



University of Piraeus  
Department of Digital Systems

Postgraduate Program  
“Security of Digital Systems”

# Android Forensics



*Sakkas Vasileios*

**Supervisor:** *Xenakis Christos,*  
*Assistant Professor,*  
*University of Piraeus*

*Academic Year 2013-2014*

Intentionally blank page

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Table of Contents

Acknowledgments .....	1
Introduction.....	2
Smartphone Security .....	3
Mobile Forensics.....	4
Live Forensics.....	4
Memory Forensics Approaches .....	5
Data Types .....	6
Forensic Process .....	8
Android concepts and components .....	10
Android version history .....	11
Device Types .....	12
Boot Process .....	13
Android SDK and ADB.....	15
File system and partitions .....	18
Data Storage .....	21
Data in Memory.....	22
Rooted Device .....	22
Accessing Data.....	24
Device Access .....	25
Logical Acquisition .....	27
SQLite Databases.....	30
Physical Acquisition .....	33
Other Forensic Techniques.....	42
Anti-forensics.....	44
Conclusion .....	45
References.....	47

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Acknowledgments

At this point I would like to take the opportunity and thank the people who contributed more in this project. Firstly, I acknowledge the help and support of the closest people of mine; my family and my friends.

Furthermore, I specially thank my professor Dr Christos Xenakis for helping me carry out this project of Android Forensics with his aid and advice.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Introduction

Mobile devices are among the most common new technologies of the latest years, gaining even more spread and success in the day-to-day life of people. Smartphones gather all these features to meet several of the wishes of the people and as time runs, they will reach a point to replace personal computers. Smartphones in general are perhaps the one electronic device that knows the most about an individual. For most people, their smartphone is rarely more than a few meters from them at any point of time. This device blends both personal and corporate information and has the ability to store vast amounts of data including text messages, e-mails, GPS locations, picture, videos, and more.

Clearly there is a need for forensics. Mobile device forensics is the science of recovering digital evidence from a mobile device under forensically sound conditions using accepted methods. Mobile device forensics is an evolving specialty in the field of digital forensics. While the number of mobile devices are used in crime activities is spreading and growing all over the world, the capability to perform the forensics analysis of such devices is limited both by technological and methodological problems. Both mobile forensics and Android forensics in particular have a set of challenges that must be overcome. Criminals could use the smartphones for committing fraud over e-mail, harassment through text messages, trafficking of illegal material, communications related to drugs, etc. The data stored on smart phones could be extremely useful to analysts through the course of an investigation. A fundamental goal in digital forensics is to prevent any modification of the target device by the examiner.

The project encompasses all the elements of the Android internals, the data types, the partitions and directories alongside with the procedures of imaging them. The Android concepts and components are analyzed showing the reason that these are necessary for the forensic procedure. The practical part contains the analysis of the device divided in logical and physical examination. This project presents a standardized procedure that covers a large part of device forensics trying to give some serious forensic material.

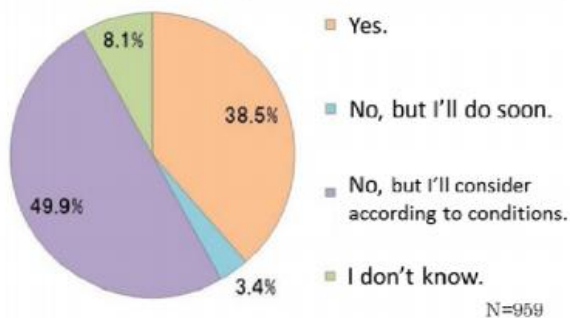
## Smartphone Security

Security covers a lot of territory such as locking a computer room, protecting files with passwords, using encryption in network communication lines etc. The goal is to keep data protected from invasion, destruction or loss. In theory anyone can secure perfectly a system but in practice it is nearly impossible to achieve. Furthermore, security will always be a result of weighing the danger against the costs.

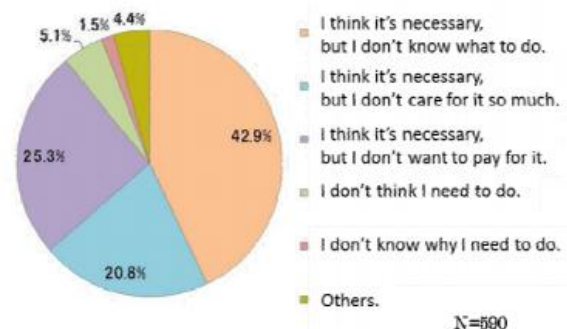
Smartphones consist of a set of factors that make it more difficult to secure these devices. The hardware and software is assembled and maintained by a large group of participants including the manufacturers' and the wireless carriers' hardware and software. Additionally, a mobile device travels a lot through different wireless networks with unknown security level. That leads to an uncertainty about the security of a network area.

NetMile Inc. has conducted a research regarding security issues raised by smart phones and presented the results.

Have you implemented information security measures on your smartphone?



Why don't you implement information security measures on your smartphone?



(NetMile, Inc.)

The research shows that a lot of people still believe that the security of a Smartphone is the same as a conventional mobile phone. Based on this, efforts need to be made to improve user awareness of information security measures.

## Mobile Forensics

The mass growth of the Smartphone devices and the universal import of them in peoples' lives cause the fact that it is rare to conduct a digital forensic investigation that does not include a Smartphone or mobile device. Often, the Smartphone may be the only source of digital evidence tracing an individual's movements and motives and may provide access to the who, what, when, where, why, and how behind a case.

The term "Mobile Forensics" refers to a branch of digital forensics relating to recovery of digital evidence or data from a mobile device or any device with both internal memory and communication ability such as PDAs, GPS computers or tablets. A mobile forensics analyst focuses on Smartphone as sources of evidence, providing the necessary skills to handle mobile devices in a forensically sound manner, understand the different technologies, discover malware, and analyze the results for use in digital investigations by diving deeper into the file systems of each Smartphone.

## Live Forensics

Physical memory analysis is vital to investigations, since it contains a wealth of information that is otherwise unrecoverable. This evidence includes objects relating to both running and terminated processes, open files, network activity, memory mappings, and more. Lack of such information can make certain investigative scenarios impossible, such as when performing incident response or analyzing advanced malware that does not interact with non-volatile storage.

A mobile device stores and processes data as communication information (contacts, chat logs, text messages, multimedia messages, emails etc.), personal data (multimedia files, calendars etc.), location data and data handled by third party applications. Some of this information is stored as live data in the system's main memory and the latter is stored in non-persistent memory such as RAM which is typically lost when a device is shut down. The live data are also called *volatile* and the data stored in persistent memory are *non-volatile*. The types of data will be discussed



further below. Hence, a forensic investigation on volatile data or *live memory forensics* analyzes the dynamic behavior of the mobile phone's volatile memory.

Live means the focus on a system's current state. A live forensics analyst's work depends on a snapshot of a chosen or closing state of the system. This type of investigation is based on the procedure of acquiring a copy of the main memory. The main memory contains the whole state of an operating system, including running and historical processes, open network connections, management data etc. Usually the image of the acquired volatile memory is stored on a non-volatile storage to decrease the risk of losing the evidence.

## Memory Forensics Approaches

Trying to give a more specific notion about this branch of forensics, emerge two different approaches of investigation: physical and logical. The physical approach performs data extraction at a low level (often with the help of special hardware equipment). The logical approach uses communication protocols offered by the phone at a higher level.

### Logical approach

*Data acquisition via file system analysis or protocol analysis of a chip provider*

Most of Smartphone forensic tools have facilities acquiring digital evidence contained on the flash memory of a Smartphone using some logical protocol between a Smartphone and a host PC. Tools analyze file system from a copied image of the file system to find out the meaningful data as the evidence. As it has fore mentioned, copy of image is done to guarantee the integrity of evidence data during analysis process.

### Physical approach

*Bit by bit physical memory dump*

The information resided in random access memory (RAM) is not possible to access it by means of disk forensics. Hence, data acquisition must be done using physical method by low level approach. The physical level access method can

enhance the recovery rate for deleted data than when using the logical level access method.

Advantages and disadvantages of each approach are quite clear. With the physical method the entire memory with all the contents included, can be extracted from the device. Many deleted data, such as images, files etc. can be restored. Usually, this method is time-consuming and sometimes require complex and special equipment to extract the raw data. Most of the time the results of analyzing this type of data are insufficient for a successful investigation. The reason is that the data in most cases are encrypted. On the other hand, logical approached allow the forensic analyst to obtain more information in a human readable form. The disadvantage of this method is the poor amount of acquired data.

## Data Types

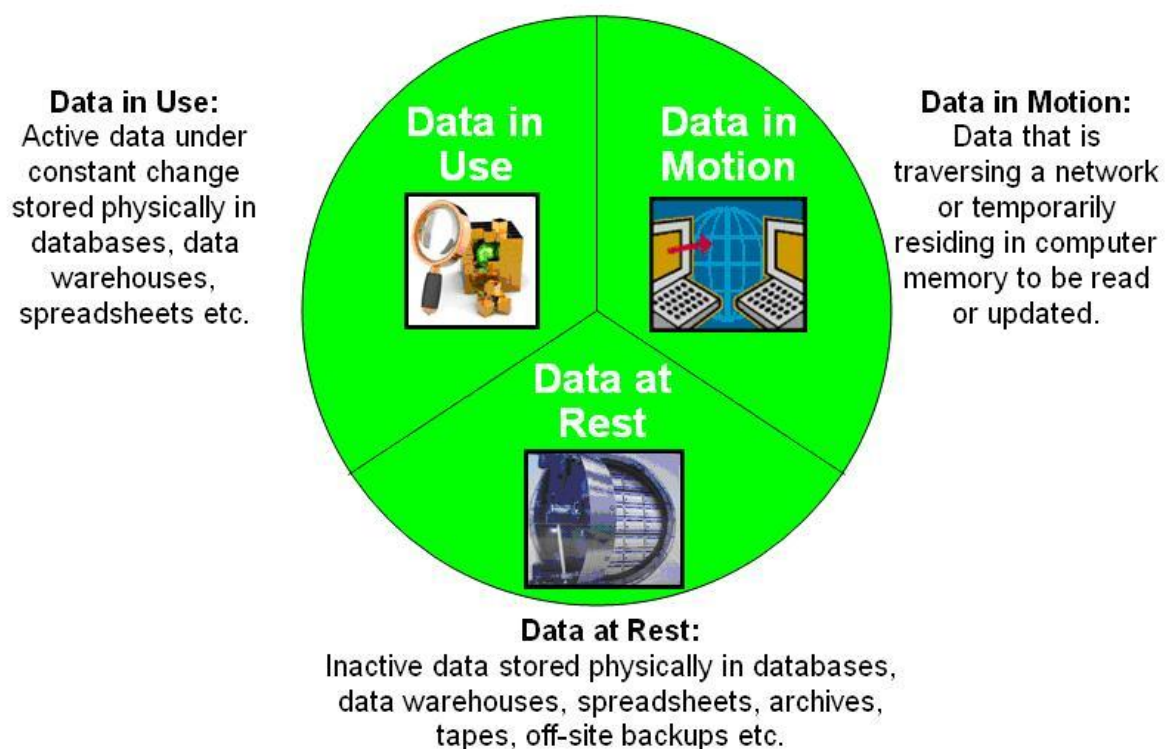
A helpful way of understanding the terms and approaches above is explaining more about the data types. The three types of data are called *data at rest*, *data in use* and *data in motion*.

