

Smartphone Forensic Analysis: A Case Study for Obtaining Root Access of an Android Samsung S3 Device and Analyse the Image without an Expensive Commercial Tool

Muhammad Faheem¹, N.-A. Le-Khac², Tahar Kechadi¹

¹University College Dublin, Dublin, Ireland

²Department of Computer Science, Abdul Wali Khan University Mardan, Mardan, Pakistan

Email: faheemkhalil@gmail.com

Received 30 April 2014; revised 25 May 2014; accepted 20 June 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Smartphone is a very useful and compact device that fits in person's pocket, but at the same time it can be used as a tool for criminal activities. In this day and age, people increasingly rely on smart phones rather than desktop computers or laptops to exchange messages, share videos and audio messages. A smartphone is almost equivalent in its application to a PC, hence there are security risks associated with its use such as carrying out a digital crime or becoming a victim of one. Criminals can use smartphones for a number of activities. Namely, committing a fraud over e-mail, harassment via text messages, drug trafficking, child pornography, communications related to narcotics, etc. It is a great challenge for forensic experts to extract data from a smartphone for forensic purposes that can be used as evidence in the court of law. In this case study, I show how to obtain the root access of Samsung S3 phone, how to create DD image and then how to examine DD image via commercial tool like UFED physical analyzer trial version which doesn't support Android devices? I will extract the messages for Viber on trial version of UFED Physical analyzer.

Keywords

Viber, Root, Android Forensic

1. Introduction

According to NIST, the current definition of digital forensics is the scientific procedures used to recognize and

classify, collect, evaluate and analyse the data while maintaining the level of integrity of the information throughout the forensics process.

Figure 1 below is showing different fields of the Digital Forensics.

Computer forensics is the process of obtaining, identifying, extracting, analysing, and documenting of computer evidence stored as data/digital/magnetically encoded information for use as evidence in civil, administrative and criminal cases [1].

Database forensics is the study of database and their metadata. Database forensics use database contents and log files in order to retrieve the relevant information.

Network forensics is an analysis of network traffic. Network forensics allows us to make forensic determinations based on the observed traffic of the network [2].

Mobile forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions [3].

According to NIST, the mobile forensics is a process of preservation, acquisition, examination and analysis, followed by reporting [4].

1.1. Android Smartphone Growth

In this digital age, smartphones are integral part of our communication as they become more similar in use to desktop computers. We no longer make only phone calls, and send and receive text messages, but also use them for social networking, online banking, buying and selling goods online, watching news and movies, playing games, etc. Nowadays more commercial and non-commercial businesses deploy their custom-made applications for smartphones, which allow employees and customers to download usable data on smartphones.

There are many kinds of smartphone operating systems available on the market, *i.e.* Android, IOS and RIM. Google's Android operating system is one of the most popular OS for smartphones, television, gaming devices and notebooks. In the first quarter of 2013, smart mobile phone shipment exceeded 300 million and Android accounted for 64% of total sales of all smartphones [5].

Figure 2 underneath is presenting a range of top selling smartphone brands in the first quarter of 2013.

Figure 3 is the projected sales growth for most common used OS according to Gartner.

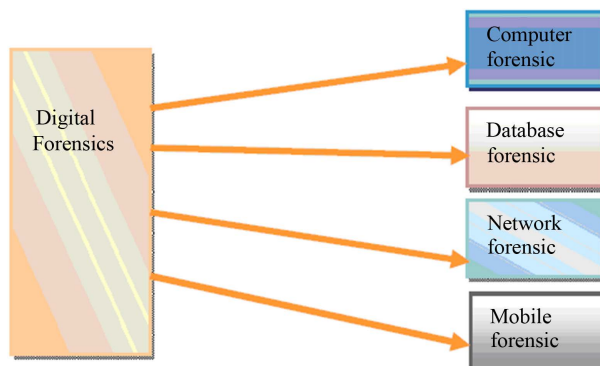


Figure 1. Digital forensics fields.

| GB | |
|---------------------------------------|------|
| Top selling smartphone models Q1 2013 | % |
| Apple iPhone 5 | 15 |
| Samsung Galaxy S III | 11.4 |
| Apple iPhone 4S | 7.5 |
| Samsung Galaxy S III Mini | 5.7 |
| Samsung Galaxy Ace | 5.4 |
| Apple iPhone 4 | 4.9 |
| LG Google Nexus 4 | 3.6 |
| Samsung Galaxy Ace 2 | 3.4 |
| Samsung Galaxy S II | 3.4 |
| BlackBerry Curve 9320 | 2 |

Figure 2. Top selling models.

