marathon bombing [3], uncovering of a child prostitution ring [4], attempt of bomb attack on Times Square in New York [5] and similar cases are just a few of numerous examples of mobile forensics and Android forensics used to investigate criminal activities.

The market offers some commercial programs that could be used to carry out a part of the forensic process related to the acquisition and analysis of the data from Android mobile devices. In spite of that, there is also a vast number of the open source tools that could be used to carry out certain forensic tasks. The hypothesis we set for this research is that the combination of various open source tools could be used to acquire a certain amount of data from Android devices, which, when analyzed, could be used as valid evidence in court of justice. Using these tools would cut down the expenses of buying the commercial software and enable forensic scientists to gain additional insights into Android forensics given that open source solutions require "step-by-step" approach.

In this paper we elaborate on one of the models of the forensic process and then we offer the framework of a possible realization for the part of the forensic process that is related to data acquisition and analysis. Having this framework in mind, concrete open source tools were used to retrieve certain data from the tested mobile device. The retrieved data was then processed in the adequate programs.

Limitations of this research relate primarily to the fact that just one model of the phone with the Android operating system installed was tested. This limitation also reflects one of the greatest challenges mobile phone forensics faces: multitude of mobile phone models, operating systems and their versions on the market. Also we have only focused on some data we considered important (sms messages, call logs, emails…), so we haven't explored all of the potential of given open source tools.

We cannot safely claim that the given data could be used as valid evidence in court of justice given that this largely depends both on legislation of the country where a trial is conducted and the concrete forensic process.

Regardless of the limitations, some of the tools examined could be used in the actual forensic processes and they would be applicable to all the versions of Android and some other operating systems.

2. METHODOLOGY OF THE FORENSIC PROCESS

Forensic investigation on mobile devices comprises of the procedures which are defined by the phases that should be completed in order to round out the forensic process. It could be stated that there are no univesally accepted procedures, so they vary depending on the author's preference. Consequently, different methodologies of the forensic process have developed over the years.

In 2011 the group of scientists developed one of the all-encompassing models which also proved to function excellent in practice. This model, is known as **SRDIFM (Systematic digital forensic investigation model)** [6].

During the **preparation phase** the investigator gets familiar with the case and makes proper preparations. Following the preparation phase is **securing the scene**, then **survey** and **recognition** which imply making the initial plan as to how to collect and analyze the data. **Documentation of scene phase** comprises of making a

sketch map of investigated area and documenting all of the electronic devices on site, including the device itself with its equipment. Once this phase is completed, it is necessary to do **communication shielding**, i.e. to prevent the device from connecting to any mobile network, WiFi, etc. RF isolation, Faraday shielding, cellular jammers, which is followed by **collection** of all the available evidence. It is recommended to connect the device to the charger immediately. Once evidence is collected, it should be **preserved** and transported for further analysis. Collected evidence is **examined** in the way that analysis and filtering are performed. In addition to that, integrity of the data should be secured. During the **analysis** phase the results obtained in the previous phases are used to perform a thorough technical review of the evidence. This phase also sees use of the techniques more advanced than those used in the research phase. This level also implies the analysis of the hidden data, file recovery as well as file decryption. All the results obtained must be documented in order to complete the final report which sums up the entire process in the **presentation** phase. Finally, the results obtained are publicly shown [6] [7].

Reliability of the data depends directly on the methodology of the forensic process. Leaving any of the phases out during the investigation results in unreliable digital evidence which will not be valid in court of justice. As we have already stated before, there is no universal standard of the forensic process on mobile devices. It is up to a forensics scientist which model he will use and apply during the investigation. In this paper we will focus on the phases of research and analysis, i.e. we will explain ways of taking certain data from a confiscated device which is considered as evidence, as well as ways of processing the data in adequate programs thus making them usable for further presentation.

3. DATA ACQUISITION

Completing a forensic process requires preparation of the proper work environment. For that purpose one must own a personal computer of the adequate performances that is free from malicious programs and which has installed a software needed for the forensic work. Some of the programs needed are Android studio i Android Software Development Kit (SDK), mobile device drivers and tools for the work of the forensics scientist. It is recommended to use the original cables of the device on which we perform acquisition to connect it to the computer.