**Data at rest:** is data recorded in storage media. Examining an Android device for data at rest, one may find the user's personal communication data such as allocated text messages (SMS/MMS) with all the extra data included in this (pictures, media files, contacts etc.). Furthermore, some call logs or voice mails with their metadata (locations etc.) can easily be recoverable. If the user has financial applications installed, the user login, passwords, account numbers or transaction details may be recoverable in some cases. Personal e-mails, web history, Google searches, YouTube URLs, media data (audio, video, pictures etc.) or geo-locations are some additional data at rest a skilled forensic analyst can recover.

**Data in use:** is all data in a non at rest state, that is on only one particular node in a network (for example, in resident memory, or swap, or processor cache or disk cache, etc. memory). Data in use, can contain sensitive data such as encryption keys, digital certificates, intellectual property or personally identifiable information. Compromising this type of data, one can gain access in encrypted data at rest and data in motion.

**Data in motion:** is all data being transferred between two nodes in a network (cellular, Wi-Fi, or other) or data located in RAM. Data and information that never persisted to the device, may have been transmitted so it has to be protected. This type of data refer to passwords that are not stored on the device and requests the user to authenticate every time an application is opened, two-factor authentication or data displayed in an application but not saved or cached to nonvolatile memory.

Securing these types of data, today, is as vital as difficult point for users. The attacking techniques has evolve, making the efficient protection of the device, hard to do. The picture below depicts the activity of data combined with the place where are stored.



## Forensic Process

A forensic process used in digital or mobile investigations vary. An analyst may have his own philosophy which can lead in different procedures. There have been many attempts to develop a process model but so far none have been universally accepted. Part of the reason for this may be due to the fact that many of the process models were designed for a specific environment, such as law enforcement, and they therefore could not be readily applied in other environments such as incident response.

For the aims of this project the chosen four steps are the following: *seizure, acquisition, analysis and reporting*.

### Seizure

The aim of seizure is to preserve evidence. To avoid losing or a possible change in files, the device will often be transported in the same state to avoid a shutdown. The only risk that remain by keeping the phone on is that it can still make network or cellular connection which could modify the evidence or it can simply drain the battery of the device. This step includes the identification and recording of data from all the possible sources.

### Acquisition

The second step in the forensic process is acquisition, meaning of retrieval of material from a device. The acquisition process usually consists of creating a bit-perfect copy of the digital media evidence. The advantage of working on a copy of the evidence is leaving the original media intact ,which allows for any evidence to be verified at a later date. Live acquisition is the data capturing while the device is on and the programs still run. An advantage of live acquisition is for example that allows capturing the contents of RAM. Where a computer is found turned on, prior to seizure, it is sometimes beneficial to make a live acquisition of the RAM in case it contains information deleted from the hard drive.

### Analysis

During the phase of forensic analysis an investigator usually recovers evidence material using a number of different methodologies and tools, often beginning with recovery of deleted material either examining a hard drive or RAM. The aim of any digital investigation is usually to prove or disprove a hypothesis. The aim of the analysis is to unravel the evidence and finally connect the dots.

**Evidence recovery:** Where the analyst identifies information relevant to the investigation and presents it in a neutral form.

**Expert analysis:** Following on from evidence recovery, the analyst draws expert conclusions from the information or connects various pieces of evidence together.

The phases of the analysis are:

- Harvesting of all acquired data
- Identification of violations or concern
- Protection of the proof
- Confirming qualified, verifiable evidence

### Report

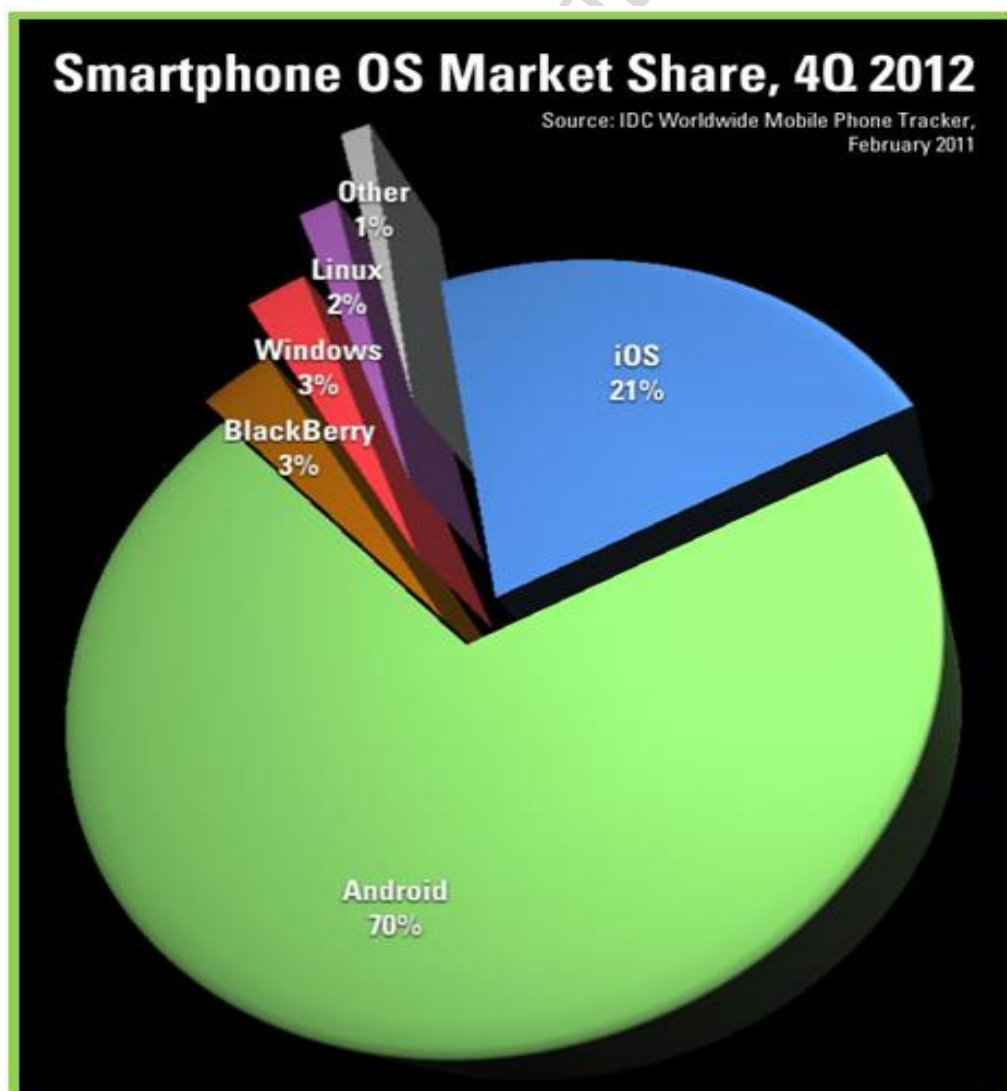
The aim of the report stage is to inform any non-technical individual in a form that can be easily understood. A forensic report typically contains several sections to help the reader understand not only what was found by the investigator, but also to detail the steps performed to acquire and analyze the data. The report should list all software and hardware used to process the data. At the end of the forensic report, a section should be included that is a summary or conclusion of the investigation. This summary will state any facts or conclusions that the investigator has identified and need to be reviewed.



## Android concepts and components

Android is an open source mobile device platform based on the Linux 2.6 kernel. Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software vendors and mobile device and component manufacturers. Two years after the introduction of the first Android device, the platform became the second largest, showing a great influence in peoples' choices.

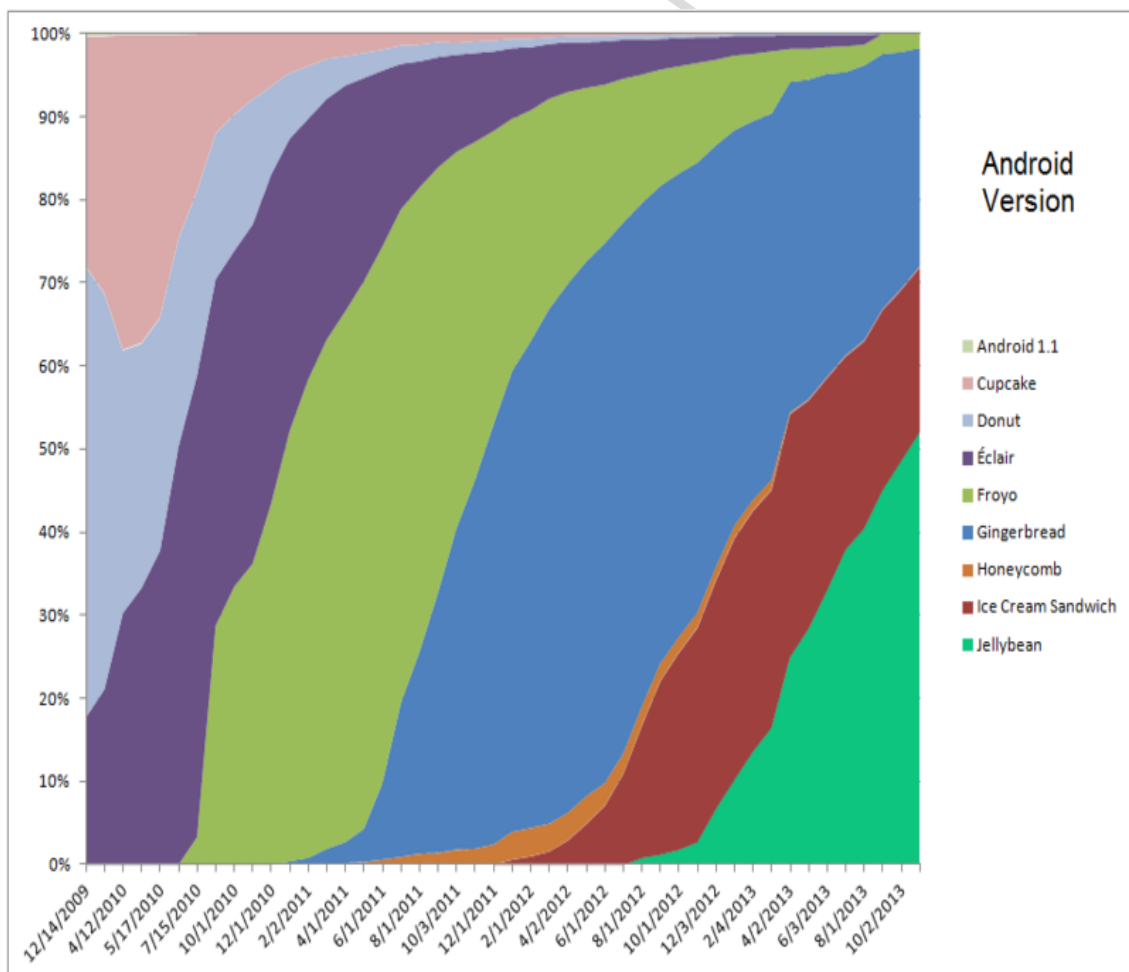
The Android operating system now has a 70 percent share of the world-wide smart phone market, with Apples iOS trailing in second with a 21 percent share. The adoption of Android and its growth make it vital that the forensics analysts be able to properly acquire and analyze evidence from the platform.



