Executive summary

This project presents a method for acquiring forensic grade evidence from Android Mobile devices using open source forensic tools, and performs a comparative evaluation of the data extracted from these tools. Tools are tested on different criteria, to critically evaluate the performance.

The report begins with the details about the Android architecture, file systems, android device rooting, etc., and then taking over to the major aim of this project. There are a number of forensic tools available, but there is no standard comparative framework constructed to evaluate the performance of these tools. These tools form a basis for deciding the outcome of an investigation, which is a highly legal matter. For this reason the performance of these forensic tools is important to be tested. A comparative framework of these tools will help decide which tool is better for forensic investigation purposes.

The investigation done is based on two test cases, in which suspect make use of smartphone's facilities' like Bluetooth, applications etc. Data when performing the test cases is recorded manually. These records act as the benchmark for comparison with the data acquired using the tools. Further critical assessment of the tools was based on the factors like CPU utilization, RAM utilization, and Load average on CPU etc. All the records were tabulated for visual representation, and to minimize the efforts to look at the numbers. Performing tests such as Precision-Recall and CHI Square test validated the results. Subsequently this report can be used as a comprehensive approach to be followed by forensic examiners when dealing with crimes associated with Android smartphones. The methods used in this report can be used to perform acquisition of Android device and inspect them safely in order to learn about any illegal acts related with Android Smartphone.

The research carried out and reported in this thesis consists of comparison and evaluation of four Android forensic tools leading to the following insights.

- Amongst all the four tools tested, ADB was the better tool in data extraction. Recovering highest amount of relevant information about the activities done by the suspect's smartphone.
- When tested on parameters like CPU utilization, AFLogical tops the list.
- Methodologies used for evidence extraction are illustrated and locations to be targeted for evidence files are specified.
- Report can be used as a complete manual by a forensic examiner to follow when performing investigations dealing with crimes conducted using Android smartphones.

Acknowledgement

I would like to show my gratitude to Dr. Theo Tryfonas for being my supervisor for this project and all the help and guidance he provided. I thank Mr. Panagiotis Andriotis and Mr. Shashank Singh Kushwah for being my internal supervisor and being a constant support throughout this project. I also thank my family for moral support. Finally I thank to University of Bristol, for providing a golden opportunity to work in this area.

List of Tables

List of Figures

Table of Contents

Introduction and Background 1

Mobile phones have transmuted from just being a simple voice communication device to a high-tech device with the capabilities of computer. They can be considered as a small computer in pocket, which always travels along with the person and hence act as a crucial part of digital evidence. According to Ericsson [24], Global mobile penetration reached 85 percent in 2011 and mobile subscriptions reached to 6 billion. With the increasing use of mobile phones, new technologies and devices with different operating systems have emerged into market. Android operating system is one such mobile platform. It is open source and run by Open Handset Alliance (OHA). OHA is an association run by Google.

Increasing computational power of mobile phones is successful in replacing computers and laptops, and hence proved to be having a greater potential for misuse. New features such as GPS, Bluetooth hacking etc. are being used to perform criminal activities. Another example can be, mobile devices are used to perform mobile banking through online applications available, and are major focuses of criminals to steal money and information. As the technology is developing people are adapting the use of mobile phones, they exchange images, videos, use internet etc., this rapid development of mobile phones serve both criminals and investigators in a same way. According to Bruce Schneier[25], Mobile phone's microphone can be used as a transmitter to eavesdrop the conversation taking place nearby the device (Schneier, 2006).

Android Forensics provides a number of prospects and challenges in the field of digital investigation. A good understanding of the Android platform and Forensic tools is fundamental to extract and analyze forensically sound data. A plethora of forensic tools are available in market, helping to make forensic techniques more sophisticated. This research report investigates open source android forensics tools, against their ability to extract information from Android devices, based on disparate parameters [2].

1.0 Aims and Objectives

There are extensive range of Android Mobile devices available, and a reasonable number of forensic tools to investigate them. But there is no standard approach to analyze the performance and efficiency of these tools. A significant level of efficiency is expected. When evaluated with various parameters the efficiency of these tools can be measured.

The main aim of this research project is to evaluate and compare the forensic tools used to extract evidences from Android mobile devices.
Objectives to back the aim are:

- To research about and understand the area of Android forensics
- Distinguish the best approaches presently used for Android forensics
- Define assessable tests, and record the expected outcomes to be obtained from forensic tools
- Detail the steps used for implementing the tools and record the results
- Set a standard to describe the results of forensic tools on Android phones. Critically analyze the tools based on the standard set and actual data recorded manually during experiments.

1.1. Background

1.1.0 Introduction

The mounting growth of mobile phones and their computing capabilities increases the reliability on them and hence are likely to be helpful in some form of digital evidence. Crimes like terrorism, intrusions to corporate or government IT systems, homicide, etc. highly rely on this form of digital evidence. With the advancement in digital forensics, it helps in refining the practices of digital investigation. Felons alongside are getting more aware of the digital investigation capabilities and are developing more complex use of computers and networks to commit crimes. Anti-Forensic tools and methods are even developed in order to conceal the activities and destroy digital evidence. New forensic tools and techniques have to be introduced for preserving digital evidence and extracting exact information from the device used for crime. Standards devised for the analysis of the toolsets should comply with both technical and legal aspects.

The fundamental step in understanding of the forensic process is the knowledge about the categories of evidences found on mobile devices. Evidence found on mobile devices can be multifarious, based on its model and its usage, whether it is a basic phone or a smart phone etc. Data can be retrieved from three places namely internal memory, external removable memory and SIM cards, but not all components can be justified as useful digital evidence. There can be multiple SIM cards, removable memory or mobile phones. Criminals are practicing using multiple phones so that the evidence can later be destroyed, but still useful information can be obtained from damaged phones or SIM cards. Another fact is that not all mobile phones are same and have same capabilities; mobile handsets with different operating systems and functionalities are available and hence increase the complication amongst the set of evidences. Getting to understand the full functionality of mobile phone helps in a better investigation. For example manufacturer documentation can provide a lot of information about the same.

Categories of evidence on Android mobile devices

Types of Evidences		
Traditional Mobile devices	• Hardware • Information set by user • Information set by device	• Date and Time, IMEI (international Mobile identity number) • Address book, to-do lists, Calendar items, etc. • Call history
Smart Phones – Android	• Information set by device • Internet usage details • Third-party applications	• images, video/audio, maps, MMS, GPS waypoints, stored voicemail, files stored on system, connected computers. • Passwords, online accounts, email, social networking information. • Malware applications, communication systems, or anything that can provide an alibi.
Local workstation	• Information transported	• Back up of phone data, third party applications, and acquisition media.
Network Provider	• Details of whereabouts • Usage information	• Different locations, current location. • Billing details, call history overtime, internet data usage.
External Memory	• Information stored by user	• Files, images, media, everything can be stored as there is availability of a large amount storage space.

Mobile Forensics is a fast paced field, which is exciting and has a powerful effect on various situations comprising criminal investigations, national security, civil litigation etc. It evolves from the digital forensics discipline giving us a variety of opportunities and a number of challenges.

When talking about the most interesting part of Android forensics, which is acquisition and evaluation of the data acquired from the devices, it is important to have ample amount of knowledge about the platform and the tools that are used during the investigation. A broad understanding will assist the forensic examiner or security professional to perform successful investigation and analysis of the Android device.

1.1.1 Android Platform

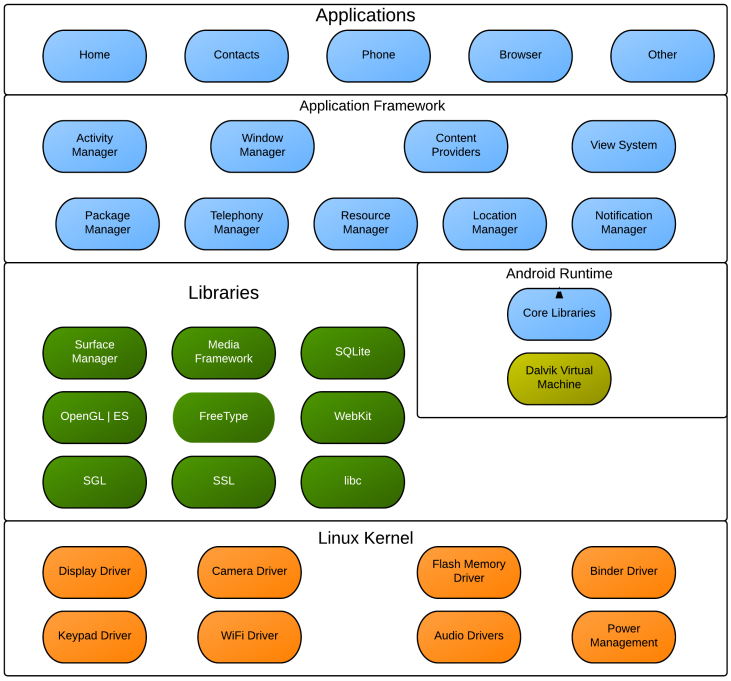
Android is an open-source mobile device platform, based on Linux 2.6 kernel. Android has made a major impact on the Smartphone market and subsequently in the area of Mobile Forensics. Android became the second largest smartphone platform after two years and a month of its release [1]. Table 1.1 shows the Android market share and number of activations per day [39].

Table 1.1

Period	Worldwide Smartphone market	Activations per day
2009	8.7%	51,100
2010	32.9%	362,000
2011	52.5%	658,000
2012	59%	934,000

Android Architecture [2]

Figure 1.1



In order to understand the logical analysis processes, it is important to have knowledge about Android architecture. The Linux 2.6 kernel provides the basic software essential to boot and manage the Android Applications. As detailed in figure 1.1, the low level functions include Wi-Fi, power, display etc. But when viewed from Forensic perspective the most important is Flash Memory Driver. In the libraries field, SQLite library provides a structured data storage and is an area that can be focused on by forensic experts.

Linux Kernel

It is the core component of Android platform. It controls the hardware and its resources; controls power management, memory management and maintain different drivers for all hardware of the device.

Libraries

It is the next layer of the Android architecture. Libraries are written in C/C++. The core of 3D and 2D graphics constitutes of openGL/ES and SGL. Media resources like MPEG, mp3, etc. belongs to Media framework. SQLite is used to provide a method for structural data storage.

Android Runtime

Dalvik virtual machine was built to achieve an efficient and secure mobile application environment. All the applications are run on their own Dalvik VM. The low-level functionalities such as memory management, access to core libraries etc. are provided using Dalvik VM and rely heavily on Linux OS. All the applications which run in Dalvik VM has a special format ".dex". Dalvik is a distinctive feature of Android and a very crucial component in Mobile Forensics domain.

Application Framework

It is written in java, and is used by all the applications. And since all the applications use same framework and APIs, it is easy to write applications. All the components has their different functionalities like, the package manager keeps the record of all the current applications in the device and updates its database when any new application is installed. Window manager manages Windows user interface. Activity manager handles the application life cycle.

Applications

All the components of the Application field are written in java, they are multi tasking and run its own processes. Android offers a variety of APIs giving the ability to developers to build innovative applications.

Table 1.2 gives the complete list of Android platforms, API level, and release date as of September 4, 2012.

Table 1.2

Version	Release Date	API Level
4.1.x Jelly Bean	July 9, 2012	16
4.0.x Ice Cream Sandwich	October 19, 2011	14-15
3.x.x Honeycomb	February 22, 2011	11-13
2.3.x Gingerbread	December 6, 2010	9-10
2.2 Froyo	May 20, 2010	8
2.0,2.1 Eclair	October 26, 2009	7
1.6 Donut	September 15, 2009	4
1.5 Cupcake	April 30, 2009	3

Having knowledge about the version of Android platform assists in determining the features of the device.
All the platforms are given API levels, and all the new versions are specified with code names. The latest code name of the Android version is Jelly Bean. When looked at large number of Android versions present, it is easily construed that it has a major impact in the field of mobile forensics and security analysis. Also there are many devices, which do not support the latest version of android, and some do get an update, in future many devices will be able to support and upgrade to the latest version. Nevertheless from a forensic viewpoint the old outliers can't be ignored.

1.2 Android Platform Highlights of the Version used in the tests for this project: *Androids 2.0 & 2.1 (ÉCLAIR)* [40]

Released October 2009 and January 2010, Table 1.3 Lists all the features of these versions.

Table 1.3 – Highlights and Features of Android version 2.0, 2.1

Features new for user	New features for Developer, APIs	Built-in Applications
• Multiple email account feature, sync functionality for contacts. • SMS/MMS search facility • New and improved Android Virtual Keyboard • Built-in flash, digital zoom for camera • HTML5 support, webpage thumbnails in bookmarks • Calendar feature for inviting guests.	• Bluetooth 2.1 • API for live wallpapers • Improved hardware acceleration using revamped graphics architecture.	• Same as Android 1.6

1.3 Software Development Kit (SDK) [4]:

Android Software Development Kit (SDK) is a very dominant tool used to assist forensic experts during their investigation. It is the core resource used in developing android applications. Android SDK comes with ADT plugin for eclipse, Platform-tools, APIs, Emulator, software libraries, etc. It is free, open-source, and supports all the major OS platforms. SDK can be downloaded free from [4].

1.3.1. SDK Install

Since SDK is the major component used during the investigation of an Android device, it is crucial to get the working knowledge of its installation and use. The following section details about the steps used for its installation.

Linux SDK Installation: - The steps are for installation on Ubuntu VM

SDK package downloaded from [4] is a .tgz. Unpack it to a safe location and it gets unpacked to a default name android-sdk-linux_x86.

1. If the version of Ubuntu used is 64-bit then the additional libraries from ubuntu 32-bit is required for SDK installation. The following command is used for its installation:

```
apt-get install ia32-libs
```

2. Next step is to install Java:

```
apt-get install sun-java6-jdk
```

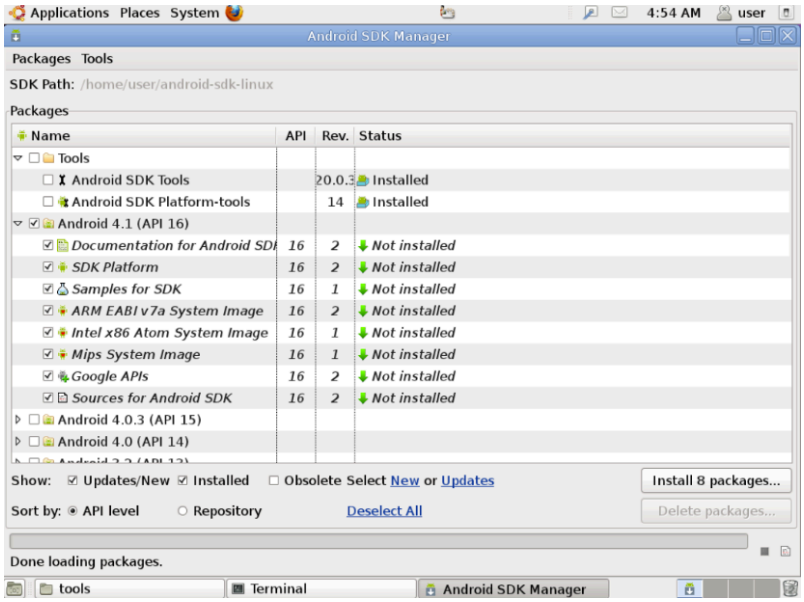
3. The Package manager of Ubuntu doesn't offer the current version of eclipse 3.6, It can be downloaded from http://www.eclipse.org/downloads/

4. From the Terminal window navigate to the tools directory of the package unpacked and run the following command:

```
# run android
./android
```

5. This command will run the Android SDK manager through which we can easily download the additional components and install the packages needed to run the emulator and SDK. Figure 1.2 shows the step 5 to install the additional packages using Android SDK manager.

Figure 1.2 Android SDK Manager



1.3.2. Connecting Android Device to the Workspace

It is important to know how to let device interact with the system. Currently almost all devices support USB interface to share files, resources and mostly to charge the battery. When working with VM, we have to select whether we want to connect the device to the host OS or the VM. On connection the device points to more than one virtual USB interface. For example when connected with Samsung Galaxy S3 (v4.0.4) with USB the following options were offered.

1. Charge only
2. Connect as Media Device (MTP) – Allows to transfer media files in windows and Android file transfer in Mac.
3. Camera (PTP) – Allows transferring photos using camera software and transfer files that were not supported by MTP.
4. Debug Mode when USB is connected – Used for Forensics and Development purposes.

1.3.3. USB Debugging:

As mentioned above “Debug mode when USB is connected” is an USB interface used to uncover the ADB (Android debug bridge), giving the forensic analysts power to interact with the Android device using USB. This mode is not prompted on the screen of the device when USB is connected;

it has to be explicitly enabled. In order to enable go to: Settings → Applications → Development → Enable USB debugging. Figure 1.3 shows an example of this mode.

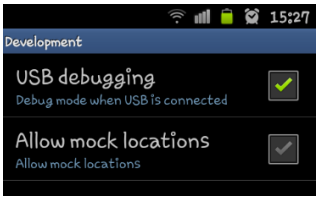


Figure 1.3: Debug mode when USB is connected.

When the Debugging mode is checked the Android device can then run the ADB daemon (adbd). This daemon will limit the access it has to the data by running under the public shell user account. On contrary the devices, which have root access, can run this daemon as a root and then can have broad access to the system. For every existing Android forensic tool this USB debugging must be checked on the device.

1.4. Android Debug Bridge [4]

Android Debug Bridge (ADB) is a command line tool used to interact with the Android device. It is a client-server program consisting of the following components when it is running.

1) A client running on development machine, which can be invoked from shell using ADB command.
2) Server, running as background process on the system, it manages the activities between client and ADB daemon.
3) Daemon running as a background process on device.

"The ADB tool can be located in the platform-tools directory of the Android SDK package"

When the ADB client is started, it checks for any server process running. If not, then it initiates the server process. On starting, server binds to local TCP port 5037 and listens the commands sent via ADB client. It then configures the link to all the running device instances. Every device instance gets a pair of sequential ports → Even Ports for Console connections and Odd Ports for ADB connections. Server locates the device instances by scanning odd numbered ports in the range 5555 to 5585. Once all the connections to instances are configured ADB commands can be used to work with those instances.

The ADB commands can be used from the terminal of the Ubuntu VM, the following detail its usage:

```
adb [-d | -e | -s < serial number > ] < command >
```

ADB client is invoked every time when a command is issued. '- d' is used to indicate the target instance to which the command will be issued.

The commands discussed below will be used throughout the experiments done using adb tool.