Depending on the possibilities, several methods of acquisition could be used. The first, and the simplest one is the manual acquisition in which the investigator searches the menu of the phone and examines the files that are accessible in user interface thus collecting data. Limitation of this method is that the investigator only acesses the data visible through user interface. The second method, called logical acquisition is performed by exstracting data from te file systems which are visible on logical store of the mobile device. This method usually does not allow recovery of the data erased, or allows it to a limited extent. Physical acquisition is the most detailed and the most complete type of file exstraction since it implies making a copy of the entire memory content [8].

Data obtained using these methods are often in form of unprocessed data and after the extraction is complete, several actions are performed in order to obtain information which is understandable and readable. Further investigation is done using the copy of the data which are obtained by acquisition, thus securing the integrity of the original data on the mobile device.

Copies of the taken data will be processed in an adequate forensic program. Data we aim to obtain are general phone data, contacts, call logs, sms messages, internet search history, social network applictions (skype, viber, facebook) data, email messages, photos, video recordings, etc.

Files obtained by using these methods are then processed in programs that extract useful data for further analysis.

4. PRACTICAL ASPECT FRAMEWORK

The first step, as we have already stated, is to form a forensic work station and to install the required programs that are to be used during the data acquisition and processing. Following that is identification of the adequate cable to connect the device to the PC work station, then installation of the drivers for the mobile device which is the object of investigation. There is always a possibility that the Android version was changed or modified by the manufacturer, so it is highly recommended to download the drivers directly from their website. Once the adequate drivers and the cable are provided, the Android device is connected to the work station. While setting a connection up, the mobile device itself offers the options for the type of protocol it will use to communicate with the PC work unit. The latest models of Android use either Media Transfer Protocol (MTP) or Picture Transfer Protocol (PTP).

One of the challenges that could be faced during the extraction of data is how to unlock the phone protected by the PIN code or some other kind of protection. There is no such thing as the universal method of going around these protections. As is the case with majority of mobile phones, the protection related data is stored on **logical locations** and they can be **taken over** using the pull command, and then extracting the protection related data from the downloaded files [9].

Any further work requires Android Debugging Bridge (ADB) which is installed within Android Studio and Android SDK. If ADB is to work, we need to select the option USB Debugging on the mobile device thus enabling it to communicate with the PC. This option is usually found within Settings-Developer options in the phone menu.

Once the Android phone is successfully connected to the PC, ADB is started and the command adb.exe. devices or adb devices enables us to find out if the device is properly connected. The possible statuses we get after the given command is issued: offline – the device is not properly connected, i.e. there is a problem in communication, device- the device is properly connected and no device which means that device is not connected to the PC [10].

Once we set the connection up, the extraction of data from the Android device could be started. Adb shell

command enables us to access the Android shell further manipulation of the device. It is recommended to provide root access to the phone if possible. Further work requires the knowledge of the basic commands-functionalities that would be used. Some of the commands needed are:

adb pull – used for transfer of files from the mobile device to the forensic station – PC. It is important to note that the access to the majority of important files requires root privileges [11] [12].

dd – this command is used to make a bit-by-bit copy of the device, i.e. for the physical acquisition of data. Copied data is transferred to a separate SD card which is forensically clean. If we try to copy the files to the existing SD card in the phone we will make changes to the original data. Upon creating a file –data.img with the copy of the data in the card, we can transfer it to to the PC by using the **pull** command or taking the card out and inserting it into work station. If we don't have a forensically clean SD card, the file could be directly transferred to the PC while its being created. This is done by using netcat and nandump programs [13].

Once the copy of data from a mobile phone is created, the data is processed using various programs. For this purpose the program Autopsy, which is considered to be one of the standard programs when it comes to data analysis, could be used. There are also other options such as SQLite browser, SQLite forensic explorer etc. There are also open source solutions which automatize the process and perform the complete acquisition and analysis. One of these programs is NowSecure Forensics Community Soluttion.

5. THE RESEARCH

For the purposes of this research we used the phone Alcatel One Touch 6012x with Android version 4.2.2 (Jelly Bean). The working station is a laptop Dell Inspiron 15, 4 GB RAM, 32-bit Operating System, Intel Pentium 3558U, with Windows 7, SP 1 installed on it.