Android's open source nature and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers etc. Moreover it has also enabled developers, forensic analysts and sophisticated criminals to understand an Android device in a much deeper level. All these pose challenges to forensic analysts, security engineers and criminals to keep up the 'mouse and cat' game.

## Android version history

The first Android release was the Android beta, in November 2007. From this time and after, nineteen latter versions were released. The diagram below depicts the global version distribution since December 2009.



## Device Types

In October of 2008 the only Android device type in the market was a Smartphone. Rumors said, other types were going to arrive but at this time it was all speculation. As time passed, not only a plethora of Android smartphones popped up but also different types of devices, with the primary types to remain smartphones and tablets. Below is a categorization of most Android devices in market and present time.

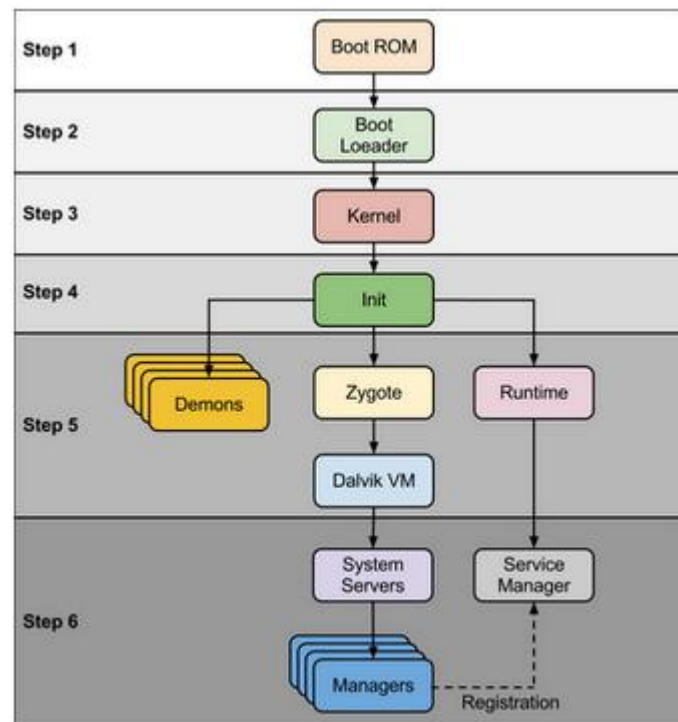
- ✚ Smartphone
- ✚ Tablet
- ✚ Netbook
- ✚ Google TV
- ✚ Smartwatch
- ✚ Car Computers
- ✚ E-readers
- ✚ Game consoles
- ✚ Smartglasses
- ✚ Other devices
  - *Media players*
  - *Office equipment*
  - *Printers*
  - *Appliances such as washing machines and microwaves*
- ✚ Other future devices
  - *Smartbooks*
  - *HUDs (heads-up display)*

As it seems, the Android community, in collaboration with other reputed companies aim to penetrate in any piece of technology in relation to people's lives.



## Boot Process

A basic point in understanding the features and procedures in this project, is knowing all about the boot process. Android is an ARM based linux operating system. Like any other computer, it has a standard boot process to load the firmware and the OS into memory. After pressing the power switch the device executes the following steps.



### 1. Power On and System Startup

When a device is first powered on, starts the boot ROM code executed from a pre defined location, hardwired on ROM. It loads bootloader into RAM and starts execution.

### 2. Bootloader

Bootloader is a small program which runs before Android operating system and as being the first program to run ,it is specific for board and processor. Device manufacturers either use popular bootloaders like redboot, uboot or develop own bootloaders. It is not part of Android Operating System. Bootloader performs execution in two stages. In the first stage it detects external RAM and loads the necessary program for the second stage. In the

second stage bootloader setups network, memory and all the processes required to run kernel.

### **3. Kernel**

Android kernel starts the same way as a Linux kernel starts. The time of launch, it setups cache, protected memory, scheduling, loads drivers etc. When kernel finishes the system setup it looks for “init” in system files and launches root process or the first process of system.

### **4. The Init Process**

Once the kernel has access to the system partition, it can process the init scripts that start key system and user processes. This is similar to the /etc/init.d scripts found on a typical Linux kernel. It also mount directories such as /sys, /dev, /proc. A fundamental issue for the forensic analysts is the way the cache is stored after boot process. At the end of the process, the browser cache is moved into a RAM disk. Any data written to this partition will be lost after power down.

### **5. Zygote and Dalvik**

Zygote process basically cold boots a VM on system start up. Once done, it listens to a socket for incoming commands. Other processes write commands to this socket, whenever it needs a new process for an application. This command is read by the Zygote process and runs another process that gets a pre-warmed up VM in which to run. This is how Zygote forks the Dalvik VM.

### **6. System Services**

The core features of the device mentioned in the previous section are started by the system server. Once the Java runtime is set up and the Zygote process is listening, the system server is started. This runs core features such as telephony, network, and other fundamental components that the device and other applications rely upon.

### **7. Boot completed**

Once system services are up and running in memory, Android has completed booting process. At this time “ACTION\_BOOT\_COMPLETED” standard broadcast action will fire.

## Android SDK and ADB

The investigation of an Android device can be much easier and clearer with the use of specific applications and tools. The Android software development kit (SDK) is the development resource needed to develop Android applications. Provides all the necessary tools and libraries that allow the creation of software applications. It is also a powerful tool in the hands of forensic analysts to build, test, debug applications or to investigate and research the features of an Android device. It can be easily installed in any Windows, Linux, Unix or Mac OS due to its custom versions.

### *Android Architecture*



The SD kit offers a lot of tools. In this project will be used the AVD (Android Virtual Device) emulator to test the behavior of some applications. It lets testing and developing without using a physical device. On the other hand, a forensic analyst may need to have the real physical device in order to implement the analysis and research. Thus, it will be explained the initial phase of an analysis, which is the connection of the device with a workstation.

A forensic analysis has the fundamental principle of keeping safe the host environment on which, the device is tested. This can be applied with the use of sandboxes or Virtual Machines. The connection of an Android device on a host PC or VM is made through a USB port. The Android is programmed to offer more than one choices after the connection. Some of them are the following:

**1. Charge only**

The phone charges over the USB connection

**2. Disk drive**

The device is mount as a disk drive; Opens the SD card

**3. HTC sync**

Syncs the contacts and calendar

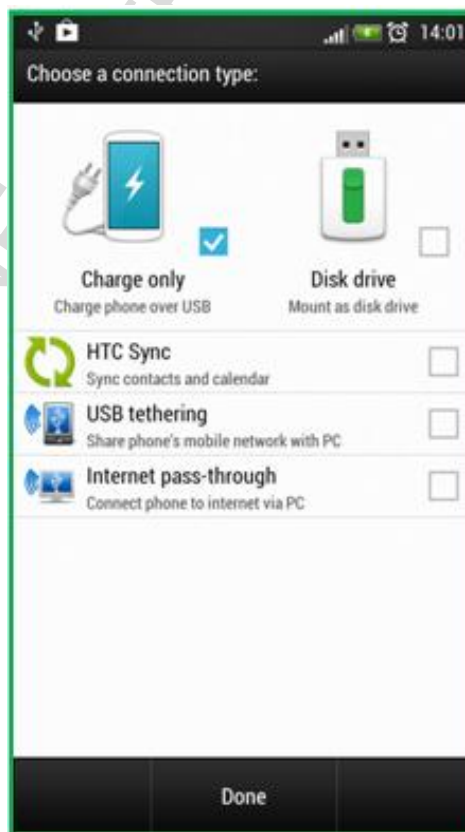
**4. USB tethering**

Shares mobile network with the connected pc

**5. Internet pass-through**

Connect the device to internet via the pc network

The first two options are the basic. The other three are of secondary importance. Charge only is the default option every time an Android device connects to a PC.



### ***ADB (Android Debugging Bridge)***

Probably one of the most important tools in this project, ADB is a versatile command line tool that lets one communicate with an emulator instance or connected Android-powered device. Adopting this on a forensic analyst, it is a client-server program that includes three components:

- A client, which runs on the analysts' development machine. The investigator can invoke a client from a shell by issuing an adb command. Other Android tools such as the ADT plug-in and DDMS also create adb clients.
- A server, which runs as a background process on the analysts' development machine. The server manages communication between the client and the adb daemon running on an emulator or device.
- A daemon, which runs as a background process on each emulator or device instance.

The Android versions have an option for developers. In order to connect and pass data through a USB connection, the USB debugging option must be enabled. Some versions pop up a warning every time the user enables this option.



## File system and partitions

The basic usage of an Android device contains calls, SMS and browsing. However, it is useful to know the internal structure of an Android device. A forensic analyst will aim to understand how data are stored and focus in acquiring and analyzing them.

As fore mentioned, Android OS used a Linux kernel. All file, and directory, operations from an application flow through a kernel abstract layer called the Virtual File System (VFS). The supported file systems vary on different Android devices. For a better and more effective investigation and analysis, it is important an analyst to know the features of the file system used on the device, in order to handle it appropriately. The common flash memory file systems are the following:

- **YAFFS2** - Yet Another Flash File System version 2 was the default AOSP flash file system for kernel version 2.6.32. YAFFS2 is not supported in the newer kernel versions, and does not appear in the source tree. However, individual mobile device vendors may continue to support YAFFS2.
- **exFAT** - The extended File Allocation Table is a Microsoft proprietary file system for flash memory. Due to the licensing requirements, it is not part of the standard Linux kernel. However, some manufactures provide Android support for the file system.
- **F2FS** - Samsung introduced The Flash-Friendly File System as an open source Linux file system in 2012.
- **JFFS2** - The Journal Flash File System version 2 is the default flash file system for the AOSP (Android Open Source Project) kernels, since Ice Cream Sandwich. JFFS2 is a replacement to the original JFFS.
- **EXT** - The standard file system type used by most Linux systems, it had not been a part of any Android device, since December 2010 when Google announced that many devices in the future are going to move from YAFFS to EXT.