I. Checking for devices attached to the machine:

The 'devices' command gives a list of devices currently attached to the development machine. The following details appear on the screen when the command " adb devices " is issued on the terminal:

a) Serial number: To identify a device instance by its console port number.

```
Format → <type>-<consolePort>
Example → 4df14c30035e4f0f   device
```

b) State: Instance can have the following connection states
Offline: instance not connected to adb or not responding
Device: instance is connected to adb
No device: device is not connected.

Example of the 'devices' command output on terminal:

```
root@ubuntu:$  adb devices
List of devices attached
4df14c30035e4f0f  device
```

II. Installing Applications

ADB can be used to copy applications from the system and install it to the Android device. The 'install' command is used for this purpose, and is given with the path to that .apk file which has to be installed.
Command:

```
adb install <path_to_apk>
```

III. Copying files from and to the Device

Commands push and pull are used for copying files to and from the device. It can copy any kind of file to any location to the device.

Copying file from device use:

```
adb pull <remote> <local>
```

Copying files to device use:

```
adb push <local> <remote>
```

Commands local and remote refer to paths to target file on machine (local) and device (remote).

Example:

```
adb  push  report.pdf  /sdcard/report.pdf
adb  pull  /sdcard/report.pdf  /home/user/Desktop
```

IV. Server – start and kill commands:

`start-server` → initiates the adb server process if its not running already.
`kill-server` → dismisses the adb server process.

V. Shell commands:

"Shell command can be used to issue commands without the need to enter adb remote shell on the device." In other words it opens a shell on Android device and allows communication with the machine

`adb shell` → to enter into the remote shell of the device.

The following appears when this command is issued on the terminal of Ubuntu VM:

```
root@ubuntu:$./adb shell
shell@android:/$
```

1.5. Rooting Android OS [5]

Rooting Android OS, also called root access is a method to acquire privileged control of the Android's system. The phone needs to be rooted in order to map the memory into an image file system, and also some the logical forensic tools work only on rooted android devices. For example, powerful forensic tool AFLogical maps the memory image directly from the device but under the condition that the device must be rooted. Once the device gets rooted the limitations put by the device manufacturer or the carriers gets dazed. Administer-level permissions are allowed to the user, and hence giving the ability to alter the system applications and settings. Once rooted, the complete replacement of the operating system is facilitated. There are a plethora of applications available to root the device. The application "Super One-Click" was used for rooting the device on which experiments were conducted for this report. It can be downloaded from www.shortfuse.org. This application requires Microsoft .NET framework 2.0+ and runs only on windows and Ubuntu machine.

The following steps are used to root an Android device:

1) Connect the device using USB to the development machine
2) Enable USB debugging mode and unmount SD card
3) Download Super One-Click and SDK should be running
4) Unzip the downloaded file, and launch the application
5) A screen as shown in figure 1.4 will be presented, click "root"
6) The running process will be displayed with a OK appearing after finishing every task
7) Once finished, check the apps menu of the device, a super user icon will be present.

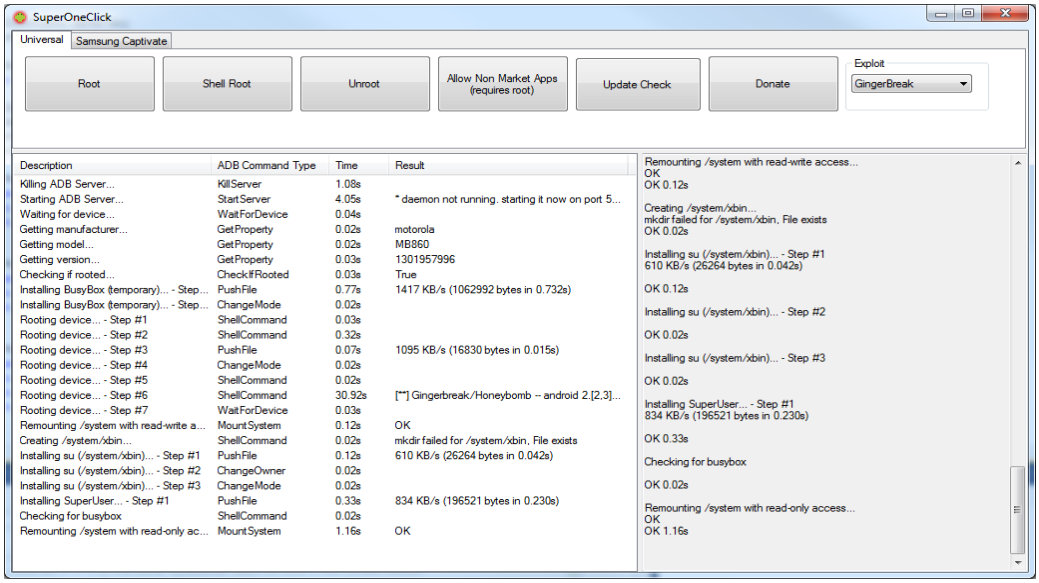


Figure 1.4 [6]: Rooting Android device using Super One-Click.

Settings can be done to avoid any application run with root access by configuring the notifications bar in the superuser. Once done, every time an application needs root access it will ask for permission first. One of the major risks involved during rooting the Android device is the danger of "bricking" the device; this means if the phone gets bricked it remains nothing but the piece of metal and glass. Super One-Click is a tested application and can be used safely on majority of Android devices.

1.6. Android File Systems [1]

Knowledge about the data stored on the Android device is required in order to understand and analyze the data acquired through an investigation. For an effective analysis, factors such as types of data stored, how they are stored, and features of physical medium on which the data is stored, play a vital role.

Data in the shell

Android is a permutation two of well-known artifacts, ones found in Linux and new ones such as Dalvik VM and YAFFS2 file system. There is huge amount of data stored on device and Applications are the primary source of this data. The list below gives the sample of the data found on a typical Android device.

i. SMS/MMS
ii. Contacts & Call history
iii. Email messages
iv. GPS coordinates
v. Pictures/Video
vi. Social media information
vii. Business information
viii. Bluetooth file sharing
ix. Web history, etc.

1.6.1. Application data storage [1]

The App data can be stored on either of SD card or internal memory of the device. The data stored in internal memory gets saved in a subdirectory of /data/data named after the package name. For example android Bluetooth has package name com.android.bluetooth, and the files are saved in /data/data/com.android.bluetooth.

Table 1.4 List of important subdirectories.

Table 1.4	/data/data/<packageName>Subdirectories
shared_prefs	Directory saving shared preferences
databases	Sqlite databases
lib	Library files required by application
cache	Cache files from web browser
files	Files saved to internal memory

There are generally six types of partitions found on Android device, cache, recovery, boot, and user data. Table 1.5 [7] presents a list of such typical scheme:

Table 1.5 [7] Partitions in Android Device

Path	Name	File System	Mount Point	Description
/dev/mtd/mtd0	pds	yaffs2	/config	Configuration data
/dev/mtd/mtd1	misc	-	N/A	Memory partitioning data
/dev/mtd/mtd2	boot	bootimg	N/A	Typical boot
/dev/mtd/mtd3	recovery	bootimg	N/A	Recovery mode
/dev/mtd/mtd4	system	yaffs2	/system	System files, applications
/dev/mtd/mtd5	cache	yaffs2	/cache	Cache files
/dev/mtd/mtd6	user data	yaffs2	/data	User data (Apps)
/dev/mtd/mtd7	kpanic	-	N/A	Crash log

1.6.2. SD cards [1] (Removable Media)

All the user files, such as multimedia files are places in SD card. The critical data remains on the phone memory of the device, but SD card also proves to be a major source of information during an investigation. In Ubuntu, SD card gets automatically mounted unless auto –mounting of USB devices is not disabled. The following Linux command can be used to check where the SD card is mounted on the system:

```
root@ubuntu:platform-tools tanujapanwar$ df -h
Filesystem      Size    Used    Avail   Capacity  Mounted on
/dev/disk0s2    297Gi   156Gi   141Gi   53%       /
devfs           186Ki   186Ki   0Bi     100%      /dev
map -hosts      0Bi     0Bi     0Bi     100%      /net
map auto_home   0Bi     0Bi     0Bi     100%      /home
/dev/disk1      186Mi   50Mi    136Mi    27%       /Volumes/Untitled
/dev/disk2      186Mi    59Mi    127Mi    32z      /Volumes/Untitled 1
/dev/sdb1       8Gi     2Gi     6Gi     75%       /media/Music
```

df command: It reports the file system disk usage. From the output on the left, the SD card was mounted on /media/Music. The "-h" is for printing the sizes in human readable format [3].

And, on the Android device the SD card is mounted as:

```
/dev/fuse /mnt/sdcard fuse  rw,nosuid,nodev,relatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
```

1.7. Forensic Tools

Mobile forensics is a new area in the field of digital forensics and the tools used for data extraction are rapidly developing. A plethora of logical forensic tools are available in the market, amongst which majority are commercial tools and a small number are open source tools. Some of the commercial tools, mainly hardware tools are very expensive and requires license to operate them. In addition to that various security factors also matters in using these tools. On contrary to that open source tools are free for use, does not any licensing and security issues does not matter much. When compared, commercial and open source tools, commercial tools are more reliable and accurate in device acquisition. Open source tools come with limitations and may contain faults in the data acquired.

The forensic tools used for this project are:

I. **Autopsy (Sleuth-kit)**
II. **Scalpel (Foremost)**
III. **AFLogical (Viaforensics)**
IV. **ADB (Android SDK)**

1.7.1. Autopsy [8]

It is a Graphical user interface to the powerful forensic tool Sleuth kit. It can analyze Windows, Unix, file systems such as NTFS, FAT, EXT2/3. Version 3.0 of the Autopsy was used. All the results are stored in embedded SQlite3 database. Some of its features are – Hash value calculation, keyword search, user activity (web search), recovery of deleted files etc.

1.7.2. Scalpel [13]

It is a file-carving tool, basically used to recover deleted files. Scalpel has originated from Forensic tool 'Foremost' and is a far more efficient that its predecessor. It reads the database of header and footer definitions and extracts matching files from the image (.img) file partition that is being analyzed. Version 1.0 of Ubuntu was used for the experiments. It can recover files such as jpeg, pdf, mpeg, email etc. More features can be added by amending the scalpel.conf file by giving the header and footer of file to be carved.

1.7.3. AFLogical [17]

AFLogical OSE is an open source tool provided by viaForensics. It can be used as a command line tool or can be installed on device using adb functionality. The extracted data is stored in .csv format. Version 1.5.2 was used for experiments.

1.7.4. ADB (Android Debug Bridge) [4]

It is a command line tool used to interact with the Android device. In order to use this tool android SDK has to be installed on the development machine. adb tool is located in the platform-tools folder of the SDK package. Typical features include copying files, installing apps, forwarding ports.

1.8. Methodology of Experiments

Experiments were conducted using two test cases; data was recorded manually for comparison with the acquired data using the tools. The result of this comparison was recorded and several tests were conducted to validate the results of the tools. Evaluation of tools was based on extraction of files, recovery of deleted files, information based on test cases, CPU usage, RAM usage, load average on CPU when the tool was working etc. Figure 1.5 depicts the methodology used for tool assessment.

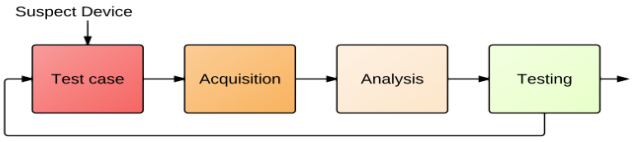


Figure 1.5: Tool assessment

1.9 Summary

A review on Android platform and Android architecture was done to get an understanding of logical forensic analysis. A list of Android platforms released till-date was discussed. Highlighted the features of version of Android platform used for the experiments in this report. Android SDK, its installation are reviewed. Procedures used to support the forensic process, like USB debugging, Android rooting etc. A brief discussion was done about the features of powerful tool android debug bridge. A brief idea about the mobile file system and application data storage on device is discussed. And finally the forensic tools used in this research project are outlined.

Device Specification & Test Cases 2

2.0 Introduction

A meaningful evaluation framework is needed for successfully completing the project. Prior to that, knowledge about the device and Development machine used for the tests is crucial. It forms the basis of the test. Test case methodology to implement the tools and data collection techniques frames the experiments.

2.1 Device Specification and Development machine

The Android device used for the experiments was Samsung Galaxy Europa and Samsung Galaxy S3. The devices were selected in such a manner that they cover the different versions of Android operating system. Both the devices were tested based on two cases in order to investigate information related to the use Android Smartphones.

Table 2.1 [9] Specification of devices used.

Specifications	Samsung Galaxy Europa	Samsung Galaxy S3
CPU	600 MHz	1.4GHz
Android OS	Version 2.1 (Éclair)	Version 4.0 (ICS)
Internal Memory	170MB	16GB
WLAN	Wi-Fi 802.11 b/g/n, DLNA, Wi-Fi hotspot	Wi-Fi Direct, DLNA, MHL 1.0, Wi-Fi hotspot
Bluetooth	V2.1 with A2DP	V4.0 with A2DP, EDR

Development Machine

The development machine used for the experiments was running under Ubuntu 64-bit and OSAF VM [10]. Osaf VM is an open source compilation of Forensics and malware software in the form of a Toolkit. A variety of forensic tools other than the once used for Android smartphones were pre-loaded onto the Osaf VM and can be used directly from there. A limitation to this toolkit is that, the forensic tools loaded onto the system were the most basic versions of the toolsets available, and cannot be updated manually unless there is a new version of this toolkit available in the market. The four tools were tested on both the machines to cover all the features from older and newer versions of the tools.

2.2. Test Cases and Methodology

Two test cases were used for investigation purpose. The data was recorded manually during the tests, the recorded data was then compared with the data acquired using the Forensic tools. The result of that comparison was analyzed by performing validation tests and the outcomes of those

tests are presented as a visual representation. Following gives the description of both the test cases and the environment that needs to be setup for performing the experiments.

2.2.1 Test case I

Piracy

Before the release of the music album, the criminal, using the Android smartphone recorded the music video in 3GPP format. Shared the recorded file using Bluetooth to three different devices. Using the wireless network of the local area and using the Internet by service provider (3 Mobile) uploaded the file on Facebook, YouTube, and using Gmail emailed the file. Then deleted the video.

2.2.1.1 Methodology used to implement the test case

Following steps were performed to complete the test case scenario.

a. Get the device
b. Record the music video and note the format, name, size, and duration of Video.
c. Connect to the local wireless network of the area
d. Login to Facebook, YouTube, and Gmail
e. Upload the files using both wireless network and service provider's network
f. Transfer files using Bluetooth to three devices
g. Delete the video
h. Record the timestamp for all the above steps
i. Wait for a period of time, the device was kept for 12 hours before extracting the data.
j. Acquire the data.

Test-Case Prospects

- File recorded – 3GPP format media file
- Wireless network – hotspot wireless network created by Android device Samsung Galaxy S2
- Bluetooth – Files shared with three different devices - MacBook Pro, Nokia E71, and Samsung i5700.
- Native Facebook, YouTube, and Gmail applications were used.

2.2.2 Test case II

Exam info

A student copies files from the teacher's computer. The files were the question papers and answers for the upcoming exam. Using the wireless network of the local area emails the file, uploads it to Dropbox, and share it using Bluetooth. After all the transfers are done, student deletes the file.

2.2.2.1 Methodology used to implement the test case

Following steps were performed to complete the test case scenario.

a. Get the device
b. Copy the files from the system
c. Connect to the local wireless network of the area
d. Login to Dropbox
e. Upload the files using both wireless network and service provider's network

f. Transfer files using Bluetooth to three devices
g. Delete the files
h. Record the timestamp for all the above steps
i. Wait for a period of time, the device was kept for 12 hours before extracting the data.
j. Acquire the data.

Test-Case Prospects

- Files Copied – 3 Docx files, 2 Pdf files, 2 java files, 8 jpeg files and 1 zip file
- Wireless network – Local wireless network
- Bluetooth – Files shared with two different devices - MacBook Pro, Samsung Galaxy S3
- Native Dropbox and Gmail applications were used.

2.3. Data Collection [2]

Data collection is a principal step of a research. When the tools are tested and analyzed a large amount of data is produced. This data is analyzed for further advancement in the investigation. Mainly there are three steps providing data for this research.

- ✓ The data recorded manually was referred to as baseline data. This baseline data was used for comparison with the artifacts acquired using the tools.

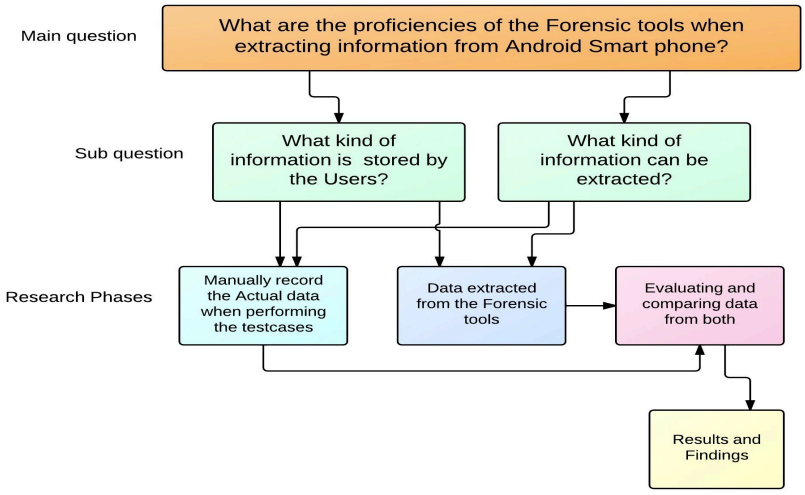
- ✓ The data which was extracted using the Forensic tools, this data provides evidential proof value and needs to be evaluated and analyzed.

- ✓ The third and final data is the result of comparison of the acquired data with the baseline data counted manually.

Figure 2.1. Represents the Investigation approach used for the Data Collection and analysis procedure.

[Note: This map focuses on the steps involved in Data collection only]

Figure 2.1. Approach for Investigation



The final step in data analysis is the visual representation of the statistics acquired using the above steps. The visual representation of the data will be in the form of tables and graphs for a clear comparison of the performance of the tools.

2.4. Summary

Both devices used for the experiments and the development machine under which all the experiments were run was stated. The test cases, their methodology to implement and the scheme of data collection were sketched. Finally the approach used for investigation was presented to depict the phases employed in data collection process.

Forensic Tools and Implementation 3

3.0 Introduction [32] [38]

Forensic tools provide excellent techniques to extract forensically sound information from the Android devices. A variety of Forensic toolkits are available in the market. Depending upon the types of devices they support, the selection of the Forensic tools narrows down to, the range of devices, version of Android OS, hardware, etc. Information from the Android device also depends on various other factors like phone's inbuilt features set by the manufacturer, modifications by service provider, and changes made to the device by the user.