A. Connecting To The Android Device

We connected the device with the original cable to the laptop and then turned on USB Debugging option. Given that this option was not available in the Settings menu, we turned it on by clicking several times on the option **Build number** in the **Settings-Abut** phone menu. That gave us access to the Developer menu and the option USB Debugging became available.

Once activating the option, we started the adb program to check whether the Alcatel had proper communication with the laptop. Since the device was connected, the command and the response were as follows:

```
C:\Program Files\Android\platform-tools>adb.exe devices
List of devices attached
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
7LLFIRZPR4JZR8BI        device
```

B. Unlocking The Phone

To unlock the phone we used the program Andriller [14] which, apart from the other options, offers this possibility as well. It is a commercial program with the possibility of the free license for the 15 day period, upon expiration of which the program must be purchased. It is free for a six month period for the law enforcement officers. To unlock the phone, we processed the file gesture key with the above mentioned program. The gesture key we previously acquired from the phone through the pull command. After executing the command, we obtained the actual unlocking pattern (Image 1).

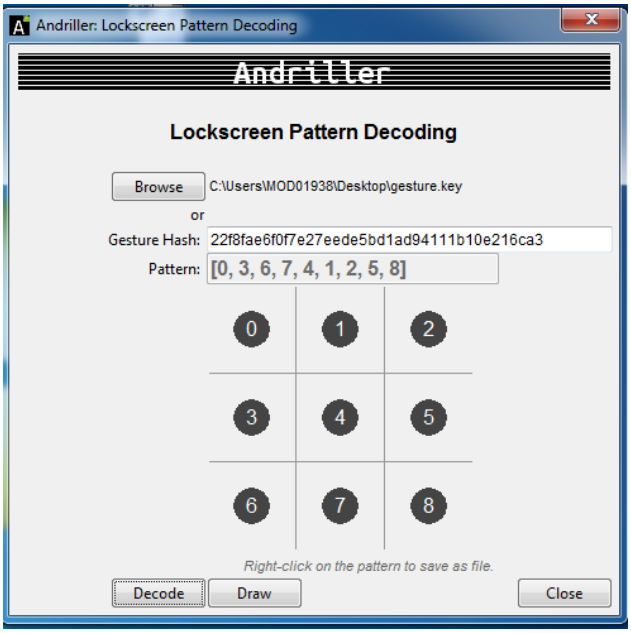


Image 1: Lockscreen pattern decoding

Open source solution would be to open the gesture file in hex editor, and then use SQLite Browser so as to compare the hash values of gesture.key with generated hash values of possible keys in order to get the unlocking pattern. The scripts could be generated in Python [9].

C. Rooting The Phone

Root access to the phone we gained after downloading the application KingRoot from google play and installing it directly on the phone. After installing it we connected to the device through ADB and then executed the command adb shell to enter the user account. Upon executing the command, we got the option to allow root access to adb application. Once we allowed it, we had the root access to the device and # appeared in shell (Image 2).

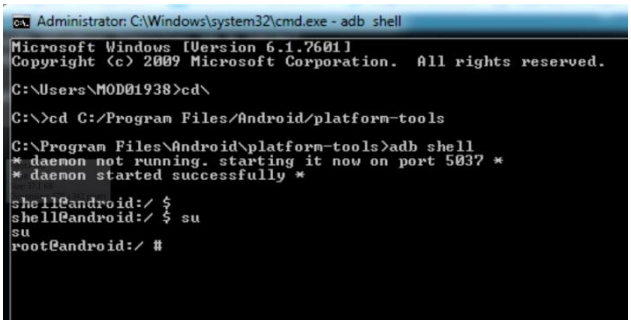


Image 2: Accessing shell as "root" user

D. Data Acquisition

The first method of acquisition was realized through using adb pull command which enabled us to copy the data from the folder to the destination folder on the C partition.

adb pull /data/data C:/forenzika/seminarski

The folder contained the data that could be interesting for further inspection:

- data from the SIM card which are stored on logical destination **/data/data/com.android.providers.telephony/databases/mmssms.db.**

- data from the SIM card about contacts and call logs **/data/data/com.android.providers.contacts/databases/contacts2.db.**

- data from the SIM card about email messages on gmail which were acquired from **/data/data/com.google.android.gm/databases/mailstore.isak.mrkaic@gmail.com.db.**

