2011

Acquisition of digital evidence in android smartphones

Andre Morum de L. Simao
Brazilian Federal Police

Fabio Caus Sicoli
Brazilian Federal Police

Laerte Peotta de Melo
University of Brasilia

Rafael Timoteo de Sousa Junior
University of Brasilia

DOI: 10.4225/75/57b2c3dc40cf3
Originally published in the Proceedings of the 9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, 5th -7th December 2011

ACQUISITION OF DIGITAL EVIDENCE IN ANDROID SMARTPHONES

Andre Morum de L. Simao[1], Fabio Caus Sicoli[1], Laerte Peotta de Melo[2],
Flavio Elias de Deus[2], Rafael Timoteo de Sousa Junior[2]
[1] Brazilian Federal Police, Ministry of Justice
[2] University of Brasilia, UnB
[1,2] Brasilia, Brazil
[1] {morum.amls, sicoli.fcs}@dpf.gov.br
[2] {peotta, flavioelias, desousa}@unb.br

Abstract

From an expert's perspective, an Android phone is a large data repository that can be stored either locally or remotely. Besides, its platform allows analysts to acquire device data, collecting information about its owner and facts that are under investigation. This way, by exploring and cross referencing that rich data source, one can get information related to unlawful acts and its perpetrator. There are widespread and well documented approaches to forensic examining mobile devices and computers. Nevertheless, they are not specific nor detailed enough to examine modern smartphones, since these devices have internal memories whose removal or mirroring procedures are considered invasive and complex, due to difficulties in having direct hardware access. Furthermore, specific features of each smartphone platform have to be considered prior to acquiring its data. In order to deal with those challenges, this paper proposes a method to perform data acquisition of Android smartphones, regardless of version and manufacturer. The proposed approach takes into account existing techniques of computer and cell phone forensic examination, adapting them to specific Android characteristics, its data storage structure, popular applications and the conditions under which the device was sent to the forensic examiner. The method was defined in a broad fashion, not naming specific tools or techniques. Then, it was deployed into the examination of six Android smartphones, addressing different scenarios that an analyst might face, and was validated to perform an entire evidence acquisition.

Keywords

Forensics; data acquisition; evidence analysis; mobile phone; smartphone; Android.

INTRODUCTION

In 2011, the Android operating system has exceeded the number of handsets sold in other systems for smartphones (Canalysis, 2011). The system has a wide acceptance in the market and it is conjectured that this is due to being open source and supporting the latest features and applications available for this type of mobile equipment. Given its ability to provide a large number of features to the user, a smartphone with the Android operating system can store a significant amount of information about its owner, being a source of evidence for facts one wants to clarify or to obtain information to support an investigation (Rossi, M., 2008).

Some of the features of Android are: browsing the Internet, image and video capture, creating and viewing documents, notes, calendar, contact management, GPS location and map navigation. From an application developer's viewpoint, Android has features that facilitate the development, publication and installation of created applications, what enhances the functionalities provided by the platform.

Unlike the data acquisition approach for computer environments, when data can usually be extracted in the same state it was found and is preserved from the time of its seizure, extraction of data from smartphones typically require intervening on the device. Moreover, given that they use embedded memories, whose direct hardware access is delicate and complex, sometimes it's needed to install applications or use tools directly on the device to proceed with the stored data acquisition. Thus, the analyst must have the expert knowledge needed to carry out forensic procedures on the device the least intrusive manner possible, controlling the environment in order to avoid loss, alteration or contamination of evidence data (Association of Chief Police Officers, 2008), which will give reliability to the forensic procedure.

ANDROID PLATFORM

The Android platform consists of the operating system, the SDK (Software Development Kit) and its applications. The SDK is a set of tools provided by the Google company that provides a development environment for creating Android applications. One of the tools is the Android Debug Bridge (ADB), which provides a communication interface to an Android system using a computer. When connected through this interface, a computer is able to access a command shell, install or remove applications, read log files, transfer files between the station and the device, among other actions.