A tool, which does not comply with forensic needs, should be thoroughly tested before use. Integrity is questionable in such types of tools because they do not compute hashes of the acquired content. In addition to that, documentation and source code are limited or not available for evaluation. This may results in miscalculation of the data acquired and off target statistics.

The mounting growth of Forensic Sciences demands more improved Forensic Tools with enhanced capabilities and a means to verify these toolsets. A comparative environment is setup under which toolsets were evaluated based on their performance on different devices and evidence extracted. Test cases are generated based on these specifications and tested on the target tools. Process of analysis should be the operational focus of the digital forensics domain. For example, when working with a problem related in analyzing video obtained from a new device, main objective should be the impact on the digital forensic as a scientific discipline rather than designing a tool focused only on that device.

Tools should be able to identify all the non-supported Android devices. Connectivity errors caused during acquisition should be identified and reported. Tool should provide all the information logically and without any modifications with the results documented properly. If the tool provides a defined capability then it is tested for the corresponding requirements for example hashing, file creation, physical acquisition, etc. The documentation of the acquired information should be presented in a human readable format, which is easily accessible by the user.

3.1 Techniques for data extraction [11]

Forensic Tools can acquire data in two ways, logical acquisition and physical acquisition. This project is based on data acquired from the Android smartphones using Logical acquisition. The following accord a brief outline about the difference between the both techniques. The comparison is made in order to delineate the choice of Logical techniques.

3.1.1 Logical Techniques

When seen from forensic perspective, Logical technique is the most recommended approach for the data extraction. A logical technique focuses on extraction of the data by accessing the file system. The file system works in a two-way fashion. It checks for the allocated data. If the data is allocated then it is accessible on the file system. And when the data is deleted, it actually gets unlinked from the file system. The data is still there on the device, just the file system is being told that there is no data allocated and it can write over that space. But if the data is not over written by the file system, it can then be easily carved out. Also the files such as SQLite database still contains the record of deleted files and can be accessed form the databases.

3.1.2 Physical Techniques [2]

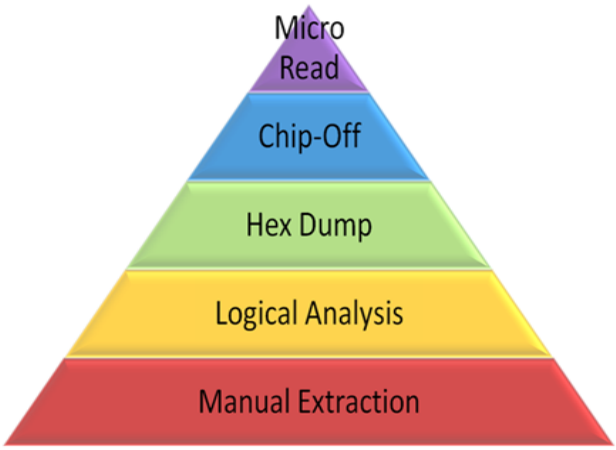
Such techniques target the physical storage medium of the device. It does not concern with the file system. It offers the access to deleted data, both allocated and unallocated. Based on the figure 3.1 the physical extraction methods consist of namely three layers; Hex Dump, chip-off, Micro Read.

- Hex Dumping of an Android device can be seen as booting a computer from CD and generating an image of the hard disk. Since this requires code level access of the OS, it needs a very skilled professional for this work.

- Next is Chip-off, this method deals with removing NAND flash physically and analyzing it manually. This method is a destructive way of acquiring data, and is used as a last option when nothing else could be done with the device.

- Micro Read is very costly to perform and is not used as a standard method for data extraction.

The following figure 3.1 [12] shows a pyramid for extraction techniques.



3.2 Forensic Tools

Following four Forensic tools were used for the experiments. All these tools were tested on the both Android devices as stated in chapter 2, table 2.1. Test cases discussed in section 2.2 of the report were used for the experiments.

- ADB (Android Debug Bridge)
- Autopsy (Sleuth-kit)
- Scalpel (Foremost)
- AFLogical (viaForensics)

First the implementation of ADB is completed, as the tools Autopsy and Scalpel are based on the image file system generated from the Android device's phone memory using ADB. Lastly AFLogical is implemented; it does not depend on the image file system of the device. But ADB is used for the installation of AFLogical application on the device. Figure 3.2 explains the dependencies of the other three tools on ADB.

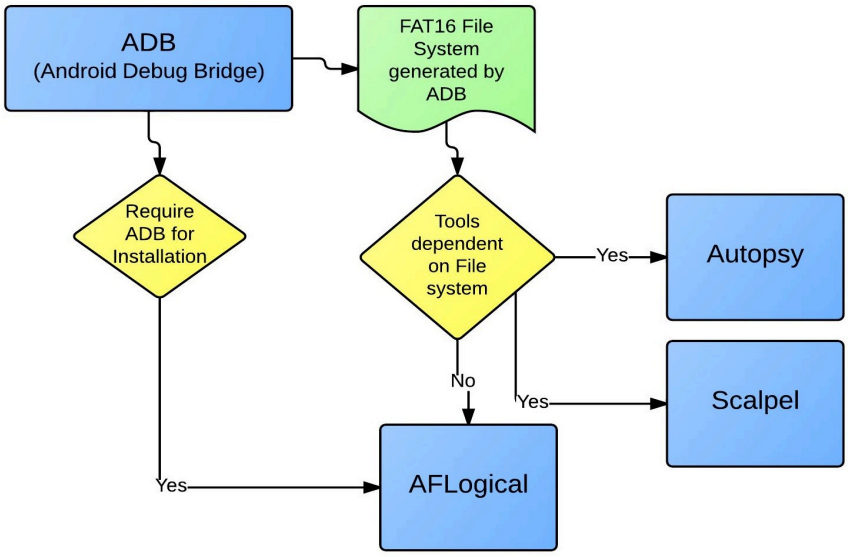


Figure 3.2. Tool Dependencies

3.2.1 ADB (Android Debug Bridge) implementation [9]

Android Debug Bridge (ADB) is a command line tool used to interact with the Android device. All the background features of ADB are discussed in section 1.4. This section deals with the implementation of ADB using the two test cases discussed in section 2.2.
The flowchart 3.3 portrays the steps comprising in the generation of the FAT16 image file system partition, this image file system is generated from phone memory of the Android Device.

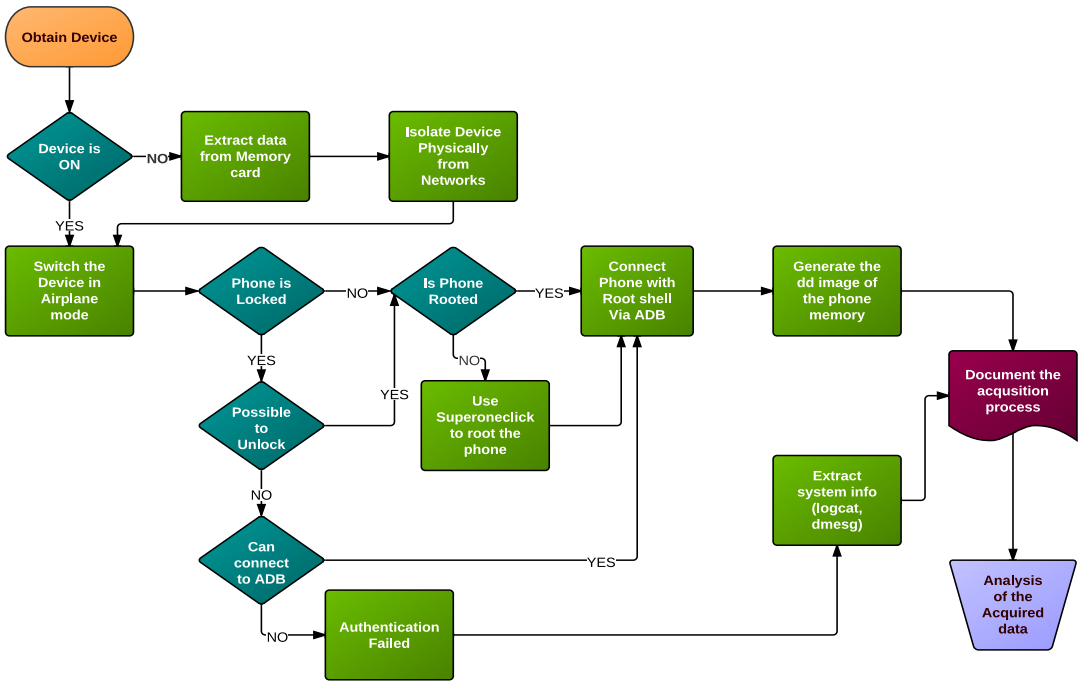


Figure 3.3 Implementation of ADB

Prior to the investigation process the Forensic examiner should read the documentation of ADB tool, which is available at [4]. The commands used for the extraction process is detailed in section 1.4 of the report.
The following steps give the broad idea about the experiment performed [9].

1. Use of an SD card is crucial; the SD card used for the experiments should be formatted and forensically wiped. Also the capacity of the storage card must be larger than the phone's internal memory.

2. Document all the actions taken during the investigation. Record the date and time of each and every step performed.

3. Connect the Android device using USB cable to the development machine and USB debugging (Section 1.3.3) option should be checked ON.

4. Image file partition generation via ADB is successful only on rooted Android devices, check for device is rooted or not. If not rooted then follow the steps detailed in section 1.5 of the report.

5. Once the phone is rooted, check for the relevant partitions of the device memory for investigation using Section 1.6.2

6. Activate the 'Airplane Mode'; this deactivates all the networks and wireless connections of the device.

7. Check whether the device has any SD card.

8. Start the ADB server by using the command
   `sudo adb start-server`

9. Dump the logs of Android device using *logcat* tool, command to be used is:
   `adb logcat -d -v -time -b main > file`

10. Generate a checksum for all the files by using *md5sum* utility of the development machine.

11. Kill the ADB server by using the command
    `sudo adb kill-server`

12. Switch the power off of the device and remove the SD card, it can then be sent to forensic labs for further investigation.

13. Insert new forensically cleaned SD card and mount the card by booting up the device.

14. Repeat steps 3-8 in order to start the daemon, type the command to enter the adb shell
    `./adb shell`

15. Then use command `'su'` to gain the root access of the machine.

16. Type `mount` to check for the relevant file system partitions to be imaged.

17. Using dd command for generating the image of the data partition, this image will be then stored in SD card using adb.

    ```
    dd if = /dev/stl13 of=/sdcard/name.img bs=4096
    ```

    Here, /dev/stl13 is the partition which was imaged from the Android device, this was the partition related to the stored user data in the device. /sdcard is the path of the Android device's SDcard location where the image file name.img will be stored. The image file generated is FAT16 type file system.

18. Then type `exit` twice.

19. Next step is to pull the generated image file form the device to the development machine. For doing this using the following command
    `./adb pull /sdcard/name.img /path`
    `path` represents the relevant path where the image file will be saved.

20. Stop the server by using command
    `sudo adb kill-server`

21. Unplug the cable between the device and the development machine.

22. Using the mount command, mount the image file onto the development machine.
    Goto the location where image file was saved, from that location, type mount command, path represents the location where the file system will be mounted.
    `sudo mount -o loop name.img /path`

23. Analyze all the files acquired, unmount the filesystem by using: `sudo umount path`

The file system has to be mounted every time it needs to be accessed. The figure 3.4 lists the features of ADB till this point to study, and the information found on accessing the mounted file system. This information is based on the test cases and only the databases related to the relevant files were browsed.

Figure 3.4 ADB – Features catalogued

ADB
➢ Interact with the Android Device
➢ Android Rooting
➢ Installing applications
➢ Map the device memory with DD command
➢ Bypass phone lock/pattern
➢ File recovery
➢ Attributes of the files mapped into image partition
➢ Call history, SMS/MMS
➢ Databases store Application data
➢ Bluetooth usage details
➢ Wi-Fi access information
➢ Network Provider used by the Android device
➢ Recovery of deleted files
➢ Command line tool
➢ Open source tool available in Android SDK

The information as detailed in figure 3.4 relates to the actual details about the files and activities encountered. In order to get the details about the information related to email, YouTube, Wi-Fi, Bluetooth etc., A forensic examiner is advised to analyze the files and folders represented in table 3.1.[9]

Table 3.1. Sources of Evidence

Path	Filename
/data/com.android.bluetooth/databases	btopp.db
/data/com.android.email/databases	emailProvider.db
/misc/bluetoothd/MAC	Classes, config, lastseen, linkkeys, names,profiles
/data/com.dropbox.android/databases	db.db
/data/com.facebook.katana/databases	fb.db, webview.db, uploadmanager.db
/data/com.google.android.youtube/shared_prefs	Com.google.android.youtube.credentials.xml
/misc/wifi	wpa_supplicant.conf
/data/com.google.android.location/files	cache.wifi, cache.cell
Logcat files	Main and events buffer
/data/com.android.broswer/databases	browser.db, webview.db,webviewcache.db

The paths given in the table 3.1 gives information about the databases, which stores the information relating to the particular file. These databases can be browsed using SQLite3 tool.

3.2.2 Autopsy [8]

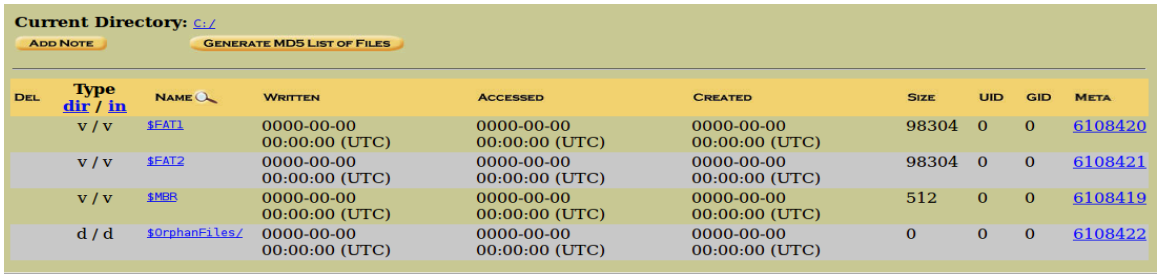
Autopsy is GUI forensic browser originated from the command line forensic tool 'The Sleuth kit'. It supports file systems such as FAT, Ext2/3, NTFS, Windows and Unix disks. Autopsy is Open source tool and runs on Unix platforms. It is HTML-based; on passing the command `sudo autopsy` in the Ubuntu terminal connection to Autopsy's server can be made. A URL appears in the terminal screen, which can be pasted to any HTML browser. It can give access to deleted files and file structures etc.
The image file system generated using ADB was analyzed using Autopsy. All the figures used in this section have been obtained by analyzing the image file for both the test cases using Autopsy.

Evidence frisking techniques of Autopsy

- ***File Catalogue***

  Shows all the files and directories, the catalogue includes the information about the deleted files as well. Refer figure 3.5

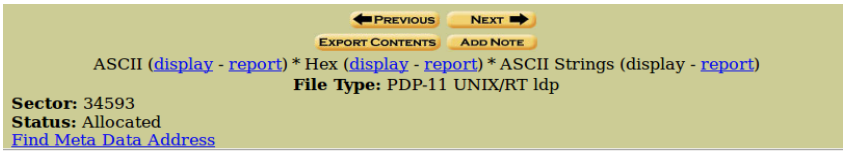
Figure 3.5 Autopsy File catalogue



- ***File Content***

  The information in the files is represented in three formats, raw, hex and ASCII strings. It does not use any client-side scripting languages. Refer figure 3.6.

Figure 3.6 Autopsy File content



- ***Hash databases***

  Anonymous files can be searched using the hash databases to mark them as useful for investigation or not.

- ***Timeline of File occurrences***

  Autopsy can create timeline of the files based on their access times and modification times. These files can be either of allocated or unallocated files.

- ***Keyword Search***

  File system can be searched using ASCII strings. Keyword search can be done on both the file system and unallocated space. Following screen shot gives an example. Refer figure 3.7 and figure 3.8

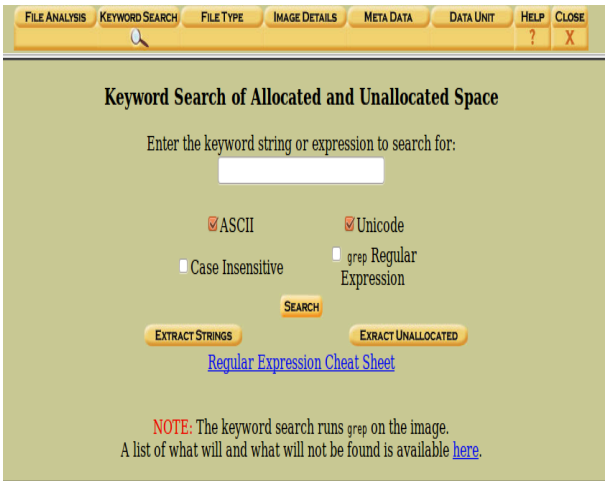


Figure 3.7 Autopsy – Keyword search

From figure 3.7, the keyword search functionality of Autopsy, allows searching of files using grep regular expressions, ASCII formats.

Figure 3.8 Autopsy – Keyword search result in hex

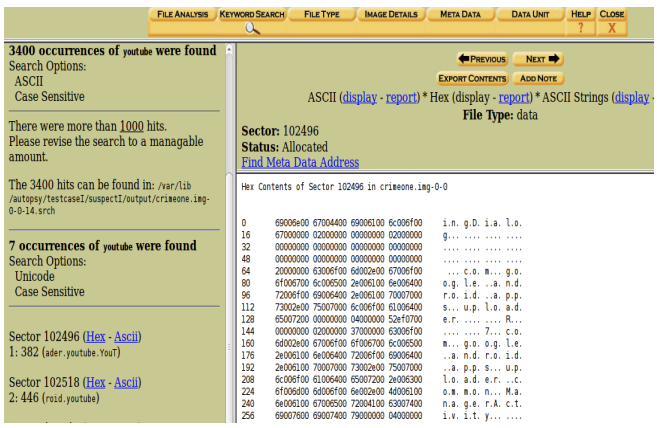


Figure 3.8 shows the output of searching a file named Androidvideo and it gives the details about all the occurrences of the file name found, with the sector number and status of the file- saying it is allocated. Metadata address can also be found.