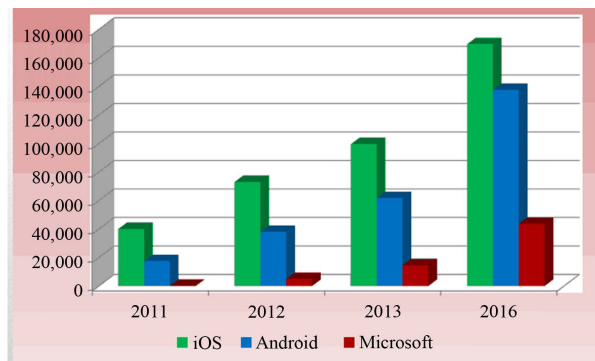


Figure 3. Sales growth graph.

1.2. Android OS Architecture

Android is an open source platform for smartphones. The term “Android” is a Greek word which means “human being”. Android is a software stack for mobile devices that includes an operating system, middleware and key applications [6].

In order to compete with Apple iOS system, Google acquired company called Android which developed an operating system with a consideration for the device for which it was created. This operating system provides users with visibility into how applications work, allows control over those applications and provides security against any malware attack.

In order to perform forensic analysis of Android system it is important to understand Android’s architecture and its core components. The basic Android architecture is Linux Kernel, and it is composed of five main components. These are presented in Figure 4 [7].

2. Related Work

The most relevant research that was close to my case studies was done by “Forensic Analysis of Instant Messenger Applications on Android Devices” Mahajan, Aditya, M. S. Dahiya, and H. P. Sanghvi (2013).

In this research, they use the commercial UFED Physical Analyzer which cost around eight thousand Euro to acquire data from mobile phone. With the commercial UFED physical Analyzer data extraction was done logically. After data extraction it was possible to get artifacts and timestamp of Viber. In this research, if the investigators want to examine the deleted data they won’t be able to do analysis because data acquisition was done logically it didn’t create bit by bit copy of the phone internal memory.

In my case I should be able to examine deleted data if use commercial forensics tools.

3. Case Study

In Previous cases, they fail to explain how we can obtain forensically sound image without access to commercial tool. In this case study, I have task to obtain image of suspected Android devices which is Samsung S3 phone and Viber was used as mean of communication between suspects. Below is the specification of the mobile phone:

- CPU: Quad-core 1.4 GHz Cortex-A9
- Memory 16 GB
- Android OS 4.1.2
- Model No GT-19300

My first task is to obtained root access of this mobile phone so I can create image which I can examine then using UFED Physical Analyzer.

3.1. A Root Access

- 1) First I need to install Android Development Tool (ADT) which is part of Android Software Development Kit (SDK) on my windows machine from <http://developer.android.com/sdk/index.html>, which is zipped file. Next I extract the file on C drive (copy all the files from folder “Platform Tools” into folder “Tools”).

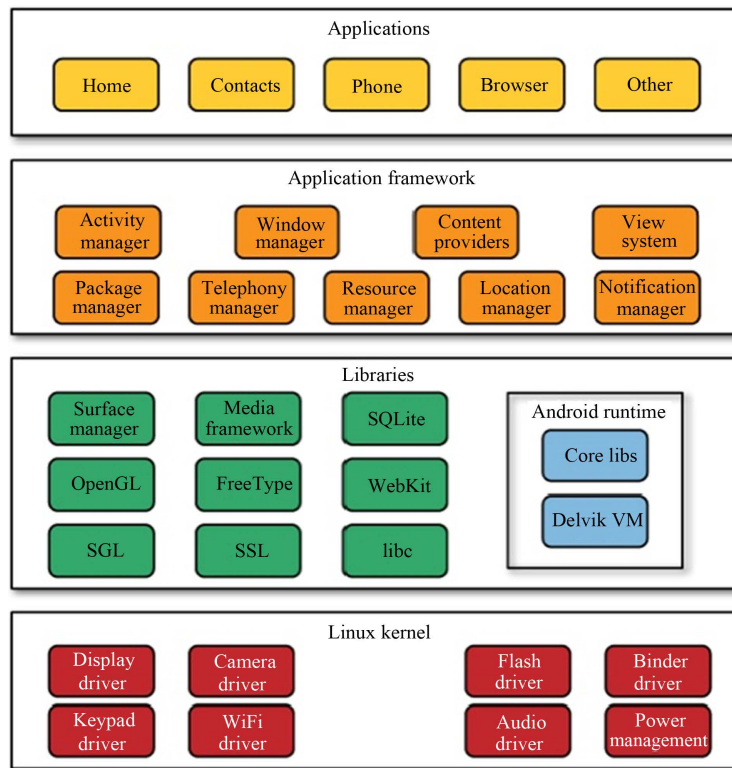


Figure 4. Android OS architecture.