From the development or forensic perspective, a developer or an investigator should know the Android internal structure. Android uses several partitions to

organize files and folders on the device just like Windows OS. Each of these partitions has their own functionalities. But most of people don't know the significance of each partition and its contents. In most Android devices the basic partitions are the following:

- /boot
- /system
- /recovery
- /data
- /cache
- /misc

And the partitions of the SD Card:

- /sdcard
- /sd-ext

The adb tool can be used at this point to depict the partitions of the Android device used. The device is an HTC Evo 3D.

```
root@kali:~/usr/share/android-sdk/platform-tools# ./adb devices
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
HT21HV201886    device

root@kali:~/usr/share/android-sdk/platform-tools# ./adb shell
shell@android:/ $ df
Filesystem      Size  Used  Free  Blksize
/dev            398M  136K  398M  4096
/system        787M  638M  149M  4096
/data          1G    345M  830M  4096
/cache         118M   4M   114M  4096
/devlog         19M   17M   1M   4096
/vendor/firmware/misc 199M   21M  178M  4096
/vendor/firmware/adsp 199M   5M   194M  4096
/mnt/asec      398M   0K   398M  4096
/mnt/obb      398M   0K   398M  4096
/app-cache     8M    0K    8M   4096
/data/secure/data 398M   0K   398M  4096
/mnt/sdcard    7G  1017M   6G  32768
/mnt/secure/asec 7G  1017M   6G  32768
/data/DxDrm/fuse: Permission denied
l|shell@android:/ $
```

Using these command above, one can see which partitions are available and also the size used and the free size of each partition. It will be listed in details the six basic partitions along with the sdcard ones.

**/boot** : The boot partition includes the android kernel and the ramdisk. It enables the phone to boot. Without this partition, the device will not be able to boot. Deleting the items in this, the phone must never be rebooted before installing a new ROM with the /boot available.

**/system** : The system partition contains the operating system of the device. Includes the user interface along with all the pre-installed applications. Without this partition the device cannot be rebooted into the OS interface. The user must put the device into the recovery or the bootloader mode to install a new ROM.

**/recovery**: The recovery partition can be considered as an alternative boot partition that lets booting the device into a recovery console for performing advanced recovery and maintenance operations on it.

**/data**: Also called user data, the data partition contains the user's data; the applications, contacts, messages etc. the user installed. Factory reset on a device, means wiping this partition and returning the device as it was after the last official or custom ROM installation.

**/cache**: This is the partition where Android stores frequently accessed data and application components. Wiping the cache doesn't affect the user's personal data but simply wipes the existing data there, which gets automatically rebuilt as the user continues using the device.

**/misc**: This partition contains miscellaneous system settings in form of on/off switches. These settings may include CID, USB configuration and certain hardware settings. Many features of the device will be affected if this partition deleted.

These were the partitions in the internal memory of the device. The external memory, SD card, as mentioned above, will have different partitions.

**/sdcard**: This is the place where a user can store documents, media, ROMs, photos and other personal data. It can be wiped without having any effect on the device's



functions. This partition defers in some types of Android devices. On devices with both an internal and an external SD card, devices like Samsung Galaxy S and several tablets, the /sdcard partition is always used to refer to the internal SD card. The external SD probably have a different named partition.

**/sd-ext:** As not being a standard Android partition, this acts as the /data partition when used with ROMs that have special features called APP2SD+ or data2ext enabled. It is useful in little internal memory devices, where users want to install more programs than the memory allows. Wiping this partition is the same as wiping the /data partition.

## Data Storage

After referring the partitions of the Android OS, it is important to know what kind of methods the Android provides, for data storage. A forensic analyst should have ample knowledge about where exactly the data are stored, in order to be able to uncover them. Therefore, it is important to be aware of each detail. The methods are the following:

1. **Internal storage** : The files are stored in the application's /data/data subdirectory and the developer has control over the file type, name, and location. The owner must have root privileges to view the files. Overriding the security settings, the analyst will be allowed to read or update the files.
2. **External storage** : As it mentioned above, the files are stored in an SD card. These files have much less restrictions and are easily readable and modifiable. The SD card and the internal data can be also used in other devices, giving the analyst great control over the names, formats or locations of the files.
3. **SQLite** : A NAND/SD card-based storage that analysts leverage is an SQL database. It is used for structured data storage and is popular due to the high quality of the base and it's open source nature. Provides also great forensic material of data with high probability of recovering them.
4. **Network** : At this time only a few users or applications take advantage of this kind of data storage. The network databases provide important forensic data and useful information.

## Data in Memory

Pointing on the data resident in memory, there is a classification of them into four categories:

- Metadata
- Data files
- Sensitive data
- Case irrelevant data

*Metadata* is the data that clarifies other data such as the number and the names of running and terminated processes, start and end time, names of the accessed files and the dll files they had used for the course of execution. Metadata is an important part of the memory examination due to its great evidence material. *Data files* are of great significance on an investigation. Contain valuable information like images and sensitive data. Finding some log files will be very helpful for the forensic process.

*Sensitive data* are parts of data such as passwords, encryption keys or URLs the user of the device has used or access. Many times this material is not part of other files; is passed as parameters to functions of the device processes. Finding this kind of information will help in detecting a person, or a place etc. Finally, data that is not part of the above categories is irrelevant with the forensic examination.

## Rooted Device

An Android user with his device at his hands has just the same privileges as any other simple Linux user . Rooting an Android device is like becoming root user in Linux systems. In other words, is basically obtaining all the rights and permissions of the Android's operating system. However, there are some differences in pros and cons between them.

Rooting an Android device enables to perform plenty of functions that where "locked up" under the low range of privileges a user had by default. Root access can be used legitimately or illegitimately; depends on the individuals' intensions . A rooted device may collect much more malicious stuff than a factory default device.

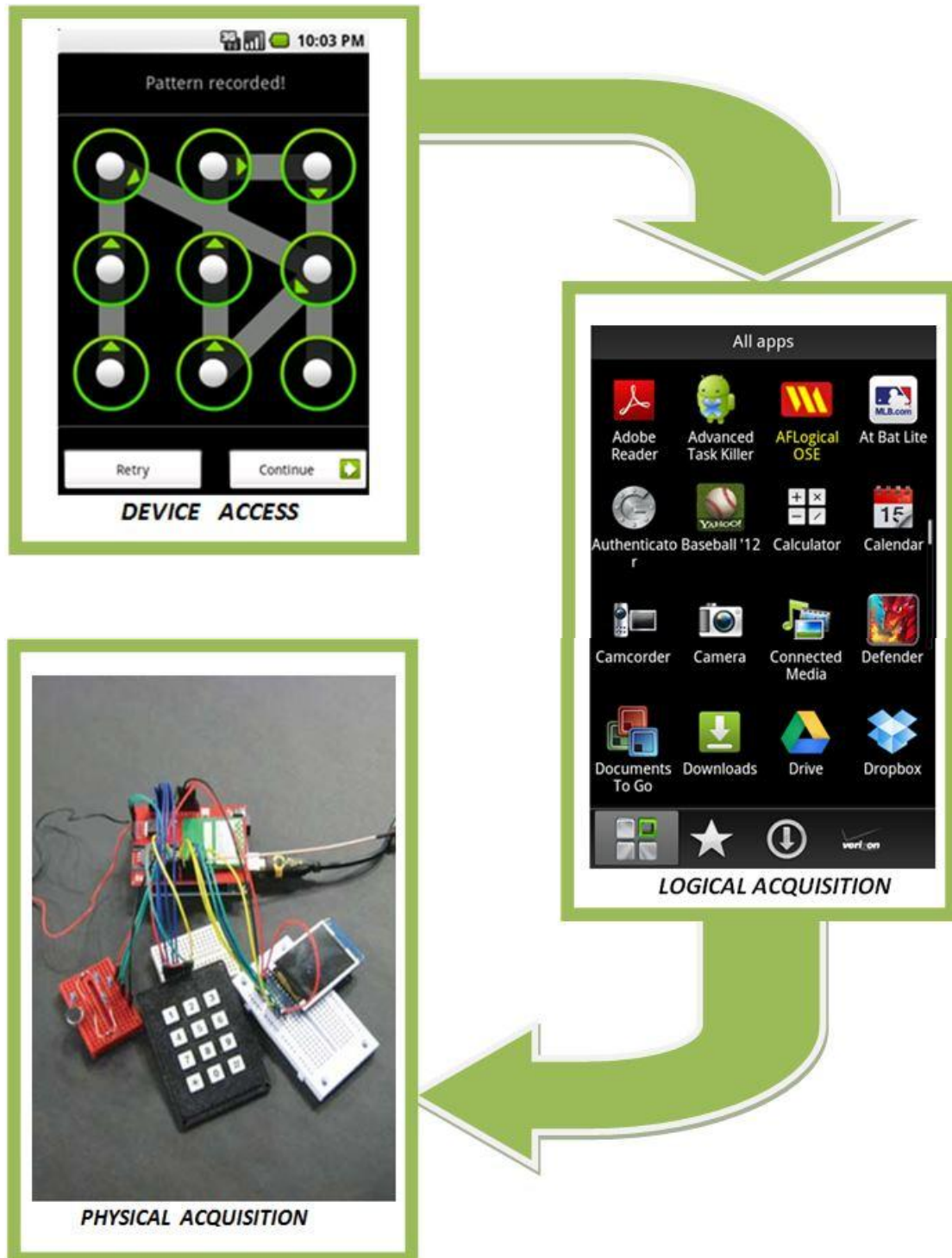
From the user's perspective, other disadvantages that may be caused by rooting a device are, the possible loss of performance and speed or the total "bricking" of the device, making the phone unable to use or perform its functions.

Nevertheless, this unrestricted access in the sanctuaries of the device is a significant tool in hands of a forensic analyst. Pure data can be extracted, security applications may use root privileges to gain access to unwatched parts etc. In this project, a rooted device is necessary in order to use more tools with better results. However, rooting should be avoided as much as possible, due to the security vulnerabilities that may be caused.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## Accessing Data

Since the investigator takes the device in his hands, there are three levels of data access. The device access, the logical acquisition and the physical acquisition. These three levels can lead to a certain forensic result.



## Device Access

Most Android devices contain different types of screen locks to protect and secure the mobile environment from non legitimate users. The owner can choose between a screen lock with password, with PIN number or with a pattern. An analyst will have to bypass it in order to gain access in the internal environment of the device.

The pattern lock was the default on the initial Android devices. To access the device, the user draws a pattern on the locked phone and, if drawn properly, the device is unlocked. If the owner has chosen to lock his screen with a pattern, research and proposals suggest that this pattern can be detected from the finger trace on the screen, as seen on the picture below.



The second type of pass code is the simple personal identification number (PIN) which is commonly found on other mobile devices. The other type of pass code currently found on Android devices is a full, alphanumeric code.

Some tools provide the option of cracking the PIN number (up to 4-digits) by brute force. FROST (Forensic Recovery of Scrambled Telephones) is a tool set that supports the forensic recovery of scrambled phones and has the option of key cracking.