- ***Metadata Analysis***

  Details about directories and files can be found in Meta data structures. Metadata information is useful for recovery of deleted files. Refer figure 3.9

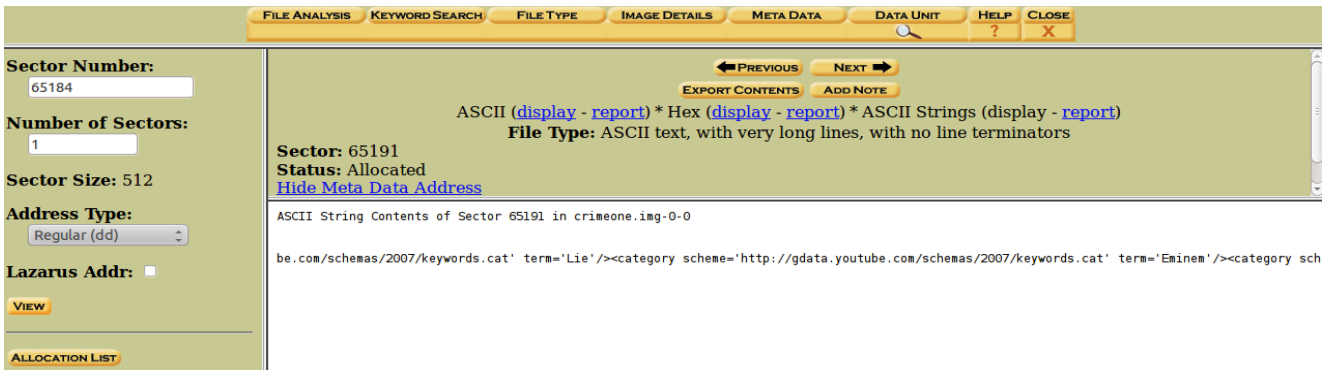
Figure 3.9. Autopsy Metadata analysis

Figure 3.9 Autopsy's Metadata analysis gives the information about the file type, MD5 of the content, SHA-1 of the content, status of the files, size of the files, timestamp, and the sectors related to the searched file. It also gives an option to export all the details into a text format.

- ***DataUnit analysis***

  It gives the location to where the file content is stored. The data can be viewed in all the three formats as discussed in figure 3.6. Figure 3.10 gives the screenshot of the Dataunit analysis of the file system used for this project.

Figure 3.10 Autopsy DataUnit analysis



From figure 3.10, DataUnit analysis of Autopsy provides the contents of the files. Files can also be browsed by providing the sector number and displayed on ASCII strings format. The file being browsed in this figure shows the details about the video uploaded on YouTube.

- ***Image details***

  All the details about the file system including system type, volume ID, file system layout, information about sectors, metadata information, content information can be found. Refer figure 3.11.

Figure 3.11 Autopsy image Details

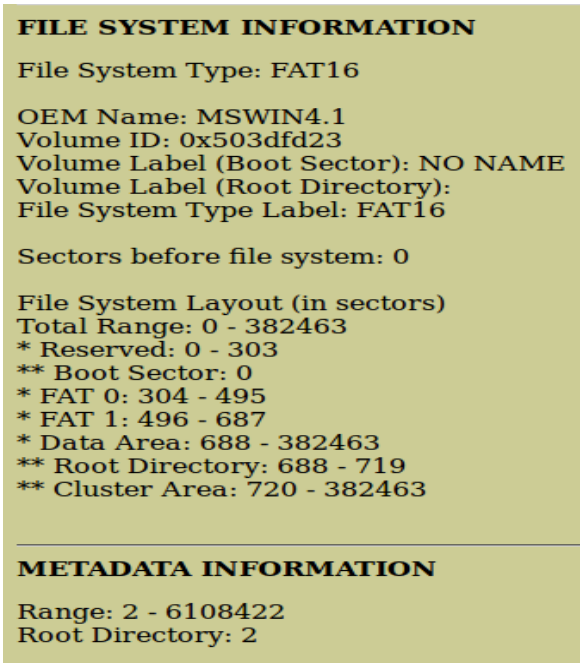


Figure 3.11 depicts the details related to the image file system being analyzed. File system information, Metadata information and Content Information were detected.
Content Information was: -

- Sector Size: 512
- Cluster Size: 4096
- Total Cluster Range: 2 - 47719

3.2.2.1. Investigation Phase [8]

In autopsy, investigation process is structured in cases. Every case contains its host with the time zone setting. File system is contained in the host. Image file connected to the host retains its integrity during the analysis. Autopsy calculates a MD5 value for all the files in the host.
The following steps were carried out to analyze the file system generated by ADB for both the test cases 2.2.1 and 2.2.2.

I. Open the terminal window of Ubuntu and type the command `sudo autopsy`. A url http:// localhost:9999/autopsy will appear on the terminal window, copy it and paste it into any browser. This will direct to the GUI page of Autopsy. Do not close the terminal window.

II. From there, click the button ‘new case’, and enter the fields saying case name, description, investigator and click new case.

III. A new case will be created with a case directory having path /var/lib/autopsy/report/, click ‘Add host’ and enter the fields saying hostname (name of the system being investigated), description, timezone and click add host. The host directory will be /var/lib/autopsy/report/host-name. Then click ‘Add image’.

IV. Following fields will be presented after step III:

- Fields asking for location - enter the path of the image file where it is saved into the system
- Type - select the image file for disk or partition, the image file used for experiment is partition.
- Import method - select symlink.

In order to analyze the image file it needs to be stored in the evidence locker. Three options are available, symlink, copy, move. The best way to link the image file to the evidence locker is symlink. Symlink creates a link between the image file and the evidence locker. On moving the image file using the move option, if an error occurs the image could get permanently corrupt.

V. For integrity verification purposes Autopsy calculates MD5 hash of the image files analyzed. The MD5 hash calculated for the image file generated for both the test cases are:

- Testcase I (2.2.1): `MD5 - 19D948B89D077B166B7ADFDBC7EAD112`
- Testcase II (2.2.2): `MD5 - D7A790CD104915B5AC4A403AA5AA309D`

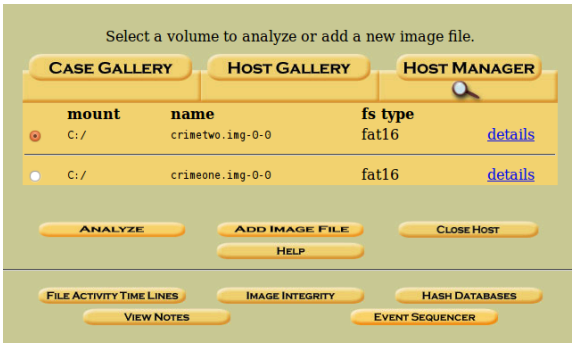


Figure 3.12:
Image files for both the testcases are attached to the evidence locker. Various options are available to handle the image files. File activity timeline for tracking the timestamp of the files in the image. Click the button 'analyze' to the analyze file system.

VI. On selecting analyze, it will direct to a new page with the options to select the analysis mode. Refer figure 3.13

Figure 3.13 Options to analyze the evidence file system

Refer to Evidence frisking techniques of Autopsy section 3.2.2 for details about the options in figure 3.13.

3.2.3 Scalpel [13]

"Scalpel is fast file carver that reads a database of header and footer definitions and extracts matching files or data fragments from a set of image files" [13]

Scalpel is open source forensic tool, originally based on powerful forensic tool 'Foremost' and far more efficient that its predecessor. Independent of file system it can carve files from ext2/3, FATx, NTFS, raw partitions. It is very powerful tool in the field of digital forensics and file carving. Linux is the most preferred platform to use scalpel. Scalpel version 1.60 was used for the experiments in this project. It can downloaded form [13] or can be installed directly from Ubuntu terminal (refer steps in section 3.2.3.1). Image file partitions acquired using adb in section 3.2.1 are used for carving purposes. It is a command line tool and carve files whose header and footer are defined in its scalpel.conf file. The file formats that are not defined in scalpel.conf file can be added manually and easily carved. Scalpel supports multithreading and asynchronous I/O.

3.2.3.0 Scalpel Internals [16]

Working of Scalpel is based on a configuration file scalpel.conf in which different types of files to be carved are specified with their appropriate header and footer information. Scalpel.conf also defines the maximum size of the file to be carved. Scalpel works in two sequential passes on the image file partition. Whole disk image is read in large blocks in its first pass. All the blocks are searched for headers and footers. Database is maintained for the location of the headers searched. When the searching of header is complete, the search for footer starts. After the completion of first pass, scalpel maintains an all-inclusive table of the header and footer locations. This table is used for file carving process for the second pass. The work queues control the carving operations and are filled by the table made by scalpel after the first pass. The entire individual queue has the following record types and works in the order of following blocks:

- STARTCARVE
  The file that is to be carved is opened and the operations dealing with carving files are performed in this block.

- STARTSTOPCARVE
  In this block the file is opened and is written with a small portion of the block and closed. File carving operation begins and ends in this particular block.

- CONTINUECARVE
  File under carving is being written with the entire contents of this block and the file stays open.

- STOPCARVE
  Carving operation finished in this block. File is now closed.
  In the second pass the image file partition is again processed in blocks.

3.2.3.1. Working with Scalpel [14]

Following steps were accomplished to carve out the files based on test cases I & II.

a. As mentioned previously the development machine used was Ubuntu, scalpel can be installed by giving the following command:

```
apt-get install scalpel
```

b. File type of the video in test case I is 3GPP, this file format is not pre-defined in *scalpel.conf* file. In order to carve out 3GPP files it needs to be added to *scalpel.conf* file. Browse to the location /etc/scalpel/scalpel.conf and use nano command to edit this *scalpel.conf* file. Similarly for carving out the email records, header and footer information of email also has to be amended in the *scalpel.conf* file.

c. For amending the *scalpel.conf* file with the details about the 3GPP and email format, include the following lines in the file [15]:

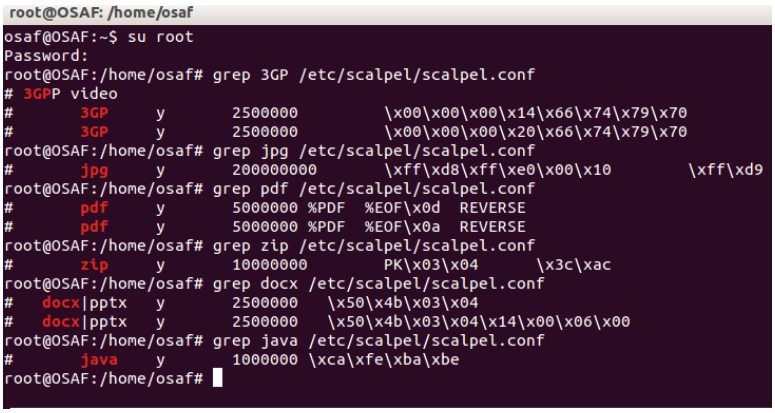
```
3GP    y       2500000       \x00\x00\x00\x14\x66\x74\x79\x70
Mail   y       2500000       \x41\x4F\x4c\x56\x4d
```

d. In the terminal write the following command for searching the header-footer of the file to be carved from *scalpel.conf* file

```
grep Fileformat /etc/scalpel/scalpel.conf
```

The following shows the output of above command on different files in their header and footer format.

Figure 3.14



In figure 3.14 the fields marked with red are the extensions of the files to be carved, 'y' represents yes which means this file format has to be searched, next column is set of numbers gives the max file size, next is the header bytes and some files like jpg, zip have file footer.

e. For example the jpg files are to be carved out for test case II. From step d. the jpg definition has been located. The *jpg.conf* file is created to store the header and footer definitions of the jpeg files to be carved. This step is done for the reason that, the deleted files can be recovered only when the '#' from the definitions of the files is being removed. And to avoid the need to edit the *scalpel.conf* file for future use, the header and footer definitions are copied to a different *filename.conf* file and from there the # can be easily deleted and the integrity of *scalpel.conf* file remains intact.
Now add them into a *jpg.conf* file by the writing the following lines in the terminal –

```
grep jpg /etc/scalpel/scalpel.conf >jpg.conf
and edit the jpg.conf file using:
nano jpg.conf and
uncomment the # from the jpg definition.
```

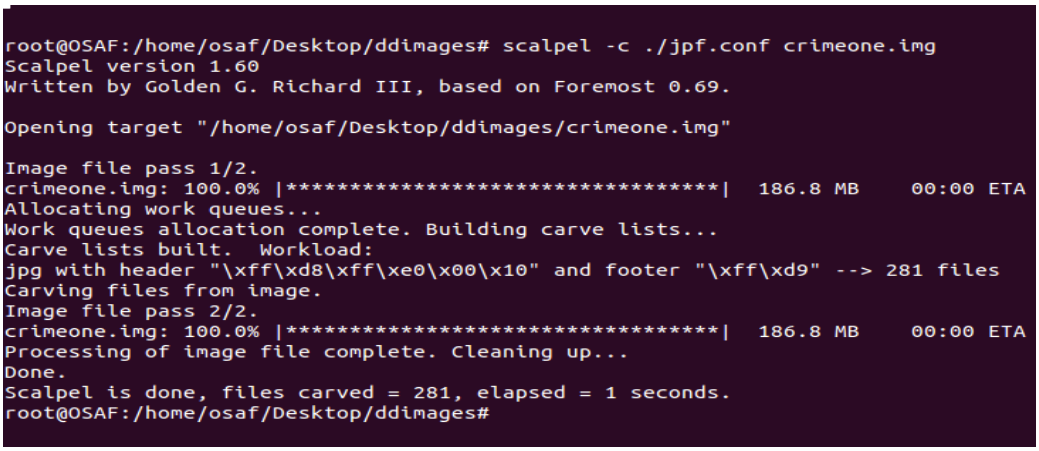
f. Browse to the location where the image file is saved and copy the *jpg.conf* file to that location.

Final step in carving out the jpg files is to write the following line in the terminal –

```
scalpel -c ./jpg.conf crimeone.img
```

Figure 3.15 shows Scalpel process of carving out the files, it works in two passes. Total 281 jpg files were carved out.

Figure 3.15 Scalpel carving process



From the first pass it carved out 281 jpeg files and the final carving process was completed in second pass. List of files based on the testcases is as follows:

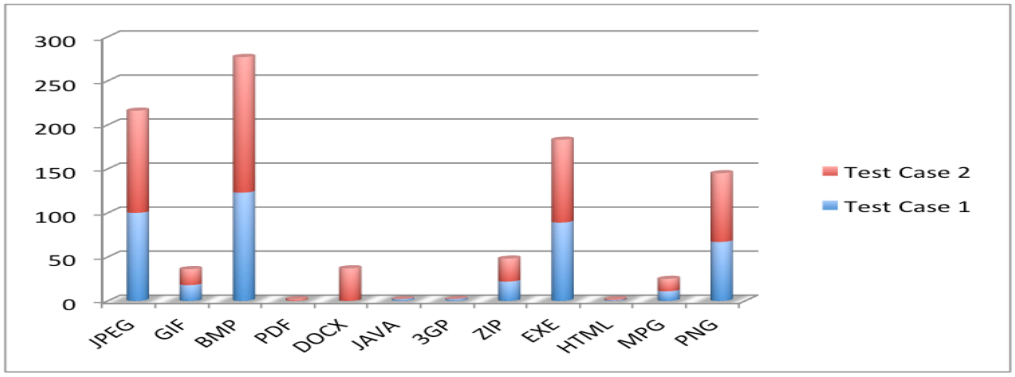
- ✓ 3GP video files – 2
- ✓ docx files – 3
- ✓ jpeg images – 6
- ✓ pdf files – 2
- ✓ java files – 0
- ✓ zip files – 1

Chapter 4 will cover the details about the files retrieved with scalpel and its comparison with the actual data, properties of the file etc.

g. A folder named scalpel-output was generated with a audit file giving the details about the process shown in figure 3.2

The following bar graph represents all the files recovered by using scalpel on both the test cases:

Figure 3.16 Data acquired from both test cases - Scalpel



3.2.4 AFLogical [1] [17]

AFLogical OSE is a free Android forensic tool provided by viaForensics. Any device that is running under Android 1.5 or later can be analyzed using this version of the AFLogical. It is an application provided by viaForensics, and uses the content provider architecture to extract the data from the device. For example content providers like SMS/MMS, Contacts, Browser history, Calendar, etc.
The version 1.5.2 of AFLogical is used for the experiments in this project. This version can extract data from the 41 content providers. The output of the data extraction is stored in the SD card of the device in CSV format and an info.xml file is generated containing the details about the installed applications and device. Following is the list of currently supported content providers:

- Browser Bookmarks
- Browser Searches
- Calendars
- Calendar Attendees
- Calendar Events
- Calendar Extended Properties
- Calendar Reminders
- Call Log Calls
- Contacts Contact Methods
- Contacts Extensions
- Contacts Groups
- Contacts Organizations
- Contacts Phones
- Contacts Settings
- External Media
- External Image Media
- External Image Thumb Media
- External Videos
- IM Account
- IM Accounts
- IM Chats
- IM Contacts Provider (IM Contacts)
- IM Invitations
- IM Messages
- Phone Storage (HTC Incredible)
- Search History
- SMS
- IM Providers
- IM Provider Settings
- Internal Image Media
- Internal Image Thumb Media
- Internal Videos
- Maps-Friends
- Maps-Friends extra
- Maps-Friends contacts
- MMS
- Mms Parts Provider (MMSParts)
- Notes
- People
- People Deleted3
- Social Contracts Activities

Features of AFLogical [18]

I. PIN LOCK/Pattern bypass of a locked Android device, provided adb is enabled.

II. Device Imaging - Functionality to forensically extract the data from the internal memory of the device from the device itself. Unlike Autopsy and Scalpel, it does not depend upon the image file system partitions generated by ADB.

III. Sleuth Kit Timeline – Creates a timeline portraying the information about the files, like when they were last accessed and modified.