The Android operating system uses the sandbox concept, where applications have reserved areas, with isolated process execution environments and limited access to resources. This way, applications cannot access areas that are not explicitly allowed (Google Inc., 2011). However, access to features may be authorized by the permissions set in the "AndroidManifest.xml" file. At the time of application installation, that file tells the user what resources will be used on the smartphone. He can accept the installation of the application after being informed of the resources that will be used or simply refuse the installation, if he does not agree with the features that the application wishes to access.

Another feature of the Android OS is the use of the SQLite database, which is free and open source. It is an easy to use relational database, which stores in a single file the complete structure of data objects (tables, views, indexes, triggers) (SQLite, 2011). Such database does not need any configurations and uses file system permissions to control access to its stored data.

Regarding the file system, currently most of Android devices adopt YAFFS2 (Yet Another Flash File System 2), which is a file system designed for flash memory and its peculiarities. It is worth noting that the major forensic tools available in the market are not compatible with that file system, making it difficult to mount Android partitions and access data stored there. However, as quoted by Andrew Hoog (Hoog, A., 2011) in late 2010, it was observed that some Android handsets were already using the EXT4 (Fourth Extended File System). There is a migration tendency to this file system, in order to support dual-core processor and multiprocessing, and to use e-MMC memories, (Embedded MultiMediaCard), which already work simulating block storage devices, that are more robust, mature and have more commercial acceptance.

Access to system partitions is restricted to the Android operating system. By default, users do not have permission to access system reserved areas. The system is shielded in order to prevent malicious or poorly developed applications to affect the OS's stability and reliability. However, it is possible to exploit a set of system or device vulnerabilities to obtain super user (root) privileges. Thus, it is possible to use applications or a shell that has full and unrestricted access to the system. As a result, a forensic analyst can make a mirror copy of all of the system partitions as well as access files which were not accessible by using Android conventional credentials. The techniques vary depending on each Android version and may also depend on the device make and model. Moreover, those techniques are often invasive and may even damage data stored on the device, so they should be used carefully.

The operating system has authentication mechanisms that use of passwords, tactile patterns or biometric information. According to the NIST guide on cell phones forensics (Jansen, W.; Ayers, R., 2007), there are three possible methods to unlock a device: investigative, software-based or hardware-based. Those can be applied to Android equipment depending on the seizure circumstances, device model and system version.

Given the characteristics described, in order to conduct a forensic data extraction, besides having knowledge about the Android platform, an analyst should evaluate the procedures to be adopted. For instance, there are scenarios in which the phone may be turned on or off, have internal or removable memory, be locked or unlocked, have access through USB debug mode or not, have some applications running that contain useful information for an investigation, and even may have root privileges enabled. Thus, the analyst must assess the correct procedures to be adopted depending on the Android smartphone status.

DATA ACQUISITION METHOD FOR ANDROID SMARTPHONES

Considering the unique characteristics of the Android platform and different scenarios which a forensic analyst may come across, a data acquisition method is proposed and its workflow is shown in Figure 1. In the figure, different scenarios and their respective procedures to be adopted by the analyst are presented. By using the proposed method, a forensic analyst may retrieve maximum information from the mobile device, so that the evidence may be documented, preserved and processed in the safest and least intrusive manner as possible.

Initial procedures for preservation of the data in a smartphone

Upon receiving a smartphone, the forensic analyst must follow the procedures in order to preserve the stored data on the seized equipment. So, he should check if the phone is turned on or not. If the phone is powered off, one should evaluate the possibility of extracting data from its memory card. It should be pointed that some Android phones have an internal memory card, so it is not possible to remove it in order to copy its data through the use of a standard USB card reader. On the other hand, if it is feasible to remove the memory card, it should be removed and duplicated to an analyst memory card to ensure its preservation. To copy data from the memory card, one may use the same approach used with pen drives. The forensic expert could use forensic tools to copy the data or even run a disk dump and then generate the hash of the duplicate data. At the end of the process, the analyst' memory card with the copy should be returned to the device.