In this project it will be proved that the above steps are not necessary if the forensic analyst flash a new recovery partition or gain root access of the device. Some users install a custom ROM which usually enables root access to the device through a modified recovery mode. Most custom ROMs install a modified recovery partition which simplifies the process used to install the custom ROM. There are several popular recovery partitions that are primarily used with custom ROMs and both offer shell access with root privileges from within the recovery console itself.

The chosen screen pattern or the PIN numbers and passwords are saved into the file system of the phone and specifically in /data/system/. Thus, by having access to the complete file system, any screen protection can be modified or deleted.

The first step is the usb connection of the smartphone to the terminal. After that, by using adb commands, a shell will be opened from the terminal to the device.

```
root@kali:~/Desktop/android-sdk/platform-tools# ./adb devices
List of devices attached
HT21HV201886    device

root@kali:~/Desktop/android-sdk/platform-tools# ./adb shell
shell@android:/ $ su
shell@android:/ #
```

The next step is to remove the gesture.key (Pattern lock) or password.key (PIN or password lock) files from the file system with the following commands. The above key files include the hash values of the passwords or the pattern the user has chosen.

```
shell@android:/ # cd /data/system
shell@android:/data/system # rm gesture.key
shell@android:/data/system # rm password.key
```

Furthermore, on most Android phones, one can bypass the pass code if he knows the primary Gmail user name and password registered with the device. After a number of failed attempts, a screen will be presented that asks if the owner forgot his pass code. From there, it can be entered the Gmail user name and password and then the pass code can be reset. This technique does not require the phone to be online as it uses credential information cached on the phone.

## Logical Acquisition

Logical extraction usually does not produce any deleted information. However, in cases of SQLite built platforms, such as iOS and Android, some database files marked as deleted may not be totally overwritten; something useful for a forensic analyst. In such cases, if the device allows file system access through its synchronization interface, it is possible to recover deleted information. File system extraction is useful for understanding the file structure, web browsing history, or app usage, as well as providing the analyst with the ability to perform an analysis with traditional computer forensic tools.

Many logical acquisition tools are in the market, with the best and more effective being the AFLogical. AFLogical performs a logical acquisition of any Android device running Android 1.5 or later. Emails, Geolocation database, applications “Cache” folder (opened files in Dropbox) or executables are not available in a logical acquisition. A more specialized edition of the AFLogical, The AFLogical Law Enforcement edition, is able to pull all logical data from an Android device, including: Browser Searches, Calendars, CallLog Calls, Contacts Phones, External Media, Messages, Providers, Maps, SMS-MMS, Search History or Social Contracts Activities.

The extracted data is saved to the examiner’s SD Card in csv format (comma separated values) which easily imports into popular spreadsheet software, making it simple to extract and analyze Android data. A note an examiner should keep in mind is that the SDcard has to be totally wiped out. Otherwise, the contents may be replaced by the results and be lost. For this project, two AVD's were used. As indicated before, AVD is an Android Virtual Device that simulates the Android environment of a mobile phone and is part of the Android SD Kit.

To help the simple version of AFLogical come with a result, some phone contacts and similar data were added in both AVDs. Additionally, SMS messages and calls were part of communication data between the Virtual Devices as seen in the figure below.

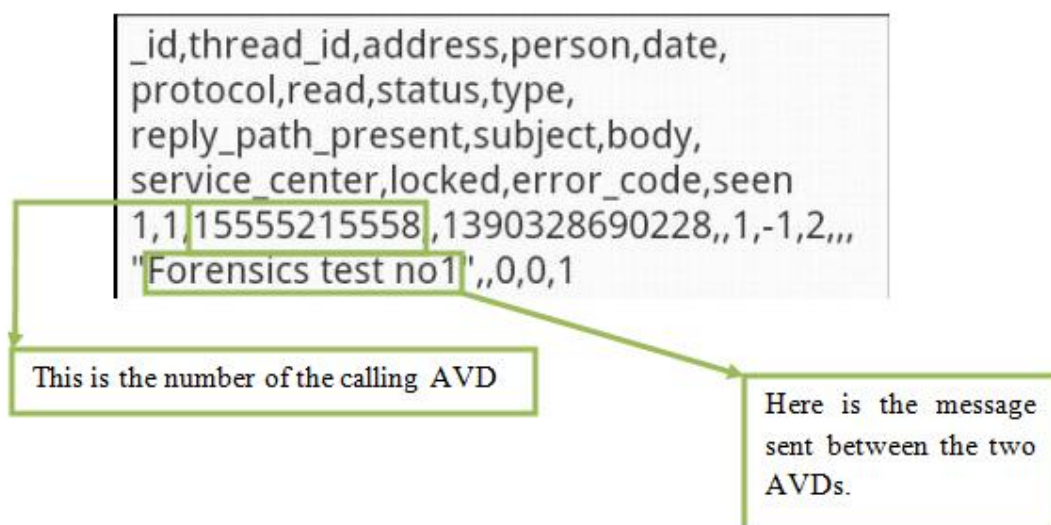




Opening the file system, where the forensic results are stored, different .csv files with communication, file structure or web history data are presented. As one can see, the folder verifies the actions of the AFLogical tool. It creates a "forensics" folder in which another folder, with the name of the date the extraction took place, contain the extracted data.

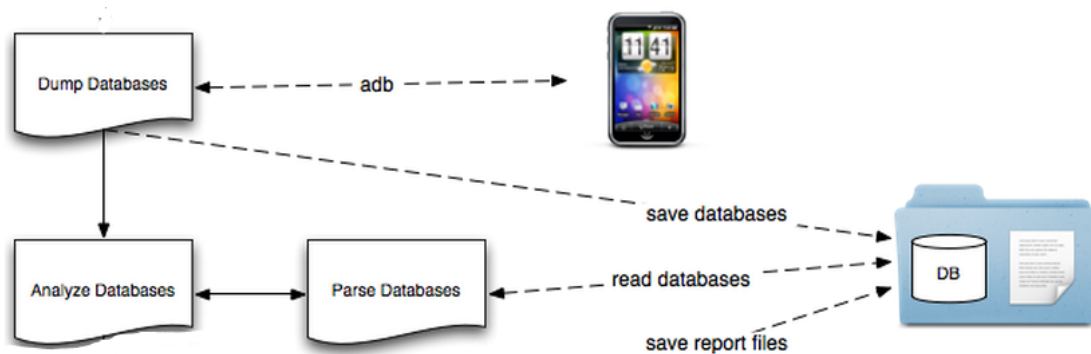


Following the previous testing process and opening the SMS and the call logs folders, the contents are depicted below.



## SQLite Databases

In this section, another part of logical acquisition has been implemented. Being more accurate, the exact category the SQLite database examination belongs, is the combination of logical and physical acquisition methods. Physical extraction of the database via adb tool and logical examination of the extracted data. As written in the Data Storage section of this project the sqlite databases are used for structured data storage and provide great forensic material of data with high probability of recovering them. The databases are stored in a specific subfolder of the /data folder. The exact path is /data/data. The database dump method is a simple one with the first step being the adb connection of the terminal with the device. After that, the examiner reads and extracts all the useful information and then prepares the report files.



Entering the internal system of the device via adb, the /data/data directory contain the paths to different types of databases alongside with the personal data of the user. An analyst can extract valuable information about the contacts, the SMS/MMS communication with date and time data, browser information such as website traffic and also application or website usernames and passwords. The extraction procedure starts with the imaging of a chosen database in order to be parsed without being modified. The files are copied in the SD card with the dd command.

```
dd if=/data/data/subdir/databases/chosen.db of=/sdcard/chosen.db
```

with if and of being the input and output file respectively. The contacts database of the device used for this project, is stored in the

```
/data/data/com.android.providers.contacts/databases
```

directory. The `contacts.db` file database copied in the SDcard and opened with the SQLite Database Browser to be analyzed. Searching the internals of the database,

356	697211096798	387111096798	135	2
357	6957115513555	387115513555	0	3
358	6985117800556	387117800556	132	1
359	6985118820572	387118820572	204	1
360	6972128429552	387128429552	0	3
361	6985128870153	387128870153	1	2
362	6972128881798	387128881798	42	2
363	2104173780546	387173780546	36	1

discovered the latest numbers called, the duration of the call alongside with the priority numbers of the contacts and other information.

Following the same process of browsing each of these folders, listing the subdirectories and looking for databases, several valuable folders were uncovered. Aiming on the SMS communication, the

`/data/data/com.android.providers.telephony/databases`

was the directory from which the SMS communication database was extracted. This database directory contains information related to the messaging applications, including picture and text message data. The `mmssms.db` database contains the MMS and Short Message Service (SMS). Furthermore, the contents in this database included some deleted messages although no messages that were deleted more than 45 days prior were available. It is likely that the retained deleted messages would depend on the device and user.

000	59	10	3069451101102	ΜΠΑΙΝΩ Γ ΜΠΑ	0	1	0	Κα
552	48	11	306957500265	Tha dw meta re.	0	1	0	Ου
199	51	12	697511000033	Me phreco mpa	0	1	0	Μι
642	2	13	698511000015	ΔΝ ΠΡΟΛΑΒΑΙΝ	0	1	0	2;N
000	1	14	SP EDITION	Η ΠΟΛΗ ΜΑΣ ΕΓ	0	1	0	SP EDITION
000	26	15	3069751100010	ΠΑΥΛΑ*	0	1	0	Στ

The above figure lists the numbers that took part in the messaging communication, the text message, the number 1 or 0 if it was read or unread respectively and the name of each contact. Using the same method the examiner would find valuable information by parsing the MMS messages or something more important such as the Voice Mails audio files.

/data/data/com.android.browser/databases/

is a separate direction for the Android browser. The contents of this folder included databases with usernames, URLs, cookies, passwords, data typed into forms, web browser history and search history. The passwords were stored in plaintext and were easily viewed with the SQLite Database Browser.

_id	host	username	password
1	2 httpsm.facebook.com	l...@hotmail.co	.....
2	3 httpslogin.live.com	v...@hotmail.com	.....

Some of the information of the history search were deleted from the device but as shown in the figure below a part of them was recovered or it was never deleted completely.

_id	title	url	created	date	visits
5	SPORT 24 - ΣΥΝ	http://www.sp	0	389974771158	56
18	RED PLANET -	http://www.re	0	389031240299	70
25	Redplanet.gr	http://m.redp	0	390478075104	82
30	Wiziwig.tv   Fo	http://www.wi	0	390594718604	12

Finally, useful and valuable information can be discovered by searching and analyzing the website cookies the browser was keeping.