IV. Data acquisition, reporting and analyzing

V. The results of searching and sorting can be printed in Pdf

VI. Pre-configured VM, supports Linux, MAC and Windows

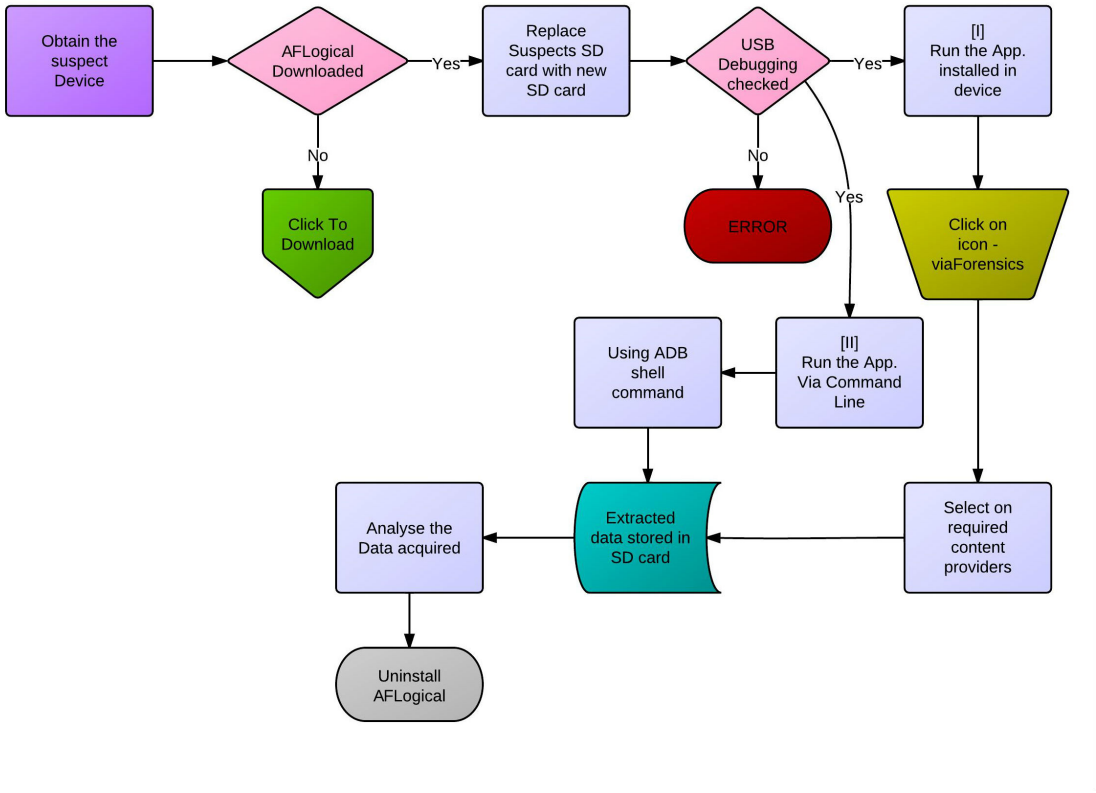
VII. Browser history, call logs, contacts, SMS etc are recoverable

VIII. Forensically sound data is extracted.

3.2.4.1 Working with AFLogical [1]

Following is the flowchart describing steps to run AFLogical on the Android device

Figure 3.17 Steps to run AFlogical



Steps to run AFLogical:

I. AFLogical is available for download at [19]. Registration on the website and approval from viaForensics is needed for downloading.

II. Replace the device's SD card with a new forensically wiped SD card.

III. Connect the device with the development machine (section 2.1), check the USB debugging (section 1.3.3) is enabled.

IV. From terminal, using adb check for the devices attached (section 1.4)

V. The downloaded application for AFLogical can be installed using ADB.

```
adb install ~/AFLogical-OSE_1.5.2.apk
```

VI. Once the application is installed in the device, it can be directly accessed from the device itself.

VII. There will be an application with AFLogical OSE icon will be created. Click the icon to launch the app.

VIII. Select the content providers for the respective data collection.

IX. After the extraction is completed, all the data gets stored into a folder named forensics in the SD card of the device, there will be subdirectory inside the forensics folder named after the date in YYYYMMDD.HHMM format.

X. adb pull can be used to copy the folder onto the development machine for further analyzing purposes. All the extracted data is stored in CSV format and can be viewed using any spreadsheet. One more file named info.xml is contained in the forensics folder. Info.xml gives details about Android version, network provider, IMEI, IMSI, installed applications etc.

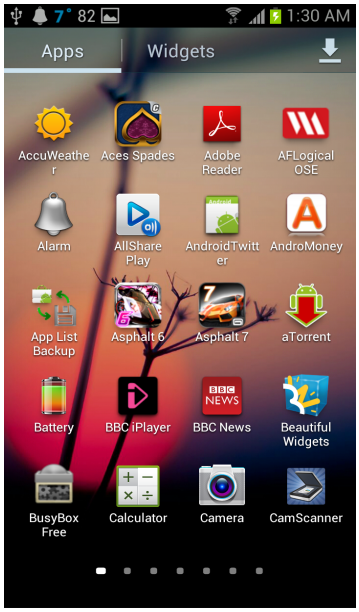
XI. Uninstall the application from the device when all the acquisition process is completed.

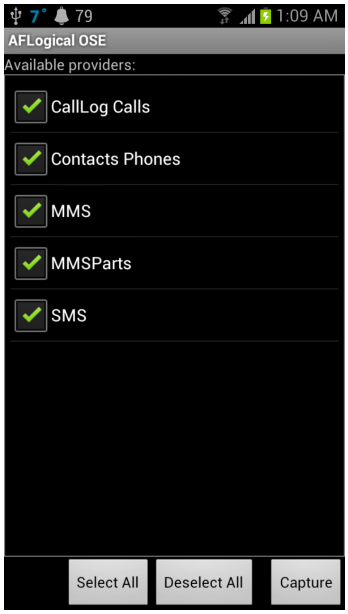
Screenshots for AFLogical in working

AFLogical app installed into the apps menu of the device, the application can now be directly launched from the device. Figure 3.18

Selecting the content providers depending on the files to be extracted. Click the capture button. Figure 3.19

Screen shot showing the final step of AFLogical on completion of the data extraction. Figure 3.20





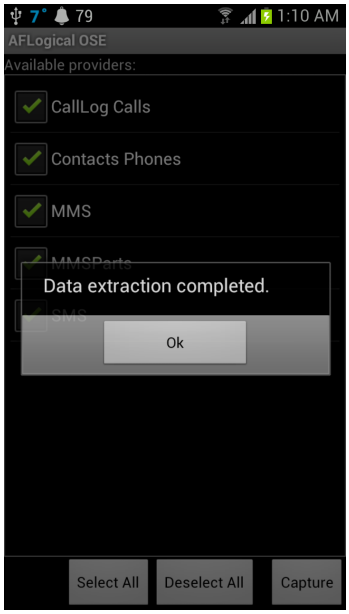
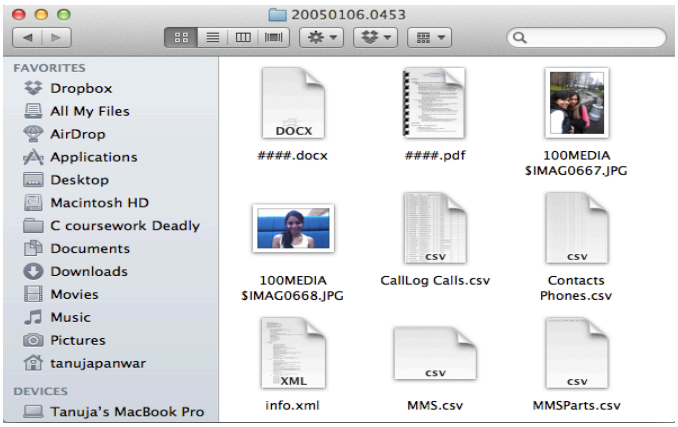


Figure 3.21 gives a sample screenshot of the files recovered.



Analysis of acquired files and its comparison with the actual data is done in chapter 4.

3.3 Summary

To explicate the choice of logical analysis for experiments, difference between both logical and physical techniques was described. All the four forensic tools used in this research project are explained, with their disparate capabilities. Steps to implement these tools are illustrated. For making things more clear, a number of figures are used for a visual representation of the tool operations.

Result Analysis

4

4.0 Introduction

This chapter details about the analysis of the results attained from implementation in chapter 3. Results achieved by tools are discussed, and are compared with the actual data that was counted manually when performing the test cases. All the comparison results are detailed in tables to get a visual idea of the results obtained. Tools were tested critically on various parameters other than just the file retrieval. Parameters such as CPU utilization, load average, RAM utilization, UP time etc. were also counted for analyzing the performance of the tools.

4.1 Analysis of results

All the results retrieved during the implementation, are tested against the actual data. The raw actual data acts a benchmark for the comparison of results attained from implementation. All the levels of data analysis are tabulated.

4.1.1 ADB Evaluation

All the features of adb based on both the test cases 2.2.1 and 2.2.2 have been portrayed in figure 3.2. The files recovered with their respective location and related information is detailed in table 4.1.

Table 4.1. ADB file recovery details [9]

Analysis of image partitions via ADB	Test Case I & II
File recovery : Both the test cases	Files recovered! Command: sudo mount –o loop videofile.img /path
Email Information	/data/com.android.email/databases/EmailProvider.db File name, MIME type, size, content uri, timestamp, fromlist& tolist etc.
Bluetooth Information	/data/com.android.bluetooth/databases/btopp.db Files exchanged, MAC address of paired devices, timestamp, size, MIME type, total bytes sent, etc.
WiFi Connection Details	/misc/wifi Visited networks, Passwords, etc.
YouTube upload Information	/data/com.google.android.youtube/com.google.android.youtube.credentials.xml ID of the user logged to youtube, timestamp
Facebook upload Information	/data/com.facebook.katana/databases/ Name of the file uploaded, user login ID, password, timestamp, size of file, etc.
Drop-box upload Information	/data/com.dropbox.android/databases/db.db Filenames, timestamp, total bytes, hash value, full path, display name & folder, etc.

Table 4.1 presents the files and their associated information retrieved using ADB. All the databases were browsed using powerful tool SQLite3. The location to the databases from where the information related to files was also found.

- Information about the email sent in both the test cases was found in the directory /data/com.android.email, a database named EmailProvider.db gives the details about the files attached to the email, information such as name of the file, MIME type, size, contentUri was found. In the same directory a database named webview.db gives the details about the username and password of the email used to send the files.

- All the information about the applications such as Facebook, YouTube, Dropbox, Gmail can be found in the databases named db.db by browsing according to the location specified in the table 3.1. Also details about the login ID and password of the user, files uploaded, size of the files, were found.

- Bluetooth transaction history can be found in database named btopp.db, information such as names of the files exchanged, their location on the SD card, MIME type, total bytes sent, request id, transaction id, start timestamp, end timestamp, destination device name, MAC address of the paired devices.

- Wi-Fi transfer details can be found in /misc/wifi/wpa_supplicant.conf, name of the visited networks and password used to connect to that network was recovered

4.1.1.1. Comparison of Results

Table 4.2 and 4.3 gives the details about comparison between the data acquired and the actual data for test cases I & II. The entry 'Meet' indicates that the tool has successfully met the expected results, 'Below' indicates that tool has partially met the expected conditions, 'Above' indicates that tool has exceptionally exceeded in meeting all the expectations. NA indicates that the detail related to that particular file is not available.

Table 4.2 Comparison chart ADB Test case I

Comparison chart	Test case I - ADB	
File : 3GPP movie	Acquired Data	Actual Data
Video Recovered	Meet	Meet
Duration of video	Below	21 seconds
Time Last Accessed	17:00:03	14:45:23
Date Created	Meet	Meet
Facebook upload time	11:14:03	17:29:12
Email sent timestamp	NA	15:29:53
YouTube Upload timestamp	NA	15:20:29
Bluetooth transfer time	Meet	Meet
Sender's email id, subject and body of email, receiver's id	Above	Meet
Login ID used for Facebook and YouTube upload	Meet	Meet
Bluetooth transfer details - parties involved, pairing info, password, timestamp	Above	Meet
Wi-Fi used to upload the Video – Name(ssid) of Wi-Fi, Password, data and time	Above	Meet

Table 4.3 Comparison chart ADB Test case II

Comparison Chart	Test case II -	ADB
File Information	Acquired Data	Actual Data
Jpeg, Docx, Pdf, Java, Zip files Recovered	Meet	Meet
Time Last Accessed	NA	23:26:14
Date Created	NA	30-08-12
Email sent timestamp	Below	Meet
Bluetooth transfer time	Meet	Meet
Dropbox Upload time	Meet	Meet
Deleted Files recovery	Above	Meet
Sender's email id, subject and body of email, receiver's id	Above	Meet
Bluetooth transfer details - parties involved, pairing info, password, timestamp	Above	Meet
Wi-Fi used to upload the Video – Name(ssid) of Wi-Fi, Password, timestamp	Above	Meet

From table 4.2 and 4.3, the video and the files were successfully recovered, time the video was last accessed was not as expected and for test case II this information was not available, upload timing of the Facebook does not match with the actual data, timestamps for Email and YouTube upload was not available, information relating to the email sent was correct and exceeded the amount of details about the file counted manually. An interesting aspect of ADB was found that for Bluetooth and Wi-Fi usage the information recovered was manifold when compared to the actual data. Similar details were retrieved for test case II. Information related to drop box activity was manifold.

For further critical evaluation of the performance of the tools, other factors such as the CPU usage, running processes, RAM utilization, Load average on CPU were also counted. This information was recorded when the tool was running. Information is taken at different times for more accurate results. Command 'top' was used to get the details; a running list of ordered processes is created and can be specified according to user's needs. Basically the processor activity in real time can obtained using 'top' command.

Table 4.4 Results of 'top' command for ADB.

Activity Monitor Chart :	
Tasks	Total = 171, Running = 3, Sleeping = 163, Stopped = 5
CPU utilization by user related processes	57.8%
CPU utilization by system related processes	42.7%
Load Average on CPU in 5min, 10min, 15min span	2.46, 2.35, 2.58
Ram utilization – Total , Used, Buffer	3GB, 2.80GB, 65MB
UP time	16 hours 42min

From table 4.4, CPU utilization for both system related and user related processes was noted, Load average on CPU in three different time spans, Ram utilization values for Total, used and buffer is shown, also the up time tells that the machine was up for 16hours 42mins.

4.1.2 Autopsy Evaluation

Figure 4.1 represents all the features of Autopsy based on test cases I and II. These features were encountered when analyzing the image file partitions with Autopsy. Some of the interesting features met when working with Autopsy are also listed; such as MD5 hash value, GPS information, and GUI interface are exclusive to Autopsy only.

Figure 4.1 Features of Autopsy

Autopsy
✧ Suspect's file recovery
✧ Email information
✧ Bluetooth transfer details
✧ Wi-Fi access information
✧ Application data recovery
✧ Deleted files recovery
✧ Browsing History
✧ Password recovery
✧ GPS information
✧ MD5 Hash value
✧ GUI interface
✧ Supports FAT16 file system
✧ Open source toolkit
✧ Efficient CPU utilization

4.1.2.1. Autopsy Results

Following screen shots details about the files recovered, all the results are based on the test cases.

Figure 4.2 Video recovered – Testcase I

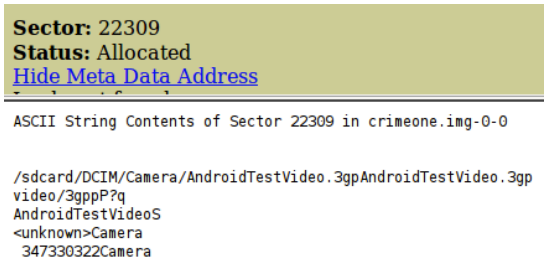


Figure 4.2
Details about the name of the video file, type of video – 3GPP
Code for camera used, sector number in the file system, status, and duration of video recovered. No information about the timestamp was recorded.

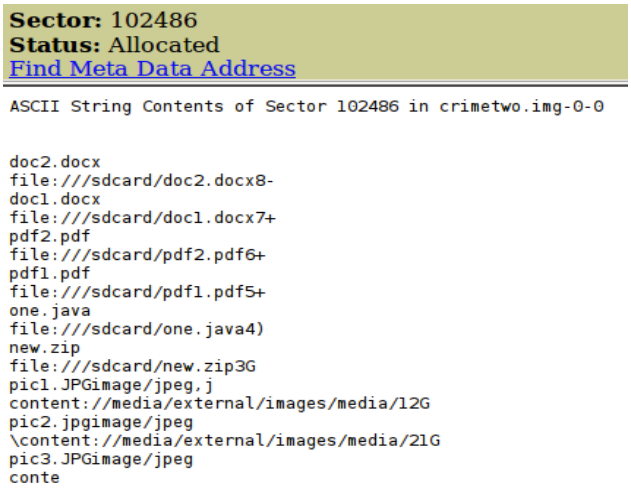
Facebook upload details

Figure 4.3

When browsed through the details of Facebook activities, information about the date and time of the file uploaded, location of the file in the device and name of the file were found.

Documents Recovery Testcase II

Figure 4.4 Documents recovery Autopsy



Information about the files last accessed and date created was not recovered. Names of the files and location in the device were recovered.