The other method was realized through executing **dd** command. We inserted the empty SD card in the phone and the device recognized it. The command **#cat /proc/partition** was executed to learn about the existing partitions, which made us decide to acquire the contents of the flash memory **mmcblk0**

Next step was to execute the command

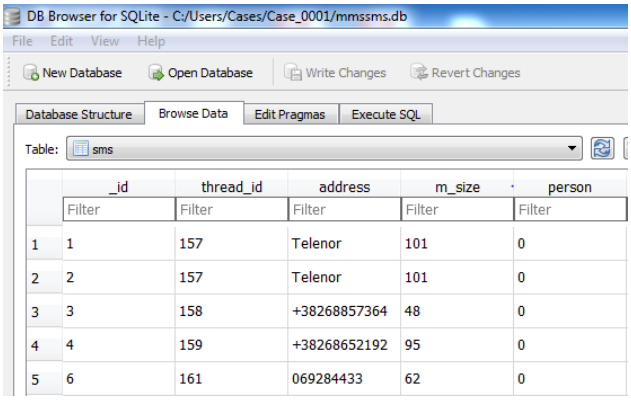
dd if=/dev/block/mmcblk0 of=/sdcard/data.img bs=512 conv=notrunc,noerror,sync

After the command was executed, the file **data.img** was created on SD card along with the folders android_secure, Android, googleota, KingMaster, LOST.DIR i e_config. We copied the files to the laptop simply by taking out the SD from the phone, inserting it into the SD card adapter and then inserting the adapter into laptop and copying the files to the D partition.

E. Data Analysis

The data acquired through logical acquisition, i.e. through executing the pull command, were processed by SQLite Browser [15] which enabled us to read **.db** files that seemed to be of interset to us

- **mmssms.db** file contained data about all of the sent and received SMS messages. More precisely, it included the phone numbers of both the sender and the receiver of the message, the date when it was sent, the message status (whether or not it was received), as well as the content of the message (Image 3).



	_id	thread_id	address	m_size	person
	Filter	Filter	Filter	Filter	Filter
1	1	157	Telenor	101	0
2	2	157	Telenor	101	0
3	3	158	+38268857364	48	0
4	4	159	+38268652192	95	0
5	6	161	069284433	62	0

Image 3: SMS messages

- The file **contacts2.db** contained data about the call logs, namely the phone number of the caller and the receiver, call duration and the date when the call was made.

- The file **mailstore.isak.mrkaic@gmail.com.db** contained the data about gmail messages including the address of both the sender and receiver, time when the message was sent and the message content.

The data we got through physical acquisition of the flash memory acquired through dd command were analyzed using the program Autopsy [16]. This program is a free one and is intended for the analysis of majority of the files on Android (such as YAFFS, .ext, etc.).

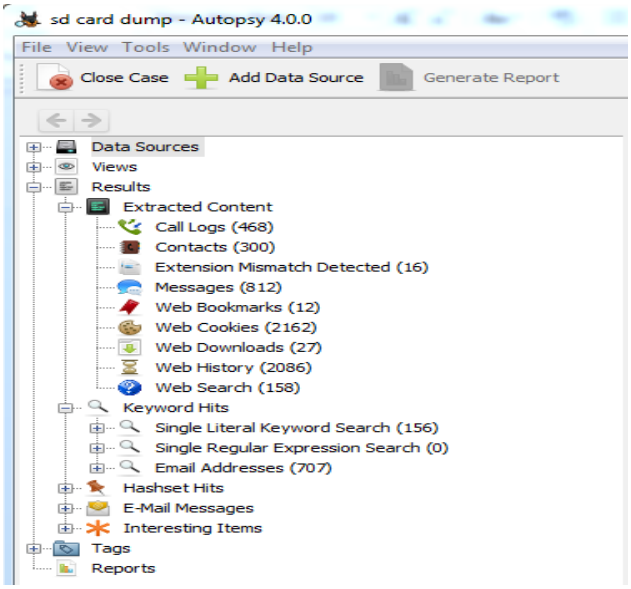


Image 4: Data acquired through Autopsy

As the result of the actions performed, we could access contacts, call logs, messages, etc., but also some of the files erased (Image 4).