- 2) To gain access to the root directory I need to enable USB debugging on the phone, which is achieved by selecting Settings and then Developer options. From the latter Debugging has to be clicked and then check box USB debugging has to be selected. Once a warning message appears, Ok has to be clicked.
- 3) The next step is a command prompt command which ensures that ADB is working properly. To execute this command a phone must be connected via USB cable to the laptop computer as shown in [Figure 5](#).
- 4) I have installed SRSRoot from <http://www.srsroot.com/> on my machine in order to gain the root access. Next I used a set of commands:

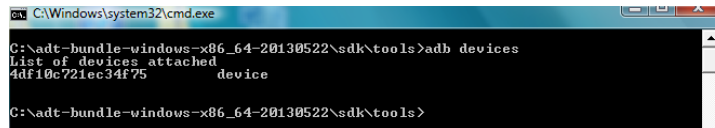

```
$ adb push /SRSRoot /data/local/tmp
$ adb shell
$ ls to see the list of directory
$ chmod 777 <filename> (chmod 777 change file permissions gives everyone permission (read/write/execute) to that file)
$. /<filename> (./ is used to execute a file)
Then su
And now
# adb shell (now the $ is replaced by # indicating that you are root) as show in Figure 6.
```
- 5) After gaining the root permission, I am able to create image of an Android device.

3.2. Creating DD Image of Memory

The Android file system is divided into number of partitions. Without a traditional hard drive, it uses Memory Technology Devices (MTD) to make connections between Linux Kernel and flash drive. The most common partitions in the Android system are boot, cache, data, and recovery.

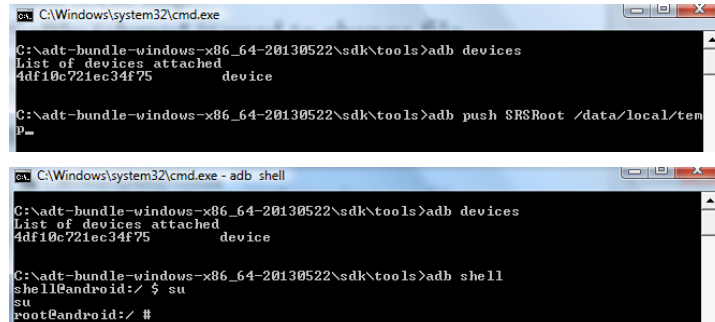
After gaining the root access to the file system of the Android Samsung Galaxy S3 phone, I used a DF command shown in [Figure 7](#) followed by a mount command [Figure 8](#) to display the partitions. Next I inserted a fresh formatted external 30 GB SD card into the phone.

[Figure 7](#) underneath presents the outputs from these two commands ([Figure 8](#)).



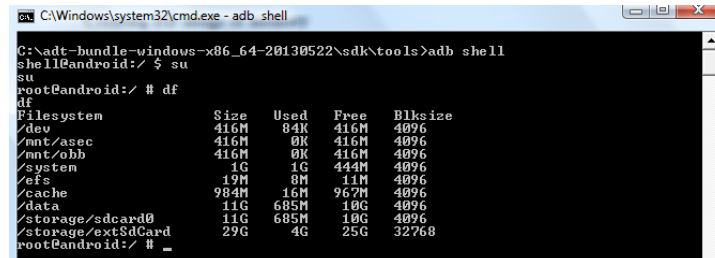
```
C:\Windows\system32\cmd.exe
C:\adt-bundle-windows-x86_64-20130522\sdk\tools>adb devices
List of devices attached
4df10c721ec34f75    device
C:\adt-bundle-windows-x86_64-20130522\sdk\tools>
```

Figure 5. Above shows that device is connected to the ADB.