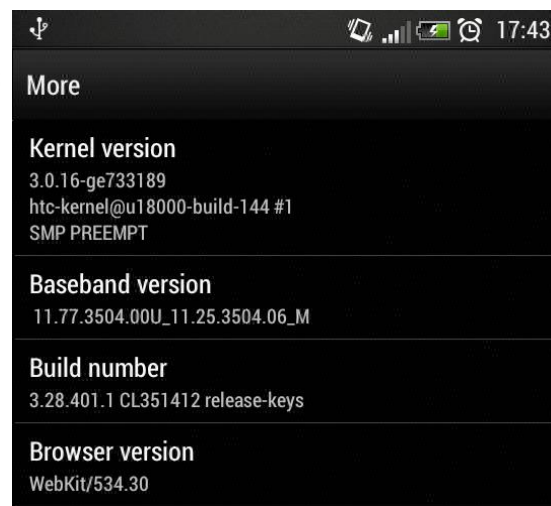
creation_utc	host key	name	value	pa	expires_utc	sec	ht	last access_utc
471570230853	.google.gr	PREF	ID=0ec58ceb24	/	543570000000	0	0	35069807725511
471634709765	.facebook.com	datr	2d9uUj9SGx5n	/	543630000000	0	1	35232780416369
471634711444	.facebook.com	lu	RgEZRgvEGuXc	/	543634000000	0	1	35232780416369
471713244860	.sport24.gr	__gads	ID=285779a7a	/	543712000000	0	0	35233437941821
471726330065	.doubleclick.net	id	22a2c41bb701	/	543726000000	0	0	35233441270983
471948850054	.dailymotion.c	v1st	5282C882183F	/	596080000000	0	0	35233442101000
471948853808	.scorecardrese	UID	705a2e7e-212	/	679948000000	0	0	35232952724627

In conclusion, the logical acquisition with appropriate tools such as AFLogical offers a categorized grouping of communication, access history and file structure data

in order the examiner to gain a first forensic insight of the device. In other words it is an early start of understanding the internals of the device alongside with the user's personal data; indispensable for a successful forensic procedure. Searching the sql databases can be a vital part of the forensic examination as it can bring to the surface valuable information such as passwords or parts of browser history that may have been deleted. However, more sophisticated analysts use most, the physical data extraction due to the ability to recover deleted files and the wide range of data stored in memory (RAM or NAND Memory). It is valuable to perform a physical examination to access deleted information that might otherwise go unnoticed.

## Physical Acquisition

Physical acquisition of a device, most of all, is necessary for one thing. Recovering deleted data. As written in previous sections the physical examination of the device is a vital part of the forensics procedure. In this project the Android device which was the main part of the forensic analysis, was an HTC Evo 3D running Android version 4.0.3. (a.k.a. Ice Cream Sandwich).



Gaining access to the root directory and having the appropriate permissions to take root actions was necessary. For better results, the file system should be totally free to access for the examiner.

In earlier stages of the Android OS the MTD software (Memory Technology Device-allows the embedded OS run directly on flash) was the connecting link between the forensic analyst and the mapping of the file system partitions. The file system of the Android device is stored in a few different places within /dev. Although it may differ for other android phones, there are six files of interest located in /dev/block ; the path for earlier Android OSs was /dev/mtd. HTC Evo 3D's contain eight different partitions ready for imaging and examination, as seen on the figure below.

```
0 /tmp ramdisk (null) (null)
1 /boot emmc /dev/block/mmcblk0p20 (null)
2 /recovery emmc /dev/block/mmcblk0p21 (null)
3 /system ext4 /dev/block/mmcblk0p22 (null)
4 /data ext4 /dev/block/mmcblk0p23 (null)
5 /cache ext4 /dev/block/mmcblk0p24 (null)
6 /misc emmc /dev/block/mmcblk0p31 (null)
7 /sd-ext ext4 /dev/block/mmcblk1p2 (null)
8 /sdcard vfat /dev/block/mmcblk1p1 /dev/block/mmcblk1
```

Each of the partitions have been explained in earlier sections of this project. The most significant ones for analysis are the /recovery, the /system, the /data and the /cache partitions. More specialized information can be found with the mount command.

```
shell@android:~ # mount

rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,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
```

```
/dev/block/mmcblk0p22 /system ext4
ro,relatime,user_xattr,barrier=1,data=ordered 0 0

/dev/block/mmcblk0p23 /data ext4
rw,nosuid,nodev,noatime,user_xattr,barrier=1,nodelalloc,data=ordered
0 0

/dev/block/mmcblk0p24 /cache ext4
rw,nosuid,nodev,noatime,user_xattr,barrier=1,nodelalloc,data=ordered
0 0

/dev/block/mmcblk0p28 /devlog ext4
rw,nosuid,nodev,noatime,user_xattr,barrier=1,nodelalloc,data=ordered
0 0

/dev/block/mmcblk0p17 /vendor/firmware/misc vfat
ro,relatime,mask=0000,dmask=0000,allow_utime=0022,codepage=cp437,ioc
harset=iso8859-1,shortname=lower,errors=remount-ro 0 0

/dev/block/mmcblk0p19 /vendor/firmware/adsp vfat
ro,relatime,mask=0000,dmask=0000,allow_utime=0022,codepage=cp437,ioc
harset=iso8859-1,shortname=lower,errors=remount-ro 0 0
```

### Imaging Process

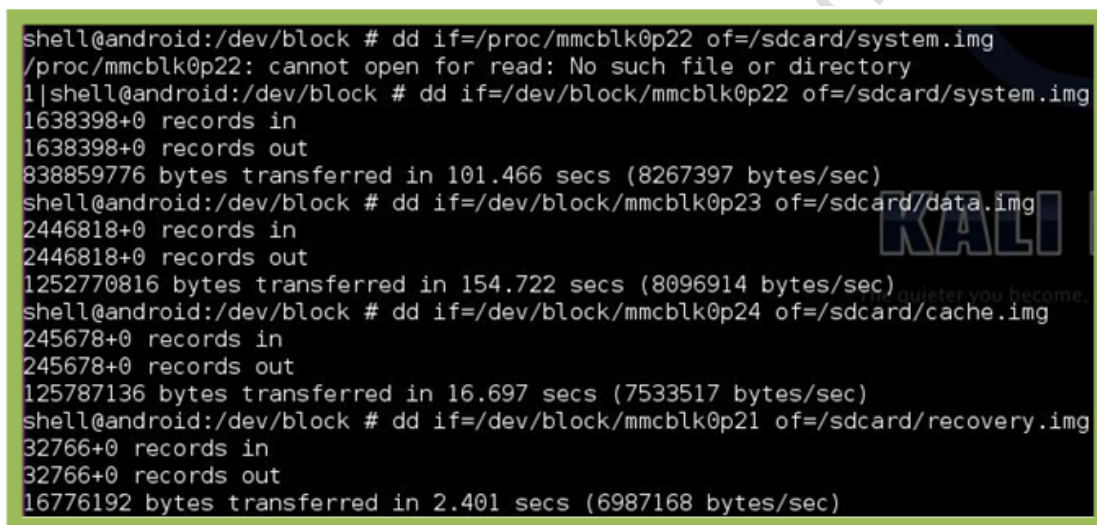
The first step of this procedure is the connection of the device with the Linux running terminal. After the customization of the terminal, starts the adb shell connection with the file system confirming the success with the # symbol. At this point, there are three strategies of physical acquisition:

1. Full nanddump of all partitions
2. A dd image of partitions
3. A logical acquisition of files using adb (SQL Database Extraction & Analysis)

The command

```
dd if=/dev/block/partitionblock of=/sdcard/partitionimage.img
```

can be adjusted to every block, every memory part the examiner needs to image. This method was used to choose specific partitions to image instead of the full memory. Most parts of the file system were wiped before the imaging to test the recovery ability. The images were stored in the sdcard. The command above will make a bit-for-bit image of the chosen file, with an option of block size to be also available. For four partitions, the dd command was used four times.



```
shell@android:/dev/block # dd if=/proc/mmcblk0p22 of=/sdcard/system.img
/proc/mmcblk0p22: cannot open for read: No such file or directory
1|shell@android:/dev/block # dd if=/dev/block/mmcblk0p22 of=/sdcard/system.img
1638398+0 records in
1638398+0 records out
838859776 bytes transferred in 101.466 secs (8267397 bytes/sec)
shell@android:/dev/block # dd if=/dev/block/mmcblk0p23 of=/sdcard/data.img
2446818+0 records in
2446818+0 records out
1252770816 bytes transferred in 154.722 secs (8096914 bytes/sec)
shell@android:/dev/block # dd if=/dev/block/mmcblk0p24 of=/sdcard/cache.img
245678+0 records in
245678+0 records out
125787136 bytes transferred in 16.697 secs (7533517 bytes/sec)
shell@android:/dev/block # dd if=/dev/block/mmcblk0p21 of=/sdcard/recovery.img
32766+0 records in
32766+0 records out
16776192 bytes transferred in 2.401 secs (6987168 bytes/sec)
```

### Memory Examination and File Recovery

Before any action on the images, it is known that the /data partition contains most of forensic evidence an analyst can retrieve. This was proved also by the size of the /data image (1.2 Gb). The dd images contain only raw data; difficult or impossible to gain a forensic result. At this point, a tool should be used to pull every kind of data the image contains, such as images, documents, pdf files, media files, zipped files etc.

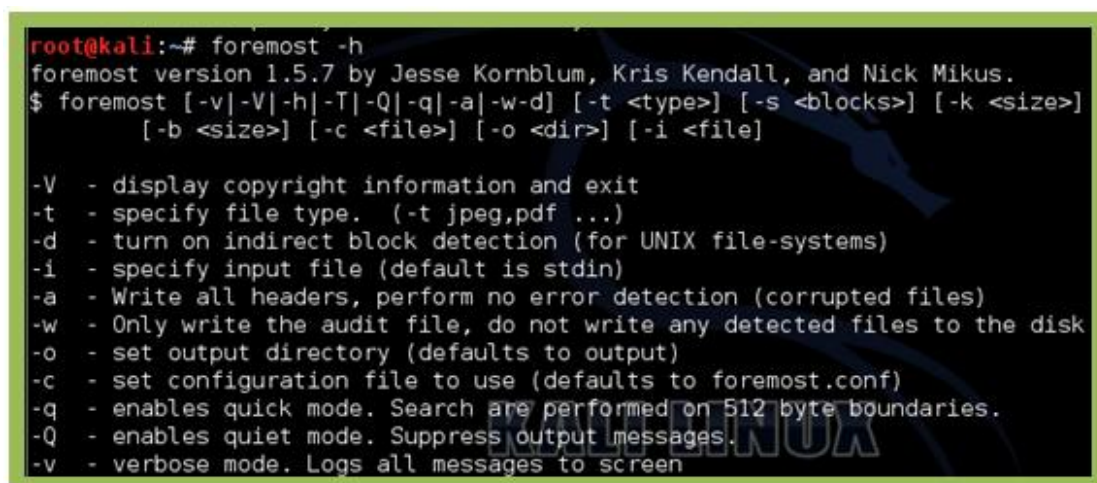
One of the best tools out there is called Foremost. The tool was created in March 2001 to duplicate the functionality of the DOS program CarvThis for use on the Linux platform. Foremost is a console program to recover files based on their headers, footers, and internal data structures. This process is commonly referred to as



data carving. Foremost can work on image files, such as those generated by dd, Safeback, Encase, etc, or directly on a drive. The headers and footers can be specified by a configuration file (usually found at `/usr/local/etc/foremost.conf`) or one can use the command line switches to specify built-in file types. These built-in types look at the data structures of a given file format allowing for a more reliable and faster recovery.