Email upload information

Figure 4.4 Email attachment activities

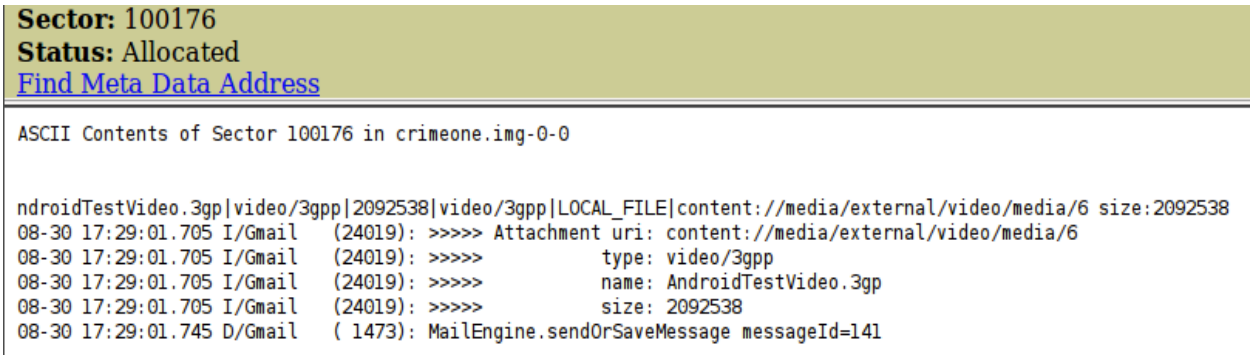
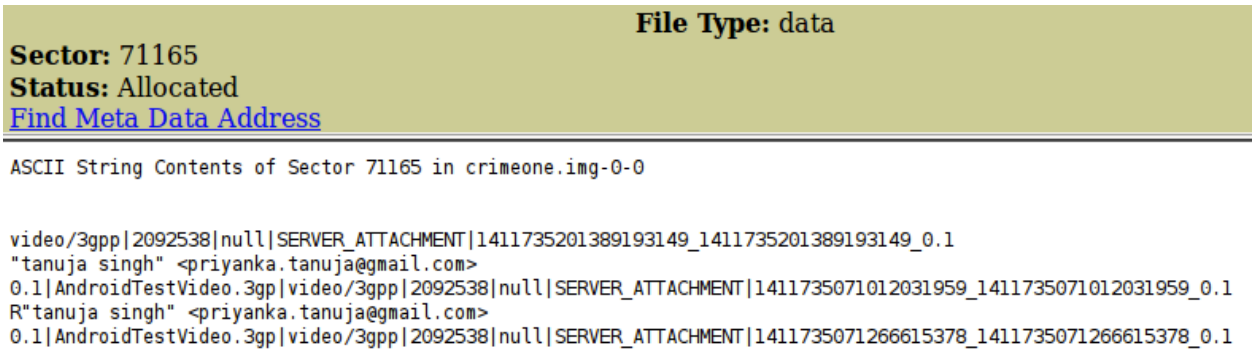


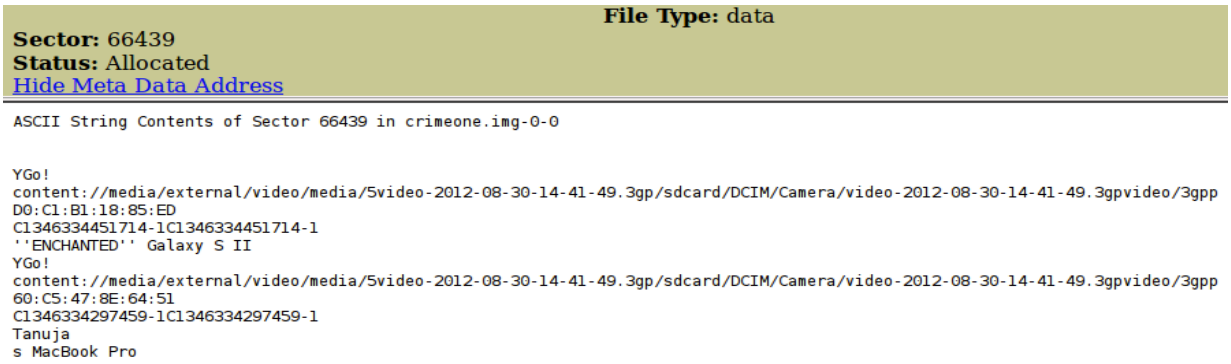
Figure 4.5 Email sending activity



Information about the date and time of the email when the compose activity was in progress and when the email sent was found. Email id used to send video file, Uri content of the video file in the device and the format of the video were recovered. File type is data that represents the information about the email recovered and status is allocated which means information related to this email activity is still in the file system and not deleted. When browsed through the corresponding sectors of the file further information related to the email activities was found. Refer figures 4.4 and 4.5.

Bluetooth Transfer Information

Figure 4.6 Bluetooth transfer information



File Type: data
Sector: 66439
Status: Allocated
Hide Meta Data Address

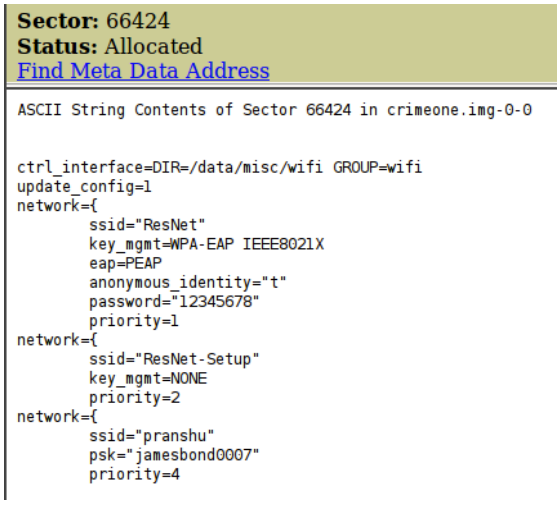
ASCII String Contents of Sector 66439 in crimeone.img-0-0

YGo!
content://media/external/video/media/5video-2012-08-30-14-41-49.3gp/sdcard/DCIM/Camera/video-2012-08-30-14-41-49.3gpvideo/3gpp
D0:C1:B1:18:85:ED
C1346334451714-1C1346334451714-1
''ENCHANTED'' Galaxy S II
YGo!
content://media/external/video/media/5video-2012-08-30-14-41-49.3gp/sdcard/DCIM/Camera/video-2012-08-30-14-41-49.3gpvideo/3gpp
60:C5:47:8E:64:51
C1346334297459-1C1346334297459-1
Tanuja
s MacBook Pro

Information such as name of the uploaded file, MAC address of the paired devices, name of the device the file was transferred to and date and time was recovered. From figure 4.6, the two devices used to transfer the files using Bluetooth were Samsung Galaxy SII and MacBook pro. The MAC address recovered for Samsung Galaxy SII and MacBook pro was D0:C1:B1:18:85:ED and 60:C5:47:8E:64:51 respectively.

Wi-Fi Connection information

Figure 4.7 Wi-Fi connection details



Sector: 66424
Status: Allocated
Find Meta Data Address

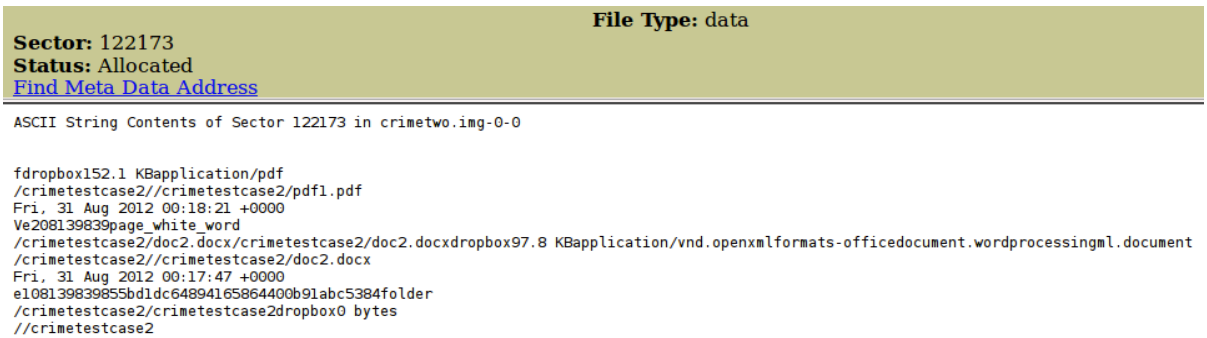
ASCII String Contents of Sector 66424 in crimeone.img-0-0

ctrl_interface=DIR=/data/misc/wifi GROUP=wifi
update_config=1
network={
        ssid="ResNet"
        key_mgmt=WPA-EAP IEEE8021X
        eap=PEAP
        anonymous_identity="t"
        password="12345678"
        priority=1
network={
        ssid="ResNet-Setup"
        key_mgmt=NONE
        priority=2
network={
        ssid="pranshu"
        psk="jamesbond0007"
        priority=4

Information such as name of the Wi-Fi network and password were recovered. Also the files sent using Wi-Fi was recovered.

Dropbox upload information

Figure 4.8 Dropbox activities



File Type: data
Sector: 122173
Status: Allocated
Find Meta Data Address

ASCII String Contents of Sector 122173 in crimetwo.img-0-0

fdropbox152.1 KBapplication/pdf
/crimetestcase2//crimetestcase2/pdf1.pdf
Fri, 31 Aug 2012 00:18:21 +0000
Ve208139839page_white_word
/crimetestcase2/doc2.docx/crimetestcase2/doc2.docxdropbox97.8 KBapplication/vnd.openxmlformats-officedocument.wordprocessingml.document
/crimetestcase2//crimetestcase2/doc2.docx
Fri, 31 Aug 2012 00:17:47 +0000
e1o8139839855bd1dc64894165864400b91abc5384folder
/crimetestcase2/crimetestcase2dropbox0 bytes
//crimetestcase2

Information about names of the files uploaded on dropbox, date and time, and number of bytes were recovered. Figure 4.8 shows the information about the pdf and docx files being uploaded. Date on which the files were uploaded was 31 august 2012 and location of the directory where the files were stored is also specified. Here the files are saved in directory crimetestcase2.

4.1.2.2. Comparison of Results

Table 4.5 and 4.6 gives the details about comparison between the data acquired and the actual data for test cases I & II. Refer to section 4.1.1.1 for the meaning of 'Meet', 'NA', 'Above', 'Below'.

Table 4.5. Comparison Chart Autopsy Test case I

Comparison chart	Test case I - Autopsy	
File : 3GPP movie	Acquired Data	Actual Data
Video Recovered	Meet	Meet
Duration of video	Meet	Meet
Time Last Accessed	NA	14:45:23
Date Created	NA	30-08-12
Facebook upload time	16:11:53	17:29:12
Email sent timestamp	17:29:04	15:29:53
YouTube Upload timestamp	NA	15:20:29
Bluetooth transfer time	Meet	Meet
Sender's email id, subject and body of email, receiver's id	Meet	Meet
Login ID used for Facebook and YouTube upload	Meet	Meet
Bluetooth transfer details - parties involved, pairing info, password, timestamp	Meet	Meet
Wi-Fi used to upload the Video – Name(ssid) of Wi-Fi, Password, data and time	Meet	Meet

Table 4.6 Comparison Chart Autopsy Test case II

Comparison chart	Test case II -	Autopsy
File Information	Acquired Data	Actual Data
Jpeg, Docx, Pdf, Java, Zip files Recovered	Meet	Meet
Time Last Accessed	NA	23:26:14
Date Created	NA	30-08-12
Email sent timestamp	15:50:04	1:06:23
Bluetooth transfer time	Meet	Meet
Dropbox Upload time	1:22:02	2:22:18
Deleted Files recovery	Below	Meet
Sender's email id, subject and body of email, receiver's id	Meet	Meet
Bluetooth transfer details - parties involved, pairing info, password, timestamp	Meet	Meet
Wi-Fi used to upload the Video – Name(ssid) of Wi-Fi, Password, timestamp	Meet	Meet

From table 4.5, it can be inferred that the 3GPP video was successfully recovered. Duration of video was matched with the actual data, whereas ADB was unable to provide this information. Time last accessed and date the video created is not available. ADB was also unable to recover

information about these two. Upload timing of video on Facebook and time when the email was sent does not match. None of the tool, ADB or autopsy was able to recover correct information about the timestamp related to email, Facebook or YouTube upload.

Information about the Bluetooth transfer, Wi-Fi networks were recovered and matched with the actual data. Unlike ADB, Autopsy did not recover any information, which exceeded the expected results.

From table 4.6 Dropbox activity does not match, recovery of deleted files was below expected results. All the recovered deleted files do not match the manual records. No timestamp recovery for files last accessed and created.

Table 4.7 Results of TOP command for Autopsy

Activity Monitor Chart :	
Tasks	Total = 130, Running = 1, Sleeping = 128, Zombie = 1
CPU utilization by user related processes	16.3%
CPU utilization by system related processes	3.0%
Load Average on CPU in 5min, 10min, 15min span	1.22, 2.38, 3.13
Ram utilization – Total , Used, Buffer	3.6 GB, 3.5 GB, 35 MB
UP time	2 hours 39min

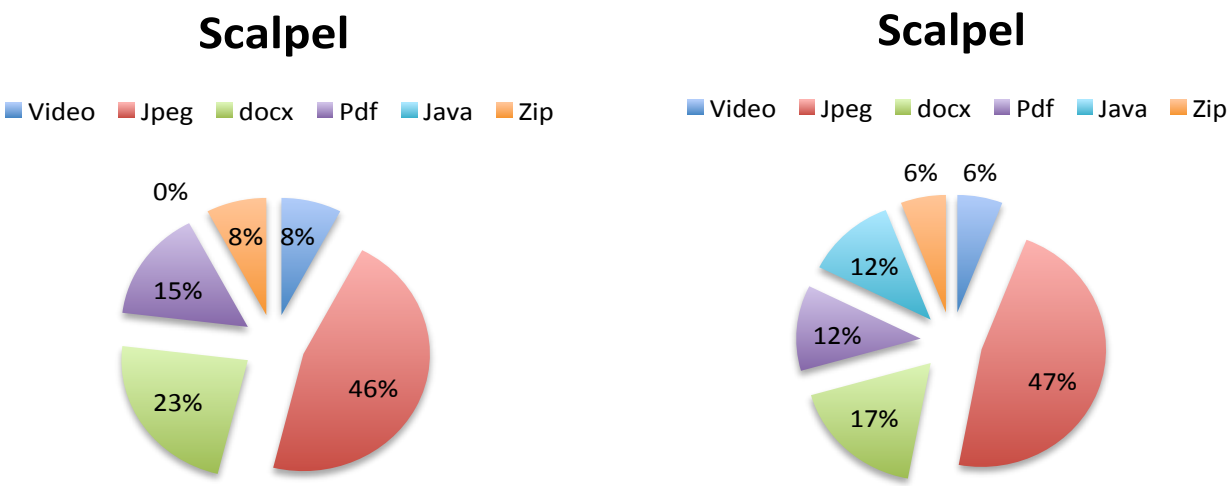
From table 4.7, CPU utilization for both system related and user related processes was noted, Load average on CPU in three different time spans, Ram utilization values for Total, used and buffer is shown, also the up time tells that the machine was up for 2 hours 39 minutes.

An interesting thing was noted, when compared with the Activity chart of ADB, CPU utilization by user related processed is 3.54 times less in Autopsy. And for system related processes it is 14.23 times less than ADB. Also the load average on CPU is nearly half in Autopsy.

But in the case of RAM utilization, from total of 3.6GB RAM in machine, 3.5GB was used, and 35MB was in buffers. When compared to ADB, ADB from total of 3GB RAM, 2.80GB was used and 65MB was in buffers.

4.1.3. Scalpel Evaluation

Figure 4.9 Files carved – Scalpel Figure 4.10 Actual files counted manually



From figure 4.9, Scalpel has recovered when run in its audit mode. In scalpel audit mode all the file type formats configures in scalpel.conf file gets carved and are stored in a folder named scalpel-output. The above results were counted when manually browsed the scalpel-output directory. Carving of 3GP video was explicitly performed and was successful. Refer table 4.8 for the detailed information about the same.

Similarly for test case II, all the files based on the test case were explicitly carved and noted down. Refer table 4.9 for the comparison chart of the acquired data versus actual data.

Table 4.8 Comparison chart Scalpel Test case I

Comparison Chart:	**Test Case I**	
File: 3GP Video	Acquired Data	Actual Data
Number	2	1
Size	2.5MB	2.1MB
Dimensions	NA	320x240
Bitrate	128 Kbps	780 Kbps
Date Created	30/08/2012	31/08/2012
Time Created	14:45:09	05:48:48
File Extension	.MOV	.3GP

From table 4.8, total number of 3GP videos recovered was two and when manually recorded the media information from the device only one video file was recorded. Both video files are not readable. Files recovered were stored in the sub-directory named 3GP under the main directory scalpel-output. Both files had extension .MOV and size of the video file does not match. On contrary to ADB and Autopsy, Scalpel was able to recover the details about the file last accessed and date created. But details about the both do not match the actual data. Interesting information, bit rate was also found, though partially correct.

Table 4.9 Comparison chart Scalpel Test case II

Chart:	Test Case	Two	Acquired Data	Vs.	Actual Data	
Files	Acquired Data of Suspect/ Actual Data	State of Acquired Data	Size/ Actual size	Date Last Modified / Actual Date	Time / Actual Time	Deleted File Recovery
Docx	3 / 3	Partially-Readable	2.5MB / 103kb	18-11-2012/ 18-11-2012	14:16/ 2:03	Yes
JPEG	6 / 8	Readable	4.7MB/ 6.95	12-08-2012/ 12-08-2012	17:15/ 13:23	Yes
PDF	2 / 2	Not-Readable	143kb/ 246kb	3-07-2012/ 18-04-2011	10:58/ 23:36	NA
JAVA	NA / 2	NA	NA/12kb	NA/ 30-04-2012	NA/ 19:11	NA
ZIP	1 / 1	NA	89kb/ 4kb	NA/ 7-08-2012	NA/ 00:20	NA

Table 4.9, all the files based on the test case II were only considered for the comparison purposes. The entire Docx files were recovered, but partially readable and there was a huge difference in the size of the files. Date last accessed is correct but the time does not match. All the deleted Docx files were successfully carved; only 6 among the 8-jpeg files were carved. The two files deleted were retrieved. Scalpel was unable to recover the JAVA files, and one zip file recovery was notified but there were no zip files present.

After discussing about both the test cases, it can be easily inferred that Scalpel was successful in retrieving information about the files. But when compared to the features of ADB and Autopsy, critical analysis of the information by checking various parameters like Bluetooth, Wi-Fi, Email, Facebook etc. was not possible. An advantage above this limitation was that scalpel was fast, and searching of data exclusively was possible. No manual browsing of data was required. Just looking for the specific files could give the results, unlike ADB and autopsy, were a lot of manual search had to be done.

Table 4.10 Results of TOP command for Scalpel

Activity Monitor Chart :	
Tasks	Total = 158, Running = 1, Sleeping = 156, Zombie = 1
CPU utilization by user related processes	62.3%
CPU utilization by system related processes	14.3%
Load Average on CPU in 5min, 10min, 15min span	3.43, 3.56, 4.38
Ram utilization – Total , Used, Buffer	3GB, 1.90GB, 81MB
UP time	5 hours 17min

From table 4.10, CPU utilization by user related processes were higher on comparison with ADB and Autopsy. But the CPU utilization by system related processes were moderate. RAM utilization, used RAM was lower amongst the three tools. 5 hours 17 minutes was the up time and Load average on CPU for all the three time spans higher than the other two tools.

4.1.4 AFLogical Evaluation

Figure 4.11 represents all the features of AFLogical based on test cases I and II. [Note: figure 4.11 just represents the features of AFLogical. No other parameter was taken into consideration.]

Figure 4.11- Features of AFLogical

AFLogical

- ❑ Internal and External videos
- ❑ Images and other media files
- ❑ Browser search history
- ❑ Call history/SMS/MMS
- ❑ Network Provider
- ❑ Installed applications
- ❑ IMSI and IMEI number
- ❑ Deleted contacts recovery
- ❑ IM accounts/chats/Providers
- ❑ Bypass locked phone
- ❑ Pre-configures VM
- ❑ Open source
- ❑ Image phone memory directly from device

AFlogical exceptionally recovered all the details about the video files and picture files from the device. All the messages, Call history, installed applications found was accurate when compared with the actual data. Critical details about deleted contacts, messages, their timestamp was recovered. For the features mentioned in figure 4.11, individual files were generated to store the details about them. So there was no additional overhead in manually searching for the relevant files for investigation. Reporting of data in pdf format was an interesting feature.