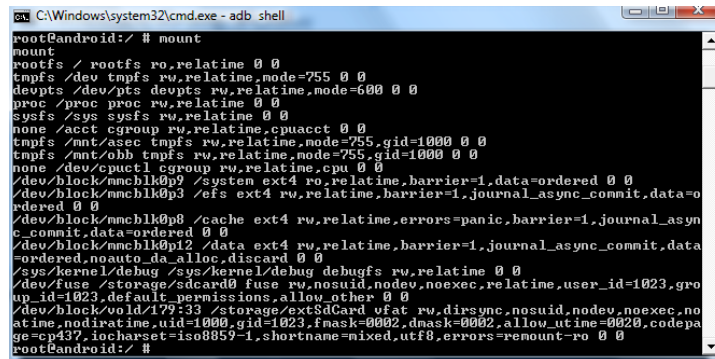
```
C:\Windows\system32\cmd.exe
C:\adt-bundle-windows-x86_64-20130522\sdk\tools>adb push SRSRoot /data/local/tmp/
P=
C:\adt-bundle-windows-x86_64-20130522\sdk\tools>adb shell
shell@android:/ $ su
su
root@android:/ #
```

Figure 6. Showing root access successfully obtained.



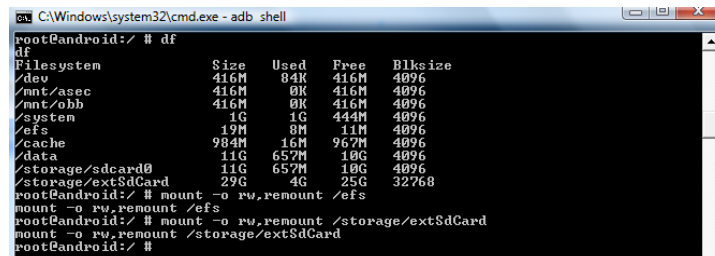
```
C:\Windows\system32\cmd.exe - adb shell
C:\adt-bundle-windows-x86_64-20130522\sdk\tools>adb shell
shell@android:/ $ su
su
root@android:/ # df
df
Filesystem      Size      Used    Free   Blksize
/dev             416M      84K    416M    4096
/mnt/asec       416M      0K    416M    4096
/mnt/obb        416M      0K    416M    4096
/system         1G        1G    444M    4096
/efs            19M        8M     11M    4096
/cache          984M      16M    967M    4096
/data           11G      685M    10G    4096
/storage/sdcard0 11G      685M    10G    4096
/storage/extSdCard 29G        4G    25G   32768
root@android:/ #
```

Figure 7. Output of DF command.



```
C:\Windows\system32\cmd.exe - adb shell
root@android:/ # mount
mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
none /acct cgroup rw,relatime,cpuacct 0 0
tmpfs /mnt/asec tmpfs rw,relatime,mode=755,gid=1000 0 0
tmpfs /mnt/obb tmpfs rw,relatime,mode=755,gid=1000 0 0
none /dev/cpuctl cgroup rw,relatime,cpu 0 0
/dev/block/mmcblk0p9 /system ext4 rw,relatime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p3 /efs ext4 rw,relatime,barrier=1,journal_async_commit,data=ordered 0 0
/dev/block/mmcblk0p8 /cache ext4 rw,relatime,errors=panic,barrier=1,journal_async_commit,data=ordered 0 0
/dev/block/mmcblk0p12 /data ext4 rw,relatime,barrier=1,journal_async_commit,data=ordered,noauto_da_alloc,discard 0 0
/sys/kernel/debug /sys/kernel/debug debugfs rw,relatime 0 0
/dev/fuse /storage/sdcard0 fuse rw,nosuid,nodev,noexec,relatime,user_id=1023,group_id=1023,default_permissions,allow_other 0 0
/dev/block/void179:33 /storage/extSdCard vfat rw,dirsync,nosuid,nodev,noexec,relatime,uid=1000,gid=1023,fsck=0002,dmaback=0002,allow_utime=0020,codepage=cp437,iocharset=iso8859-1,shortname=mixed,utf8,errors=remount-ro 0 0
root@android:/ #
```

(a)



```
C:\Windows\system32\cmd.exe - adb shell
root@android:/ # df
df
Filesystem      Size      Used    Free   Blksize
/dev             416M      84K    416M    4096
/mnt/asec       416M      0K    416M    4096
/mnt/obb        416M      0K    416M    4096
/system         1G        1G    444M    4096
/efs            19M        8M     11M    4096
/cache          984M      16M    967M    4096
/data           11G      657M    10G    4096
/storage/sdcard0 11G      657M    10G    4096
/storage/extSdCard 29G        4G    25G   32768
root@android:/ # mount
mount -o rw,remount /efs
root@android:/ # mount -o rw,remount /storage/extSdCard
mount -o rw,remount /storage/extSdCard
root@android:/ #
```