Foremost is designed to ignore the type of underlying file system and directly read and copy portions of the drive into the computer's memory. It takes these portions one segment at a time, and using data carving searches this memory for a file header type that matches the ones found in Foremost's configuration file. When a match is found, it writes that header and the data following it into a file, stopping when either a footer is found, or until the file size limit is reached.

Foremost is used from the command-line interface, with no graphical user interface option available. It is able to recover specific file types including *jpg, gif, png, bmp, avi, exe, mpg, wav, riff, wmv, mov, pdf, ole, doc, zip, rar, htm* and *cpp*. There is a configuration file (usually found at `/usr/local/etc/foremost.conf`) which can be used to define additional file types.



```

root@kali:~# foremost -h
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k <size>]
  [-b <size>] [-c <file>] [-o <dir>] [-i <file>]

-V - display copyright information and exit
-t - specify file type. (-t jpeg,pdf ...)
-d - turn on indirect block detection (for UNIX file-systems)
-i - specify input file (default is stdin)
-a - Write all headers, perform no error detection (corrupted files)
-w - Only write the audit file, do not write any detected files to the disk
-o - set output directory (defaults to output)
-c - set configuration file to use (defaults to foremost.conf)
-q - enables quick mode. Search are performed on 512 byte boundaries.
-Q - enables quiet mode. Suppress output messages.
-v - verbose mode. Logs all messages to screen

```

The *help* command on the terminal displays the preferences a user can make depending on the type of file carving he wants. The figure above depicts the exact kind of preferences of the tool. The execution command

```
foremost -i /image/path/imagefile.img -o /chosen/path/outputfile
```

is the simplest version that starts data carving of the file. *-i* defines the input image file and *-o* the output one. After the execution the tool starts to search for headers, footers and other info of the stored data.

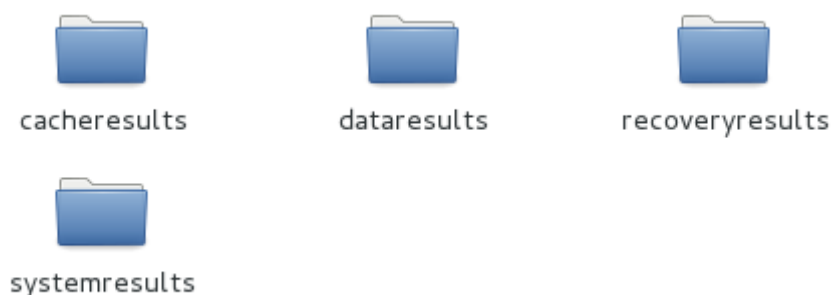
Using this command customized for each extracted image, the carving procedure runs four times.

```
foremost -i /root/Desktop/ddimages/...img -o /root/Desktop/result/
```

The sdcard image can be copied with two ways. Either with the above one, or by running the connected device to be mounted as an external data drive.

### Recovered Data

As it was mentioned before, in the device and its SDcard were stored only a few images and other kind of data. Most of the previous stored data were intentionally deleted. The Foremost tool recovered a lot of the deleted data and categorized them by its nature. Most of the recovered documents were not of a real evidentiary value. Other files were extremely fragmented and while they were reported that they were corrupt and could not be opened, Foremost recovered parts of them that could be readable.



The */data* partition contains the user's personal data. Thus it is more likely to recover useful forensic material. Opening the recovered data to search and analyze, the Foremost tool managed to extract many types of data according to the configuration file and the plethora of files the Foremost creators state that retrieves.

The figure below depicts the above statements. The number of recovered elements depend on the type of memory storage, the encryption and the number of the stored data (deleted or not).



### Audit File Report

```
Foremost started at Mon Jan 27 13:45:52 2014
Invocation: foremost -i /root/Desktop/ddimages/data.img -o /root/Desktop/dataresults/
Output directory: /root/Desktop/dataresults
Configuration file: /etc/foremost.conf
```

```
-----
File: /root/Desktop/ddimages/data.img
Start: Mon Jan 27 13:45:52 2014
Length: 1 GB (1252770816 bytes)
```

```
Finish: Mon Jan 27 13:47:31 2014
```

```
23719 FILES EXTRACTED
```

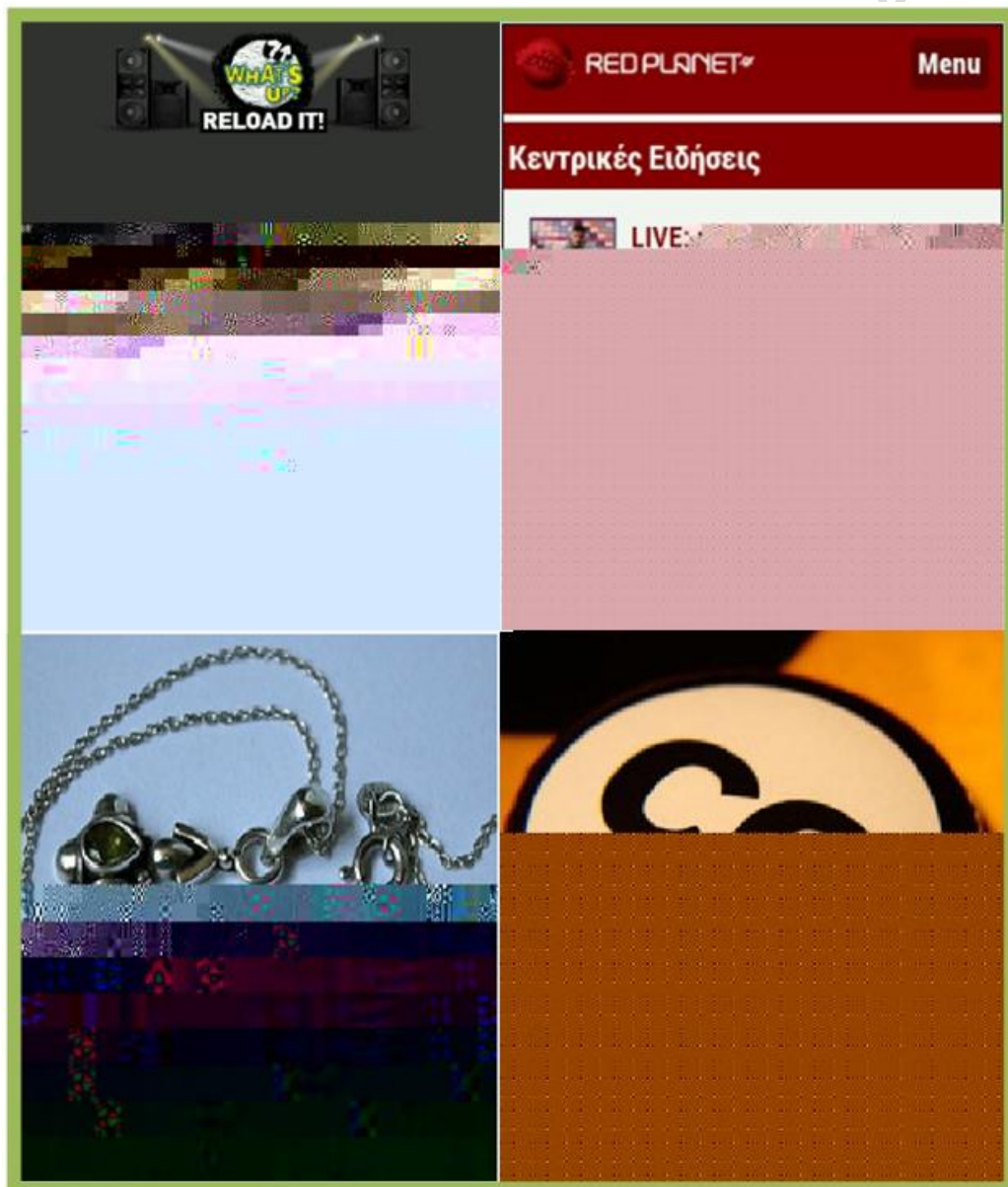
```
jpg:= 3382
gif:= 305
bmp:= 5
mov:= 16
mp4:= 12
rif:= 3
htm:= 32
zip:= 252
png:= 19712
```

```
-----
Foremost finished at Mon Jan 27 13:47:31 2014
```

### Recovered Images

A serious point to be mentioned is that most of the recovered data were not stored in the internal memory; it was accessed by the user in the browsing process. The images were either full recovered or parts of them. One way or another, plenty of forensic material was extracted.

*Corrupted image files*



*Intact image files*

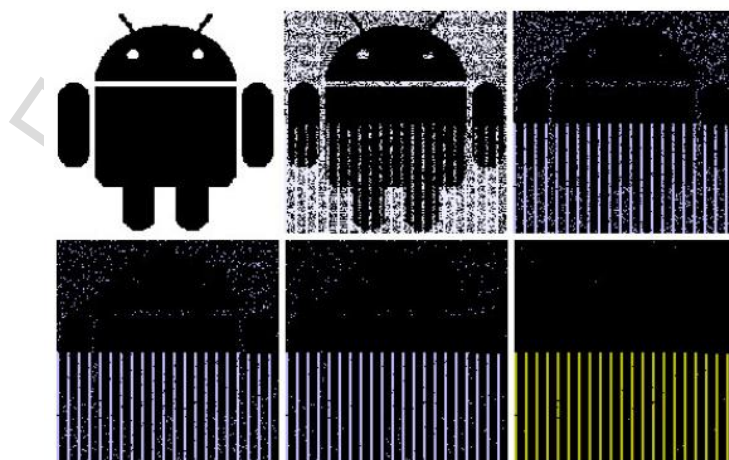


Evidence found in a physical memory examination can be used to reconstruct crimes finding temporal, relational and functional information. It is also a very useful way of defeating anti-forensic techniques alongside with the compromise detection on a live system.