F. Other Tested Open Source Tools

AF Logical OSE [17] is the program intended for data extraction through Content Provider. This program enables us to acquire some basic data such as SMS/MMS, contacts, logs, calendar, etc.. By using this program we managed to get the data about the SMS messages, call logs and contacts (Image 5).

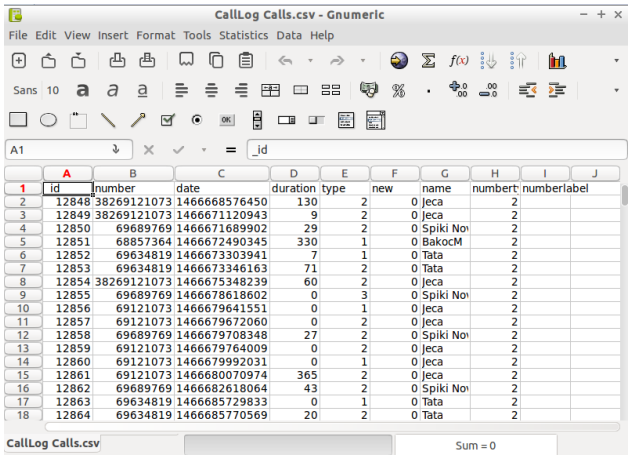


Image 5: Call history

For logical acquisition and analysis of data we used program **Mobledit** [18], version 5.5.0.1148, which enables us to work in Lite regime if we don't want to buy the license. However, there are also some limitations, such as inability to export the files. Upon the installation and command execution we managed to set up a successful communication between the laptop and the Android device, so we could access the basic information about the SIM card, search the contacts, call logs, SMS i MMS messages and access the calendar.

NowSecure Forensics Community Solution [19] (up until a year ago it was called **ViaExtract)** is a tool which enables logical and physical acquisition of data. Its manufacturer is NowSecure. Free version enables logical exstraction of data, while the purchased version enables both logical and physical exstraction.The program is downloaded within the virtual surrounding and is installed on Santoku linux based on Ubuntu distribution. After being started, the program automatically recognized the mobile device This program also offers the option to root the phone by using certain exploits. After starting it, we proceeded to start Backup acquisition, Logical acquisition and File acquisition (Image 6).

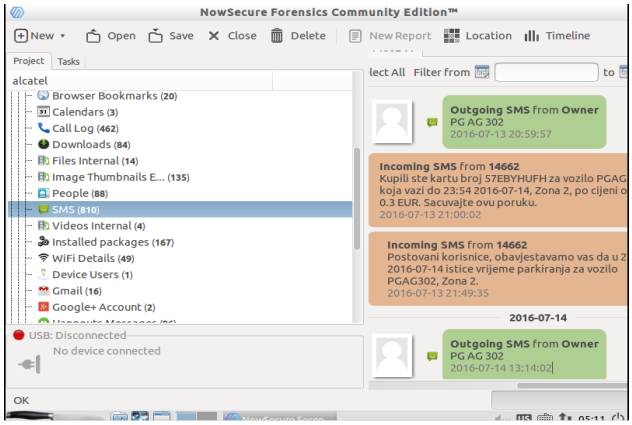


Image 6: SMS messages

NowSecure Forensics Community Solution is a great free tool which enables us to acquire a lot of data in short period of time.

6. RESULTS

On the phone that was used for testing we performed logical and physical acquisition of data, and then the analysis of the results obtained. The programs used as well as their basic characteristics are outlined in the table below.

Table 1: Comparative analysis of programs characteristics

Program	Acquisition	Analysis
NowSecure Forensics Community Soluttion	Yes	Yes
ADB (pull, dd)	Yes	No
Autopsy	No	Yes
SQLite browser	No	Yes

The most complete data from the device that was the object of our research were retrieved through the NowSecure program which performed the acquisition and the analysis automatically, yet it was not possible to extract the results from the program to the PC. Similar results were obtained by step-by-step acquisition by using dd functionality and then analyzing the file obtained using the programs Autopsy and SQLite browser, however it was possible to extract the data to the PC.