Based on the content providers (3.2.4) and test cases, information relating to media files, such as video and Jpeg images was successfully recovered. Based on 'browser search' content provider; trivial information about the login timings of Gmail, Facebook, YouTube, Dropbox was recovered. Under no circumstances the information related to Bluetooth and Wi-Fi was identified. Since the open source version AFLogical was used, so only a limited number of content providers were supported.

Table 4.11 Comparison chart AFLogical

Test case I		
File Attributes	Acquired Data	Actual Data
Number	1	1
Size	2.1MB	2.1MB
Dimensions	320x240	320x240
Bitrate	780 Kbps	780 Kbps
Date Created	31/08/2012	31/08/2012
Time Created	05:48:33	05:48:48
File Extension	.3GP	.3GP

Table 4.12 Comparison chart AFLogical Test case II

Test Case II Acquired data Vs. Actual data						
Files	Acquired Data of Suspect/ Actual Data	State of Acquired Data	Size/ Actual size	Date Last Modified / Actual Date	Time / Actual Time	Deleted File Recovery
Docx	0 / 3	NA	NA / 103kb	NA / 18-11-2012	NA / 2:03	No
JPEG	8 / 8	Readable	6.90 MB/ 6.95 MB	12-08-2012/ 12-08-2012	13:23 / 13:23	Yes
PDF	1 / 2	Not-Readable	NA / 246kb	NA / 18-04-2011	NA / 23:36	Below
JAVA	0 / 2	NA	NA/12kb	NA/ 30-04-2012	NA/ 19:11	NA
ZIP	0 / 1	NA	NA / 4kb	NA/ 7-08-2012	NA/ 00:20	NA

Table 4.11 shows that the information about the video recovered were exceptionally accurate. All the parameters shown in the table were matched with the actual data. As mentioned previously, the open source edition of AFLogical was able to retrieve only this limited information about the Testcase I. Rest of the details like, email, Wi-Fi, Bluetooth, and application data etc. was not found.

Just the video recovery is not enough in deciding the outcome of an investigation. When compared to the criticality of evidence extracted from ADB and Autopsy, AFLogical lags behind. But was better than scalpel when speaking in terms of file recovery only.

Analogously from table 4.12, only information about Jpeg files was found to perfection. No information about rest of the files was found. One Pdf file was recovered and that too was not readable. All the deleted jpeg files were recovered. Also from the browser search history an interesting thing was noted, all the pictures that were browsed using the application Facebook were recovered. But this detail does not support the evidences required for test case II.

Based on the data acquired from all the four tools and test cases, it can be roughly implied that AFLogical was the least suitable tool for the critical analysis of the files. But AFLogical was top-notch when compared with the files it successfully recovered and the same files recovered using the other three tools.

Table 4.13 Results of TOP command for AFLogical

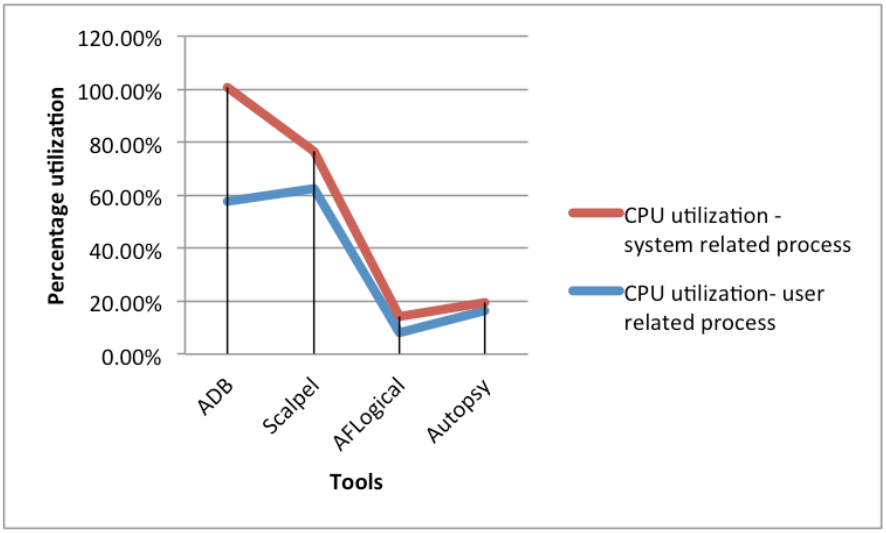
Activity Monitor Chart :	
Tasks	Total = 102, Running = 2, Sleeping = 98, Stuck = 2
CPU utilization by user related processes	8.10%
CPU utilization by system related processes	6.1%
Load Average on CPU in 5min, 10min, 15min span	1.25, 0.84, 0.68
Ram utilization – Total , Used, Buffer	3GB, 989MB, 149MB
UP time	3 hours 25min

Amongst all the four tools CPU utilization was least in AFLogical, also the load average on CPU is less as compared to the other three tools and is analogous with the load average on CPU by Autopsy. AFLogical utilized the RAM in best way, only 989MB of RAM was being used out of total 3GB RAM and 149MB was in buffer. Up time for the machine was 3 hours 25 minutes.

Comparison of results from 'top' command

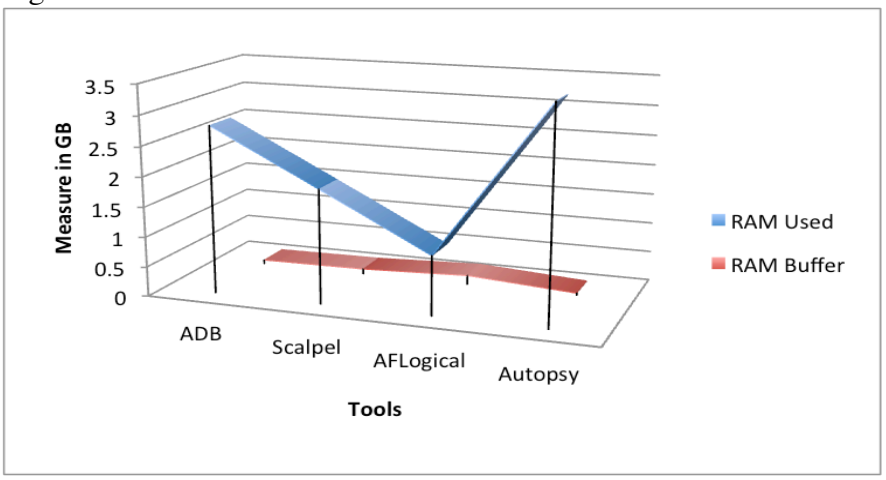
A large amount is data is produced using forensic tools; evaluation of these results is a strenuous task. Graphical representation is used to simplify the task of analyzing at the results and evade the need to look at the numbers.
Values of CPU utilization, Ram utilization and load average on CPU are being graphed.

Figure 4.12 CPU utilization-



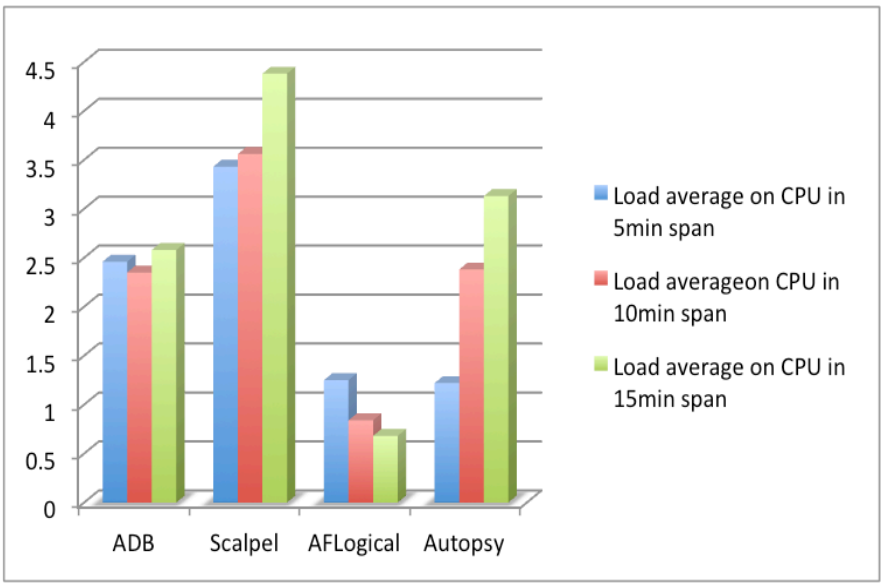
In both the cases, AFLogical was better in CPU utilization. ADB has the highest value among all three tools. Autopsy and AFLogical were nearly same.

Figure 4.13 RAM utilization-



An interesting reading about the autopsy was noted. Unlike CPU utilization, RAM utilization used was highest. AFLogical was better in this case too. ADB and Scalpel have quite comparable results.

Figure 4.14 Load Average on CPU



Yet again, AFLogical gave exceptional results than the other three tools. Scalpel has the highest value for Load average on CPU. The figures for ADB and Autopsy were moderate.

4.2 Summary

All the results retrieved during the implementation, are tested against the actual data. The raw actual data acts a benchmark for the comparison of results attained from implementation. All the levels of data analysis are tabulated comparing the actual data with the retrieved data. Comparison amongst the tools is made; this comparison is based on data acquired from the tools. Parameters other than the data retrieval, like CPU utilization, load average etc., were also measured and mapped. Testing tools on different parameters led to a deeper insight about comparison among the tools and their individual performances.

Validation Testing 5

5.0 Introduction

"Validation is the confirmation by examination and the provision of objective evidence that the particular requirements for a specific intended use are fulfilled." [ISO 17025]

Precision-Recall and CHI Square tests are performed to validate the results from the four tools. Accuracy test and precision recall test measure the performance of tools in terms of data extraction. CHI Square tests checks whether the null hypothesis chosen to compare the acquired and actual data is true or not. All the results from these tests are tabulated. Lastly a grand diagram about the tools comparison is made. This diagram consists of the results obtained from the data acquisition, analysis and validation test.

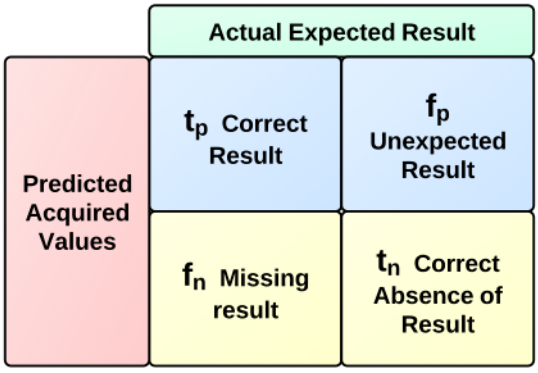
5.1. Precision and Recall [20]

In information retrieval, Precision is the fraction of relevant retrieved instances and Recall is fraction of retrieved relevant instance. Hence both are based on measure of relevance. In easier words, higher the value of Precision means more appropriate results were returned than inappropriate. And high value of Recall means most of the appropriate results were returned.

5.1.1 Definition in Classification context

In order to compare the results obtained from the tools and the actual results, the terms, true positive, true negative, false positive, false negatives were used. Positive and Negative refer to the observed or in forensic terms retrieved values. And the terms true and false signify whether the observed or retrieved values correspond to actual expected values. Figure 5.1 explains the same.

Figure 5.1 Precision-Recall Classification conditions [20]



Definitions:

Precision = $\frac{tp}{tp+fp}$

Recall = $\frac{tp}{tp+fn}$

True negative rate = $\frac{tn}{tn+fp}$

Accuracy = $\frac{tp+tn}{tp+tn+fp+fn}$

Values for the above test are taken from the information retrieved by the tools. Since the capabilities of all the tools are not similar enough, the parameters chosen for the test vary for all the four tools.

ADB

Refer table 4.2 and 4.3 for values used in the test.

1. t_p = 17, all the files were recovered from both the test cases
2. f_p = 2, Mismatch in the values of Facebook upload time and last accessed time of video
3. f_n = 4, based on test cases, the information related to the files which was not available, here, Date and time for email sent, YouTube upload and timestamp for files in test case II were not available.
4. t_n = 0

- Precision = 0.894
- Recall = 0.809
- True negative rate = 0
- Accuracy = 0.739

Autopsy

Refer table 4.5 and 4.5 for the values used in the test.

1. t_p = 17, all the files were recovered from both the test cases.
2. f_p = 4, Mismatch in the email sent timing and dropbox upload time in testcase II , facebook upload and email sent timing for testcase I.
3. f_n = 5, based on test cases the NA information, refer table 4.5 and table 4.6
4. t_n = 0

- Precision = 0.809
- Recall = 0.772
- True negative rate = 0
- Accuracy = 0.653

Scalpel

Refer to table 4.8 and 4.9 for the values used in the test.

1. t_p = 7 (correct readable files form both the test cases)
2. f_p = 5 (Partially readable and not-readable files)
3. f_n = 4 (Files not recovered, i.e fields marked with NA)
4. t_n = 1 (1 Zip file carved, but when checked manually the location of the file, no Zip files were found.

- Precision = 0.583
- Recall = 0.636
- True negative rate = 0.166
- Accuracy = 0.470

AFLogical

Refer to table 4.11 and 4.12 for the values used.

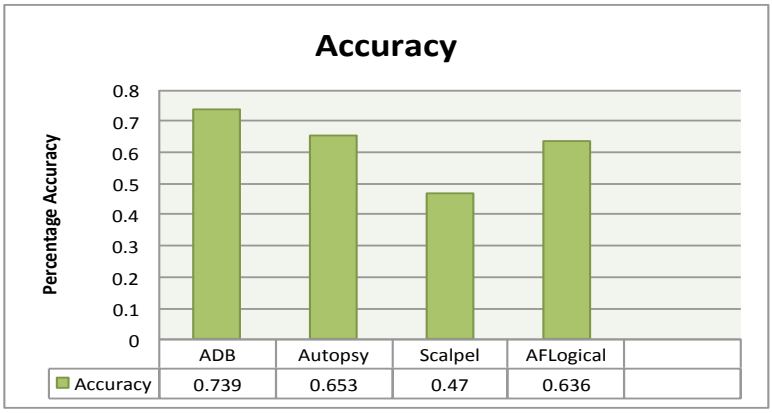
1. t_p = 9 (8 Jpeg and 1 Video file recovered)
2. f_p = 1 (1 Pdf recovered but not readable)
3. f_n = 7 (refer table 4.11 and 4.12 for the files marked NA)
4. t_n = 5 (Besides all the files recovered by AFLogical, Files such as SMS, Contacts etc. were also recovered. These files were accurate but does not provide any evidence based on test cases)

- Precision = 0.900
- Recall = 0.562
- True negative rate = 0.833
- Accuracy = 0.636

Table 5.1 shows the results of accuracy calculated for all the four tools.

Table 5.1	ADB	Autopsy	Scalpel	AFLogical
Accuracy	0.739	0.653	0.470	0.636

Figure 5.2 Accuracy chart

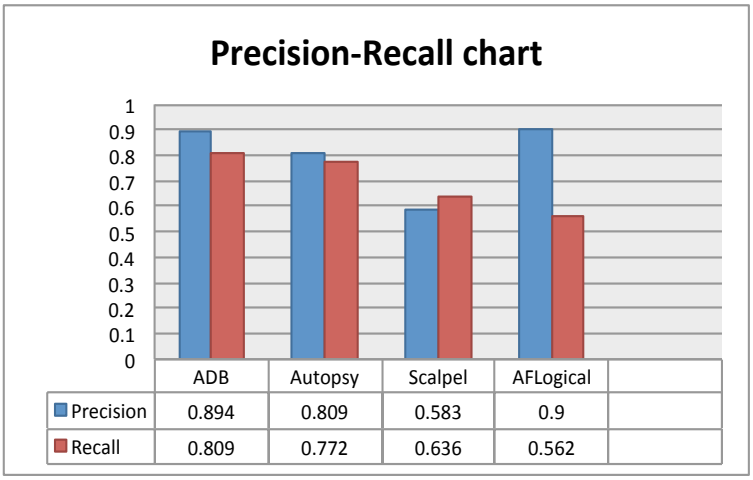


Using the accuracy values obtained from accuracy test the graph in figure 5.2 is plotted. Again from the graph on left, ADB was able to extract evidence in much efficient way. Accuracy for Autopsy and AFLogical is analogous. Scalpel was least accurate. Scalpel also had highest value of load average on CPU.

As mentioned previously, higher the value of Precision means more appropriate results were returned than inappropriate. And high value of Recall means most of the appropriate results were returned.

Chart represented in figure 5.3 illustrates the results based on Precision and Recall

Figure 5.3 Precision-Recall charts



Based on the definition of Precision in section 5.1, AFLogical gave the most appropriate results than irrelevant.
And, from definition of Recall in section 5.1, ADB gave most the appropriate results.

On comparison of the results obtained from Accuracy and Precision-Recall test, ADB's performance was outstanding amongst all the four tools. With accuracy value of 0.739 ADB leads the other three tools. Then comes the Autopsy with accuracy value of 0.653. AFLogical was the best tool when tested on parameters of 'top' command. But in the validation testing AFLogical stands third in accuracy. Scalpel for all the cases, whether it is data extraction or CPU utilization it was lagging behind the other tools.

5.2 Nonparametric Statistical tests

Nonparametric tests are done when test assumptions do not demand for data belonging to any particular distribution. Since a lesser number of assumptions are made, they are more broadly valid than the parametric tests. This feature also makes them Robust. Such tests are easier to compute and can handle data measured on minimal scale.

5.2.1 CHI Square (X^2) Test [22] [23]

CHI Square is most substantial and practiced test from the family of nonparametric tests. CHI Square test is used to test the difference between the results from a study and the result expected. It tests the difference between distributions of categorical variables. In this report CHI Square test is done to test the difference between the actual results counted manually and the results obtained from the tools.

Null hypothesis

The data extracted using the forensic tools is accurate to successfully predict the outcome of a forensic investigation.

Computation Equation

$$X^2 = \sum \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency}$$

$$X^2 = \sum \frac{(Fo - Fe)^2}{Fe}$$

Degrees of Freedom = $d_f = (number\ of\ levels) - 1$

CHI Square can calculate the difference between the distributions of categorical values. Since all the four tools had their own unique features, therefore CHI Square was performed based on the individual independent variables recorded for all the four tools. Refer tables 4.2, 4.3, 4.5, 4.6, 4.8, 4.9, 4.11, 4.12 for the values used in the test.