(b)

Figure 8. (a) Output of mount command; (b) Command for obtaining read write access.

rootfs-Kernel mounts the Root File System at Startup
 devpts-simulated terminal sessions
 Proc-information about Kernel, processes and configuration
 tmpfs-most important—this is RAM (stored on a different chip)
 cramfs-compressed ROM file system
 Next I will be creating a DD image of the partitions:

- Data
- System
- Cache
- EFS—contains sensitive information like Mac address, IMEI, product code, wireless

Before DD command I make it sure to give read/write permission to each partition which is as follows:
 mount -o rw, remount /xxxxx where xxxxx = partition name

Example of DD commands

```
DD if = /dev/block/mmcblk0p12/data of = /storage/extSdCard/data.dd
DD if = /dev/block/mmcblk0p8/cache of = /storage/extSdCard/cache.dd
DD if = /dev/block/mmcblk0p3/efs of = /storage/extSdCard/efs.dd
DD if = /dev/block/mmcblk0p09/system of = /storage/extSdCard/system.dd
```

Efs image **Figure 9**.

Data image **Figure 10**.

The DD command was executed successfully and output was saved onto the external SD drive, and then I used the command below to move the output from SD card to the host machine.

ADB pull/sdcard/xxxxx/pathto download

Where xxxxx = image and pathto download is where I want to save it on host machine.

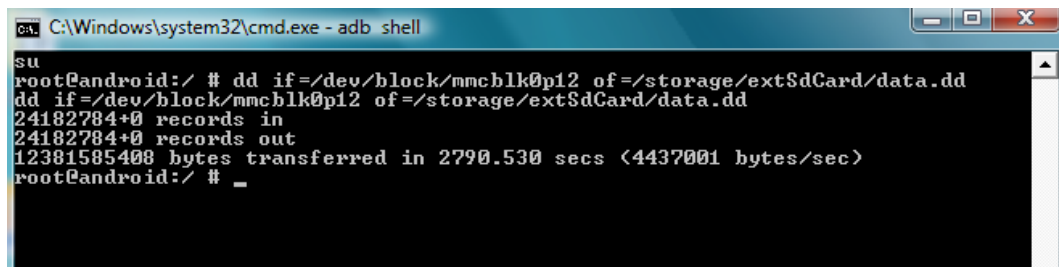
4. UFED Physical Analyzer Examination

I need to install trial version of UFED Physical Analyzer which is free to use for maximum one month, but it doesn't support any Android phones. But I have obtained the image of the phone and I can use UFED Physical Analyzer I open a blank project and add source file as my DD image. The UFED Physical Analyzer does recognize this image I can do analysis on this image. My aim in this case study is to find Artifact connected to any viber communication below is screen dump I obtained **Figure 11**.



```
C:\Windows\system32\cmd.exe - adb shell
dd if=/dev/block/mmcblk0p8 of=/sdcard/extSdCard
2097152+0 records in
2097152+0 records out
1073741824 bytes transferred in 238.034 secs (4510875 bytes/sec)
root@android:/ # dd if=/dev/block/mmcblk0p8 of=/storage/extSdCard/efs.dd
dd if=/dev/block/mmcblk0p8 of=/storage/extSdCard/efs.dd
2097152+0 records in
2097152+0 records out
1073741824 bytes transferred in 240.049 secs (4473011 bytes/sec)
root@android:/ #
```

Figure 9. EFS image.



```
C:\Windows\system32\cmd.exe - adb shell
su
root@android:/ # dd if=/dev/block/mmcblk0p12 of=/storage/extSdCard/data.dd
dd if=/dev/block/mmcblk0p12 of=/storage/extSdCard/data.dd
24182784+0 records in
24182784+0 records out
12381585408 bytes transferred in 2790.530 secs (4437001 bytes/sec)
root@android:/ # _
```

Figure 10. Data image.