Table 2: Comparative overview of the results

Data retrieved	**NowSecure (Backup, Logical and File acquisitions)**	**dd, SQLite browser and Autopsy**
Audio	59	76
Images	270	506
Videos	4	4
Deleted data	-	66220
Viber Calls	513	513
Viber Participants	458	458
Viber Messages	9072	9072
Web History	20	2086
Contacts	36	300
Call Log	462	468
SMS	810	812
Gmail messages	16	16

The comparative overview of the data retrieved is given in the table above. The overview does not include all the data, just those we considered to be the most important. Based on the results, we can safely claim that a significant amount of data can be retrieved by using the open source tools.

7. CONCLUSION

This paper explores open source tools which enable data acquisition and analysis. Even though there is a multitude of these forensic tools, they are all limited in their functionalities and it is only by their combination that we can get satisfactory results.

Android forensics using open source tools is a challenging task and it largely depends on the tools available. Commercial programs like Encase i Oxygen Forensic™ Suite are user friendly and they offer a wide range of possibilities. Unlike them, open source programs must be used in combination to achieve comparable results. Limitations of the open source tools are reflected in the fact that they cannot offer ready solutions. Namely, after certain programs are used to perform acquisition and analysis, it is preferable to use a combination of different programs to compare and consolidate the data if the results obtained are to be valid. Also, majority of the open source tools are limited in their functionalities, so it happens that the analyzed data cannot be retrieved, or some functionalities only have trial versions. This in turn requires more time for extraction and analysis of data, then it is the case when commercial tools are being used.

Having expansion of the Android platform in mind, it is expected that the number of available forensic tools will rise in near future and their functionalities would develop. In this paper we used the latest researches and the up-to-date literature, yet some links to the forensic software already became unavailable or outdated.

One of the examples is how the program NowSecure which we used for the purposes of our research is no longer available on the manufacturer's site. In the light of these facts, it is obvious forensicists, apart from having the knowledge of the field must also constantly follow trends and latest findings in the field

REFERENCES

[1] http://www.statista.com/statistics/263441/global-smartphone-shipments-forecast/, Accessed August 01, 2016.

[2] http://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/, Accessed August 01, 2016.

[3] http://forensicstore.com/the-boston-marathon-how-smartphones-are-changing-investigations, Accessed August 01, 2016.

[4] http://forensicstore.com/taking-down-child-prostitution-rings/ , Accessed August 01, 2016.

[5] http://www.forensicon.com/forensics-blotter/cell-phone-email-forensics-investigation-cracks-nyc-times-square-car-bombing-case/ , Accessed August 01, 2016.

[6] Soufiane Tahiri, Mastering mobile forensic, Packt Publishing 2016., page 5.

[7] Ankit Agarwal, Megha Gupta, Saurabh Gupta & Prof. (Dr.) S.C. Gupta, "Systematic Digital Forensic Investigation Model", International Journal of Computer Science and Security (IJCSS), Volume 5, Issue 1, 2011, Available: http://www.cscjournals.org/manuscript/Journals/IJCSS/Volume5/Issue1/IJCSS-438.pdf

[8] Rohit Tamma, Donnie Tindall, Learnig Android Forensic, Packt Publishing 2015., page 8.

[9] Rohit Tamma, Donnie Tindall, Learnig Android Forensic, Packt Publishing 2015., pp 128-136.

[10] Rohit Tamma, Donnie Tindall, Learnig Android Forensic, Packt Publishing 2015., page 50.

[11] https://developer.android.com/studio/command-line/adb.html, Accessed August 06, 2016.

[12] Andrew Hoog, Android Forensics: Investigation, Analysis and Mobile Security for Google Android 1st Edition, Syngress, 2011,.page 218.

[13] Rohit Tamma, Donnie Tindall, Learnig Android Forensic, Packt Publishing 2015., pp 147-152.

[14] https://andriller.com/downloads, Accessed 08. Jully 2016.

[15] http://sqlitebrowser.org/ , Accessed 08. Jully 2016.

[16] http://www.sleuthkit.org/autopsy/, Accessed 08. Jully 2016.

[17] https://santoku-linux.com/howto/howto-use-aflogical-ose-logical-forensics-android/, Accessed 12. Jully 2016.

[18] http://mobileditlite.en.softonic.com/download, Accessed 12. Jully 2016.

[19] https://www.nowsecure.com/forensics/community/, Accessed 12. Jully 2016.