Independent variables based on the features of ADB and Autopsy:

Table 5.2 Independent Variable I for CHI Square test

	Independent Variables
1	Files Recovered
2	Time Last accessed
3	Date Created
4	Facebook details
5	YouTube details
6	Bluetooth transfer details
7	Wi-Fi usage details
8	Drop box upload info
9	Email information

Independent variables based on features of AFLogical and Scalpel:

- Video file,
- Jpeg,
- Docx,
- Java,
- Pdf,
- Zip

5.2.1.1 Steps in calculation of the values of the CHI Square X^2:

Observed Responses = F_o, Expected Responses = F_e

$$Fo - Fe \rightarrow (Fo - Fe)^2 \rightarrow \frac{(Fo-Fe)^2}{Fe} \rightarrow \sum \frac{(Fo-Fe)^2}{Fe}$$

If $X^2_{.05} > \sum \frac{(Fo-Fe)^2}{Fe}$ then null hypothesis is accepted.

Values of Fo and Fe calculated for ADB and Autopsy based on ranks (Refer table 5.2)

ADB	1	2	3	4	5	6	7	8	9
Fe	17	2	2	2	1	4	2	1	4
Fo	17	1	1	1	0	4	2	1	3

Autopsy	1	2	3	4	5	6	7	8	9
Fe	17	2	2	2	1	4	2	1	4
Fo	17	0	0	1	0	4	2	1	2

Values of Fo and Fe calculated for Scalpel and AFLogical

Scalpel	Video	Jpeg	Docx	Pdf	Java	Zip
Fe	1	8	3	2	2	1
Fo	1	6	3	2	0	1

AFlogical	Video	Jpeg	Docx	Pdf	Java	Zip
Fe	1	8	3	2	2	1
Fo	1	8	0	1	0	0

Based on the steps given in 5.2.1.1, table 5.3 was populated.

Table 5.3 Test results	ADB	Autopsy	Scalpel	AFLogical
CHI Square Value	2.75	6.5	2.5	6.5
Degrees of freedom, d_f	8	8	5	5
$X^2_{.05}$ based on d_f	15.51	15.51	11.07	11.07
Null hypothesis	True	True	True	True

From the results of table 5.3, Null hypothesis for all the four tools comes out to be true and can be accepted. (Refer 5.2.1 for Null hypothesis). Based on the independent variables the degrees of freedom (d_f) are chosen. The value of d_f corresponding to CHI $Square_{0.5}$ is checked and as explained before if the value of $X^2_{.05}$ > CHI Square value then the null hypothesis is accepted. From table 5.3, the CHI Square value for: - ADB is 2.75 < 15.51 ($X^2_{.05}$ based on d_f), Autopsy is 6.5 < 15.52 ($X^2_{.05}$ based on d_f), Scalpel is 2.5 < 11.07 ($X^2_{.05}$ based on d_f) and AFLogical is 6.5 < ($X^2_{.05}$ based on d_f).
And hence it can be accepted that the data extracted using the forensic tools is accurate to successfully predict the outcome of a forensic investigation.

5.3 Results Collation

Table 5.4

Comparison Chart	Tools			
Features	**ADB**	**Autopsy**	**Scalpel**	**AFLogical**
File recovery – Video	✓	✓	✓	✓
Jpeg	✓	✓	✓	✓
PDF document	✓	✓	✓	✓
Java files	✓	✓	×	×
Zip File	✓	✓	✓	×
Docx files	✓	✓	✓	×
File attributes	✓	✓	✓	✓
Time last accessed	×	×	✓	✓
Date created	✓	×	✓	✓
Application data recovery – Facebook, YouTube, Gmail, Dropbox	✓	✓	×	×
Bluetooth transfer details	✓	✓	×	×
Wi-Fi usage Details	✓	✓	×	×
CPU utilization	**IV**	**II**	**III**	**I**
RAM utilization	**III**	**IV**	**II**	**I**
Load Average on CPU	**II**	**III**	**IV**	**I**
Accuracy Test	**I**	**II**	**IV**	**III**
Precision-Recall test	**I**	**II**	**IV**	**III**
CHI Square Test	**True**	**True**	**True**	**True**

Table 5.4, summarizes the entire emblem of the tool's performance. All the parameters mentioned in the table were chosen to critically evaluate the performance of the tools. Fields are marked with a tick or a cross, which means that, either the tool successfully recovered all the information about that particular field or it did not recover anything. One point to be noted is that, even when the tool was not able to retrieve all the information about the file, it is still marked as a tick for partial data recovery. Only the cases when tool was not able to recover anything are marked as cross.
Going further, there are fields marked as I, II, III, IV. These fields represent the performance of the tools in descending order. For example, in case of CPU utilization, AFLogical is marked as I; this means AFLogical was best among the four tools in the case of CPU utilization. Finally for the CHI Square test, all the fields are marked as true. This explains that the null hypothesis chosen for all the four tools were accepted. (Refer section 5.2.1 for null hypothesis).

5.4 Summary

From the results of Accuracy test, ADB was the best tool in data recovery and information retrieval. Precision Recall test manifested that; ADB was again the better tool than others. Finally from the CHI Square test it can be inferred that the null hypothesis chosen at start of this chapter was true for all the four tools. And tools were adept enough to provide sufficient information about the evidence required to decide the outcome of an investigation. Finally the a grand diagram about the tools comparison is made. This diagram consists of the results obtained from the data acquisition, analysis and validation test.

Conclusion 6

This research project was based on analyzing the efficiency of Android forensic tools. This was achieved by evaluating the information retrieved from Android devices. Four open source tools were used to test two Android devices. All the experiments and result evaluation were based on two test cases. The actual data on the device counted manually was used as a baseline to compare with the data retrieved using the tools. The results were evaluated on various different parameters like extraction capabilities, data recovery, file attributes, tool's effect on CPU etc. Lastly the validation tests were performed to validate the results obtained from all the four tools. From the results of all these tests performed, it was found that ADB was better tool amongst the all four tools when speaking in terms of data extraction and efficiency. But when viewed from the aspect of CPU utilization, RAM utilization etc., the results of ADB were not very exceptional. A tool has to be efficient in all the areas whether it is data extraction or resource utilization. AFLogical on contrary was not very efficient in data acquisition but topped the list when check from the point of resource utilization. Autopsy stood second amongst the four tools, both in terms of acquisition and resource utilization. Scalpel was the least acceptable and efficient tool for any of the above cases discussed. To conclude, ADB is a tool, which can be safely used to perform forensic investigations and can be trusted by forensic professional when deciding the outcome of an investigation.

6.1 Achieving the Objectives

Following objectives were outlined at the start of this project –

- To research about and understand the area of Android forensics
- Distinguish the best approaches presently used for Android forensics
- Define assessable tests, and record the expected outcomes to be obtained from forensic tools
- Detail the steps used for implementing the tools and record the results
- Set a standard to describe the results of forensic tools on Android phones. Critically analyze the tools based on the standard set and actual data recorded manually during experiments.

I. *To research about and understand the area of Android forensics*

A comprehensive research about the Android platform and the basics needed to perform the investigations using forensic tools is reviewed. Review about the Android architecture to get the understanding of logical analysis. Different versions of Android platforms were discussed. A review on Android SDK was performed to assist during forensic investigations. Procedures used to support the forensic process, like USB debugging, Android rooting etc. A brief discussion was done about the features of powerful tool android debug bridge. Forensic analysis in this research project focuses on analyzing file systems. A review was done about the Android file systems and data storage locations. Finally a brief discussion was done about the Forensic tools used in this research project and the methodology used to evaluated these tools. Chapter one completes the objective one.

II. *Distinguish the best approaches presently used for Android forensics*

Reviewing the techniques and methods used by Government and international forensic groups like NIST and ACPO completed this objective. NIST [26] and ACPO [27] were identified and studied.

Also practices by viaForensics [1] was distinguished and reviewed. Besides the standard methods used by these international agencies, some other methods which are not as efficient as standard ones were also studied.

III. *Define assessable tests, and record the expected outcomes to be obtained from forensic tools*

This part of the objective was met in chapter two. Details about the device specification and development machine used were presented. Two test cases were defined to conduct the experiments and methodology used to implement the test cases was explained. To record the results from forensic tools, a data collection map was made to depict the steps that will be used to collect the data from the tools and analyze it.

IV. *Detail the steps used for implementing the tools and record the results*

Discussion about the choice of logical techniques over physical analysis techniques was made. All the steps needed in implementing the tools were detailed. In order to improve the understanding of the steps used in tool implementation, visual representation with the help of flow-charts and pie charts were used. Chapter three completes the objective IV.

V. *Set a standard to describe the results of forensic tools on Android phones. Critically analyze the tools based on the standard set and actual data recorded manually during experiments.*

Actual data acquired manually was used as standard to compare the data extracted using the tools. Results from all the tools were tabulated to give a visual representation of the comparison made. Various parameters other than the data extracted were measured. For further critical analysis of the tools, Validation tests were performed to measure the accuracy of the data extracted from the tools. Chapter 4 and Chapter 5 complete the last objective.

6.2. Critical Analysis

When proceeding with the project a few problems were faced. Since Android forensics is a developing field in Digital forensics, the first problem was to find the relevant literatures relating to this project. Amongst a large number of papers and articles found only a few were having some academic value.

Another problem faced with this project was finding a significant assessment structure. In the beginning the goal was set to evaluate the Hardware Forensic tools to test the android devices. But the tools were very expensive and license required to use them was not granted to non-forensic professionals. The project had to switch from hardware forensic tools to open source free software forensic tools.

6.3 Future Work [28] [9]

The future work for this project will be based on the newer versions of the Android operating system. Studied has to be done to check whether the file system used by the Android devices is still the same. And also, can the forensic examiner use same methodology as proposed in this report for the future versions of Android operating system. The technique presented in this report to root the device can be more advanced with less possibilities of bricking the device. Also new methods to bypass the phone lock/pattern will be a great step towards future work.

The production of high-end smart phones poses a high impact on digital forensic investigation, and hence requires a better analysis on a device-by-device basis. The new generation mobile phones and

operating systems are becoming largely independent of the hardware. This can help with data extraction and examination by lessening the need to learn about the different devices and the platform they are operating on.

The requirement to reverse-engineer the peculiarities discovered in traditional operating systems or discrete implementations will drop over time. Reverse engineering of third party applications can decide the future of mobile phone forensics. These applications are platform-dependent and written with native SDKs or tool sets of the third party. This learning of capability of the third-party application can be fundamental to forensic examiners in some cases, as they can be malicious in behavior, can allow crime indirectly, or can even store the valuable forensic data. These third-party applications are also doorways to cloud services, which are largely accommodating mobile devices.

Present connections among workstations, notebook PCs and phones looks phone as a satellite device. The advancements in connectivity between devices and other sources of data can amend this and allow improved links between connected systems. To conclude, the trend will always continue that the mobile device will remain an important part in forensic examinations and have a huge contribution in criminal investigations.

Bibliography

1) Andrew Hoog (2011). *Android forensics*. 225 Wyman Street, Waltham, MA 02451, USA: Elsevier. 1-379.
2) Vijith Vijayan. (April 2012). Android forensic capability and evaluation of extraction tools. Available: http://napier.academia.edu/vichiee/Papers/1702107/Android_Forensic_Capability_and_Evaluation_of_Extraction_Tools. Last accessed 15 September 2012.
3) Jack wallen. (feb 25, 2011). *Using the Linux df command*. Available: http://www.ghacks.net/2011/02/25/using-the-linux-df-command/. Last accessed 4 september 2012.
4)Android (2012). Installing the SDK. Available: http://developer.android.com/sdk/installing/index.html. Last accessed 11 september 2012.
5) Chezi-Schlaff. (26 September 2012). Rooting (Android OS). Available: http://en.wikipedia.org/wiki/Rooting_(Android_OS). Last accessed 17 August 2012.
6) CL ShortFuse. (2010). *Super one click*. Available: http://forum.xda-developers.com/showthread.php?t=803682. Last accessed 7 august 2012.
7) Timothy Vidas, Chengye Zhang, Nicolas Christin. (2011). Toward a general collection methodology for Android devices. *Digital Investigation*. 8 (-), S14 - S24.
8) Brian Carrier . (2003). *The Sleuth Kit Informer*. Available: http://www.sleuthkit.org/informer/sleuthkit-informer-1.txt. Last accessed 19 september 2012.
9) Panagiotis Andriotis, George Oikonomou, Theo Tryfonas. Theo.Tryfonas@bristol.ac.uk. Forensic Analysis of Wireless Networking Evidence of Android Smartphones. July 7 2012.
10) Shadi Dibbini, Nick Tate, Rob Unger, Jordan Rogers . (2012). *Open source Android Forensics*. Available: http://osaf-community.org/home.html. Last accessed 7 august 2012.
11) Andrew Hoog. (2011). Android forensic techniques. In: Andrew Hoog *Android forensics*. 225 Wyman Street, Waltham, MA 02451, USA: Elsevier. 195-364.
12) Cindy Murphy. (2011). *Cellular Phone Evidence Data Extraction and Documentation*. Available: http://www.ericjhuber.com/2011/08/detective-cindy-murphys-cell-phone.html. Last accessed 27 August 2012.
13) Golden G. Richard III. (2011). *Scalpel: A Frugal, High Performance File Carver*. Available: http://www.digitalforensicssolutions.com/Scalpel/. Last accessed 22 September 2012.
14) Falko Timme . (2009). *Recover Deleted Files With Scalpel*. Available: http://www.howtoforge.com/recover-deleted-files-with-scalpel. Last accessed 29 August 2012.
15) Daryl (2011). *scalpel.conf*. Available: https://github.com/int0x80/anti-forensics/blob/master/scalpel.conf. Last accessed 13 August 2012.
16) Golden G. Richard III Vassil Roussev. (2005). Scalpel: A Frugal, High Performance File Carver. *Scalpel*. 1 (-), 1-10.
17) Andrew Hoog . (2012). *AFLOGICAL™* . Available: https://viaforensics.com/products/android-forensics-tool/. Last accessed 9 September 2012.
18) Andrew Hoog . (2012). *VIAEXTRACT*. Available: https://viaforensics.com/products/viaextract/. Last accessed 10 September 2012.
19) Andrew Hoog . (2012). *AFLogical™ Open Source Edition Now Available for Download*. Available: https://viaforensics.com/products/tools/aflogical/. Last accessed 11 September 2012.
20)wikipedia. (2007). *Precision and recall*. Available: http://en.wikipedia.org/wiki/Precision_and_recall. Last accessed 19 September 2012.
21) Bruce H. Bailey and Scott L. McDonald . (1997). DATA VALIDATION, PROCESSING, AND REPORTING. In: - *Wind Resource Assessment Handbook*. CESTM, 251 Fuller Road Albany, NY 12203: AWS Scientific,. 9.1 - 9.15.
22)James P. Key. (2010). *CHI Square*. Available: http://www.okstate.edu/ag/agedcm4h/academic/aged5980a/5980/newpage28.htm. Last accessed 24 September 2012.

23)Preacher, K. J.. (2001). *Calculation for the Chi-Square Test*. Available: http://www.quantpsy.org/chisq/chisq.htm. Last accessed 23 September.
24) Georgina Enzer. (2012). *Global mobile penetration hits 85%*. Available: http://www.itp.net/588064-global-mobile-penetration-hits-85. Last accessed 18 August 2012.
25) Bruce Schneier. (2006). *Remotely Eavesdropping on Cell Phone Microphones*. Available: http://www.schneier.com/blog/archives/2006/12/remotely_eavesd_1.html. Last accessed 19 August 2012.
26) Wayne Jansen Rick Ayers. (2007). Guidelines on Cell Phone Forensics. *Recommendations of the National Institute of Standards and Technology*, 1-104.
27) "Good practice guide for computer based electronic evidence," Association of Chief Police Officers.
28) Eoghan Casey, Benjamin Turnbull. (2011). Digital Evidence on Mobile Devices. In: Eoghan Casey Digital Evidence and Computer Crime. 3rd ed. Baltimore, USA: Elsevier. 1-44.
29) Eoghan Casey, Aaron Stanley, Tool review – remote forensic preservation and examination Tools, Digital Investigation, Volume 1, Issue 4, December 2004, Pages 284-----297, ISSN 1742-----2876, 10.1016/j.diin.2004.11.003.
30) Barrie Mellars, Forensic examination of mobile phones, Digital Investigation, Volume 1, Issue 4, December 2004, Pages 266-272, ISSN 1742-2876, 10.1016/j.diin.2004.11.007.
31) Barrie Mellars, Forensic examination of mobile phones, Digital Investigation, Volume 1, Issue 4, December 2004, Pages 266-----272, ISSN 1742-----2876, 10.1016/j.diin.2004.11.007.
32) Yinghua Guo, Jill Slay, Jason Beckett, Validation and verification of computer forensic software tools— Searching Function, Digital Investigation, Volume 6, Supplement, September 2009, Pages S12-S22, ISSN 1742-2876, 10.1016/j.diin.2009.06.015.
33) Rick Ayers, Wayne Jansen, Ludovic Moenner, Aurelien Delaitre. (2007). Cell Phone Forensic Tools:. An Overview and Analysis Update. NISTIR 7387 (1), 1-153.
34) Lodovico Marziale, Golden G. Richard III, Vassil Roussev, Massive threading: Using GPUs to increase the performance of digital forensics tools, Digital Investigation, Volume 4, Supplement, September 2007, Pages 73-----81, ISSN 1742----2876, 10.1016/j.diin.2007.06.014.
35) Simson Garfinkel, Digital forensics XML and the DFXML toolset, Digital Investigation, Volume 8, Issues 3–4, February 2012, Pages 161----174, ISSN 1742----2876, 10.1016/j.diin.2011.11.002.
36) Emmanuel S. Pilli, R.C. Joshi, Rajdeep Niyogi, Network forensic frameworks: Survey and research challenges, Digital Investigation, Volume 7, Issues 1–2, October 2010, Pages 14---27, ISSN 1742-----2876, 10.1016/j.diin.2010.02.003.
37) Richard III GG, Roussev V. Scalpel: a frugal, high-performance file carver. In: Proceedings of the 2005 digital forensics research workshop (DFRWS 2005).
38) Maynard Yates, Hongmei Chi: A Framework for Designing Benchmarks of Investigating Digital Forensics Tools for Mobile Devices, Volume 1, March 2011, Pages 179-184, ISBN: 978-1-4503-0686-7 doi>10.1145/2016039.2016088.
39) Charles Arthur. (2012). *Android over 50% of Smartphone sales*. Available: http://www.guardian.co.uk/technology/2012/may/16/android-smartphone-market-50-percent. Last accessed 24 September 2012.
40) *Android 2.0 Platform Highlights*. Available: http://developer.android.com/about/versions/android-2.0-highlights.html. Last accessed 12 September 2012.