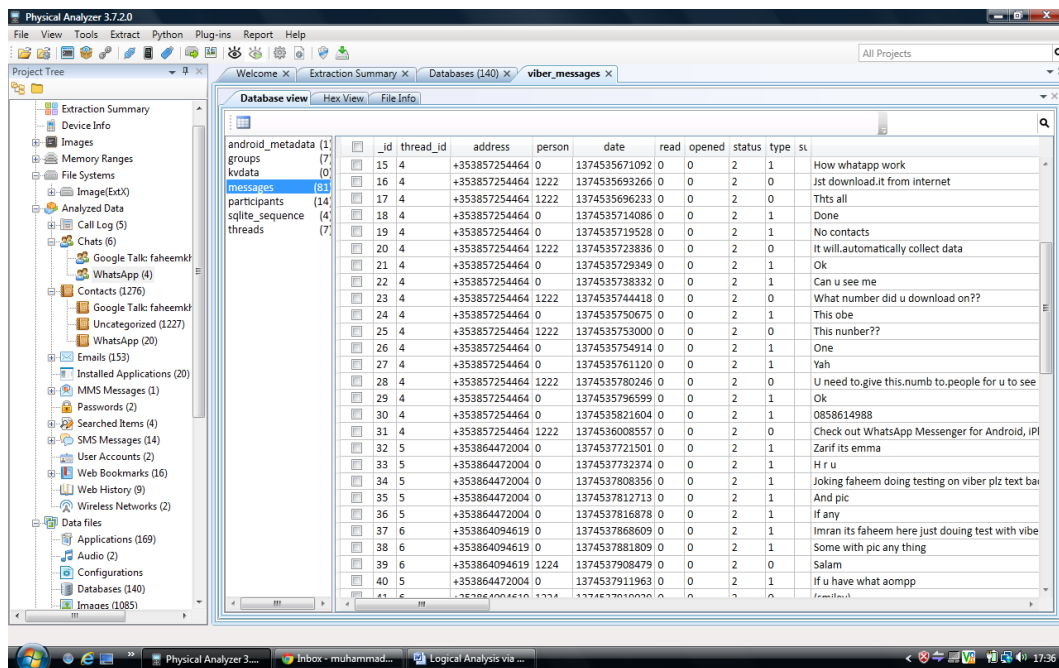


Figure 11. UFED Physical Analyzer screen dump.

Below is table for all Viber related artifacts.

I found most of the Viber artifacts in /data/data/com.viber.voip/ directory, while Viber messages with time stamps in the folder /data/data/com.viber.voip/databases/viber_messages.

SIM number with which the Viber account was activated and the time stamps are in the folder called /data/data/com.viber.voip/files/preference/activated_sim_serial.

The country code which Viber is registered with is in /data/data/com.viber.voip/files/preference/reg_viber_country_code.

First SMS sent via Viber is to be found at /data/data/com.viber.voip/files/preference/viber_first_sms.

The Viber activation code with time stamps is in the folder /data/data/com.android.providers.telephony/databases.

Pictures sent via Viber are in the folder /data/data/com.android.providers.telephony/app_aparts.

5. Conclusion and Future Work

The Android forensics is a relatively new and constantly evolving discipline as a result of new models of the Android phone coming to the market. Some research shows that the Android phone will exceed iPhone sales within next few years.

The above results show we still can work around and do forensic analysis even without full licence copy of the forensic tools. The results obtained in this cases studies are exactly the same or similar if we use expensive commercial tools. In this research, I was particularly focused on how to gain the root access and acquire data from the Samsung Galaxy S3 phone followed by the forensic analysis of data using UFED Physical Analyzer.

This is by no means a conclusive project. In my future research, I would like to further investigate more tools and different application if I am able to fully explore all artifacts using trial version of forensic tools.

References

- [1] Vacca, J.R. (2002) Computer Forensics. Charles River Media, Hingham.
- [2] Cohen, M.I. and PyFlag (2008) PyFlag—An Advanced Network Forensic Framework Communication of the ACM. *Digital Investigation*, 5, S112-S120.
- [3] http://www.webopedia.com/TERM/M/mobile_phone_forensics.html

- [4] Jansen, W.A. (2007) Guideline on Cell Phone Forensics.
- [5] Palo Alto 31st of January 2011. <http://www.kantarworldpanel.com/>
- [6] Available from <http://developer.android.com/about/index.html>
- [7] Available from <http://www.android-app-market.com/android-architecture.html>

Scientific Research Publishing (SCIRP) is one of the largest Open Access journal publishers. It is currently publishing more than 200 open access, online, peer-reviewed journals covering a wide range of academic disciplines. SCIRP serves the worldwide academic communities and contributes to the progress and application of science with its publication.

Other selected journals from SCIRP are listed as below. Submit your manuscript to us via either submit@scirp.org or [Online Submission Portal](#).