## Other Forensic Techniques

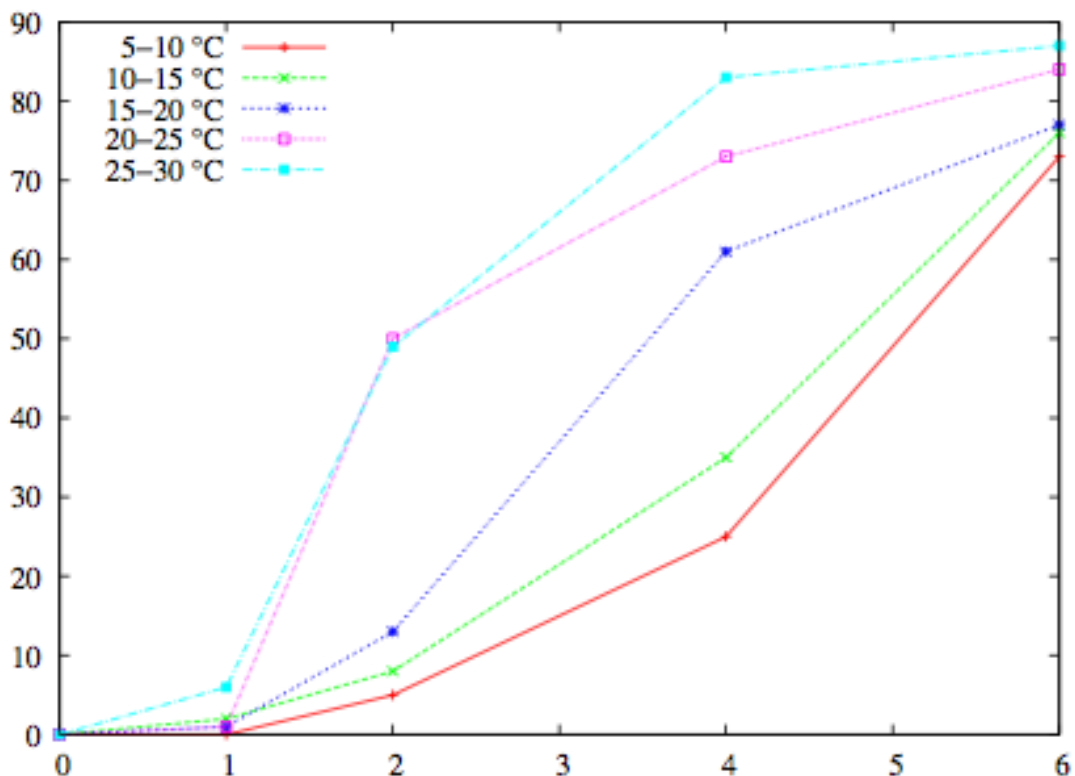
As the time passes, more and more techniques appear. One of the most effective in finding serious evidence, is the random access memory examination. Passwords, encryption keys, URLs, images and other sensitive data can access the RAM while the user is using a device. Alongside with the progress of these techniques, many tools have been developed.

A software, called FROST, has been open-sourced by the researchers and is reasonably easy to use, and has shown that it can extract the RAM memory, encryption keys etc. from Android smartphones, even if the phone is locked and the disk is encrypted. The attack vector is referred to as a cold boot attack. Cold booting (or hard booting) is where the system reboots by cutting the power completely, and then turning back on. When a computer restarts normally (i.e. a warm reboot), there are usually processes in place that clear/sanitize the system's memory. However, by cold booting and bypassing these processes, the contents of RAM are preserved. The image below depicts the stages of volatile data in RAM from the time it accesses the memory to the time it is wiped.



Using the Frost tool, the researchers can access all data stored in RAM. Given that phones are rarely switched off, this often contains a significant cache of sensitive personal data. It can be recovered fully intact address book contacts, thumbnail photos, and Wi-Fi credentials, and partially calendar entries, emails, text messages, high-resolution photos, and Web history. There are no easy defenses against the attack, other than turning a phone off before it's out of the owner's possession. Rebooting a phone more often may also leave less sensitive data in its memory. The researchers say they haven't yet tested the attack on other phones, but believe that it would likely be much more difficult on iOS.

Borrowing an image from the Frost researchers' page, the cold boot attacks by freezing the device is being further understood by showing the deterioration of data in memory (in percent of memory lost) over time (in seconds) at different temperatures.



## Anti-forensics

To maintain the interest in the "cat and mouse" game between an individual with illegal purposes and a forensic analyst, appear some anti-forensics techniques used as countermeasures to forensic analysis. Lending Dr. Mark Rogers' of Purdue University definition of this newly recognized as a legitimate field of study, anti-forensics are "Attempts to negatively affect the existence, amount and/or quality of evidence from a crime scene, or make the analysis and examination of evidence difficult or impossible to conduct." Many different opinions appeared about the field of anti-forensics. Some claimed that has malicious intentions, others stated that the result will be a production of better forensic tools and better education for the forensic examiner. Anti-forensics also called counter-forensics or attacking forensics; depending on the targets of the individual with malicious intentions. A closer look on the types of anti-forensics are listed below.

Data hiding is the attempt to hide the evidence and make the finding and access hard for the analyst. Some tools based in Android OS, move some kind of evidence such as SMS. The messages one tries to hide are transferred in different storage place to be undetectable. Other tools use the common forms of obfuscation or encryption of evidence. On the other edge, definitive hiding of the evidence can be by total destruction of data. That kind of tools, are time scheduled to delete specific data or to wipe a chosen storage place. Another technique is the altering of evidence; the modification of data to present different results than the malicious individual wants. A variation of the above method is the counterfeiting of evidence. The result is the attempt of misleading the forensic analyst by adding massive amounts of fictitious data to existing data sets. Tools programmed to perform such actions, inject data into the phone's internal memory.

Anti-forensic methods rely on several weaknesses in the forensic process including: the human element, dependency on tools, and the physical/logical limitations of computers or mobile devices; although these methods are not so widespread for smartphones. Every new method arrives must be a lesson for every investigator to think and look twice and to not take everything at face value. Anti-



forensics tools and methods will continue to provide difficulties and challenges to the mobile device investigation as smartphones penetrate more and more into the daily routine. Better education and learning of these weaknesses will lead in a production of more powerful and adoptive tools and fewer successful attempts of frustrating the forensic examiners and their techniques.

## Conclusion

It is rare to conduct a digital forensic investigation that does not include a Smartphone or mobile device. Most of the times, the Smartphone may be the only source of digital evidence tracing an individual's movements and motives and may provide access to the who, what, when, where, why, and how behind a case. A successful forensic investigation can be achieved after collecting all the above "clues" and manage to find evidence. After that, a documentation along with an audit file with the tools used is preferred.

Proper documentation is essential in providing individuals the ability to re-create the process from beginning to end. Furthermore, it has to be formed in a simple way to be understandable by most of people. As part of the reporting process, making a copy of the software used and including it with the output produced is advisable when custom tools are used for examination or analysis. Reporting has to be done on case to case basis. There are different ways of reporting evidence in corporate cases and criminal cases. Reported evidence should be clear, give direct or indirect reference to the possible scenarios of crime.

Digital evidence, as well as the tools, techniques and methodologies used in an examination is subject to being challenged in a court of law or other formal proceedings. In a criminal case, where evidence should be presented in the court of law, it is also required to map the findings with respective laws. In addition to evidence, it is also required to present Chain of Custody.

This project, prepared a methodology of Android devices forensic investigation finding "holes" in the device demonstrating probable guilt of the "user" alongside with evidence that could prove any illegal activity. Comparing the two basic types of memory investigation-logical and physical- both yielded significant outcome

but in most sophisticated users investigation, the physical memory acquisition would probably help the examiner to gain much more forensic evidence and material. The reason is the ability of physical examination to retrieve deleted data. This kind of data will have much greater value from existing data on device.

To summarize, analyzing Android for forensic purpose employs totally different techniques than the traditional forensics. It involves heavy manual intelligence and interference. Maintaining integrity of primary evidence is also a challenge. There are tools available in the market for Android Forensics but still there are gaps to be filled and a lot to be done in this direction.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## References

1. Android Forensics Investigation, Analysis, and Mobile Security for Google Android, Andrew Hoog, 2011 Elsevier, Inc
2. Live Memory Forensics on Android with Volatility, Holger Macht, January 2013, Department of Computer Science
3. Android Forensics: Simplifying Cell Phone Examinations, Jeff Lessard, Gary C. Kessler, September 2010
4. Practical Mobile Forensics, Satish Bommisetty, Rohit Tamma, May 2014
5. Acquisition and analysis of volatile memory from android devices, Joe Sylve, Andrew Case, Lodovico Marziale, Golden G. Richard, Department of Computer Science, University of New Orleans, New Orleans, 2012, USA
6. Android Memory Capture and Applications for Security and Privacy, Joseph T . Sylve, University of New Orleans, 2011
7. Forensic analysis of mobile phone internal memory, Svein Y. Willassen , Norwegian University of Science and Technology
8. Mobile Device Forensics, Andrew Martin, Joey Niem , August 29, 2008
9. Data Carving Concepts, Antonio Merola, Rick Wanner, November 10th 2008
10. Analysis of the Android Architecture, Stefan Brahler, 2010
11. Android Forensics: Automated Data Collection and Reporting from a Mobile Device, Justin Grover, 2013
12. Understanding the Android File Hierarchy, Bill Anderson, 2013, <http://www.all-things-android.com/content/understanding-android-file-hierarchy>
13. Tools: Memory Imaging, Forensics Wiki, 2013, [http://www.forensicswiki.org/wiki/Tools:Memory\\_Imaging](http://www.forensicswiki.org/wiki/Tools:Memory_Imaging)
14. Mobile Device Forensics, Wikipedia, 2013, [http://en.wikipedia.org/wiki/Mobile\\_device\\_forensics](http://en.wikipedia.org/wiki/Mobile_device_forensics)
15. Comparison of Android Devices, [http://en.wikipedia.org/wiki/Comparison\\_of\\_Android\\_devices](http://en.wikipedia.org/wiki/Comparison_of_Android_devices)

16. Types of Android Devices,  
<http://www.androidauthority.com/android-everywhere-10-types-of-devices-that-android-is-making-better-57012/>
17. Android File System Structure/Architecture/Layout Details,  
<http://techblogon.com/android-file-system-structure-architecture-layout-details/>
18. Data at rest, Data in use, Data in transit,  
<http://padraic2112.wordpress.com/2007/07/26/data-at-rest-data-in-transit-data-in-use/>
19. Mobile Device Forensics, Richard Ayers Presentation
20. Mobile Forensics, Javier Martinez Presentation
21. Foremost Tool, <http://foremost.sourceforge.net/>
22. Android Forensics: Cracking the Pattern Lock Protection,  
<http://resources.infosecinstitute.com/android-forensics-cracking-the-pattern-lock-protection/>
23. "Frost" Attack Unlocks Android Phones' Data By Chilling Their Memory In A Freezer, Andy Greenberg, 2013,  
<http://www.forbes.com/sites/andygreenberg/2013/02/14/frost-attack-unlocks-android-phones-data-by-chilling-its-memory-in-a-freezer/>
24. Android Dominates Market Share But Apple Makes All The Money, Tony Bradley, 2013, <http://www.forbes.com/sites/tonybradley/2013/11/15/android-dominates-market-share-but-apple-makes-all-the-money/>
25. Android Debug Bridge, <http://developer.android.com/tools/help/adb.html>