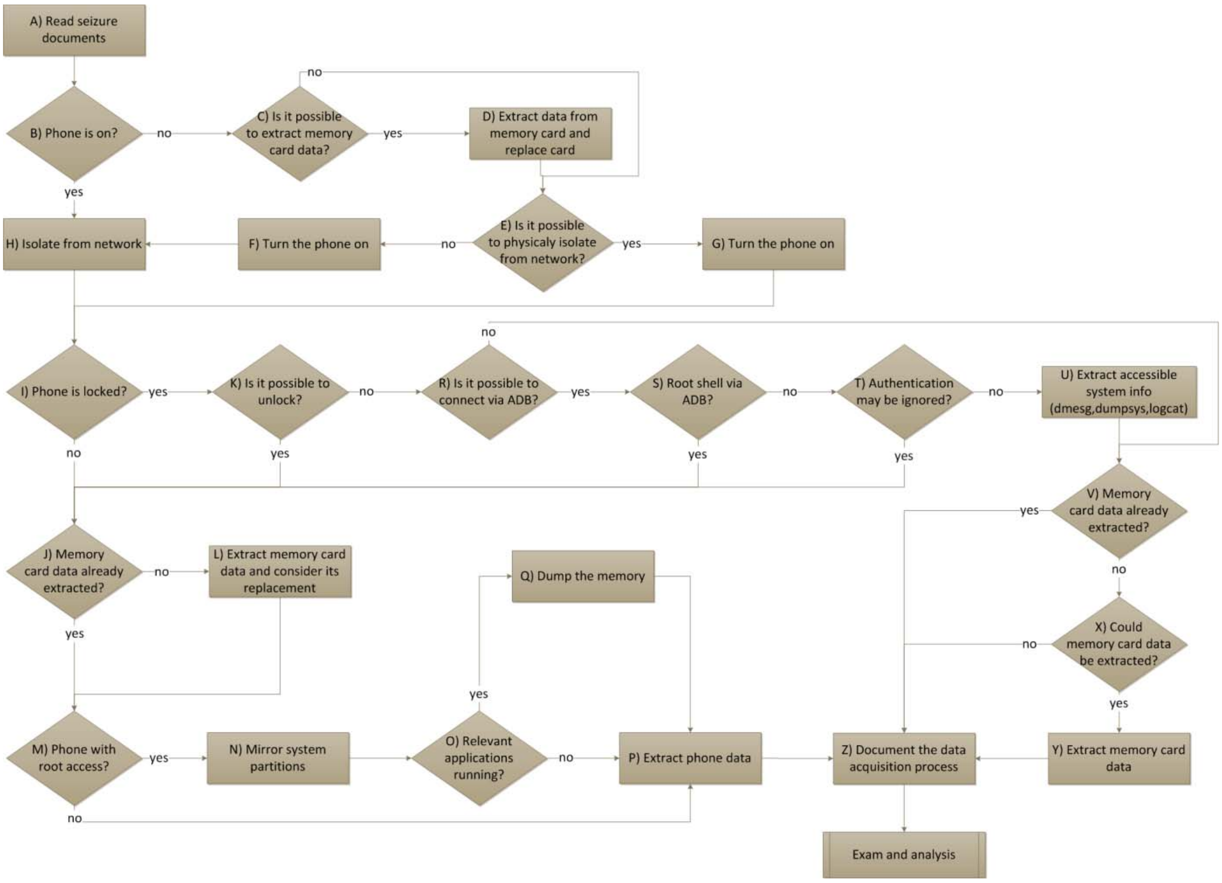
A) Read seizure documents
B) Phone is on?
no
C) Is it possible to extract memory card data?
no
yes
D) Extract data from memory card and replace card
yes
H) Isolate from network
F) Turn the phone on
no
E) Is it possible to physicaly isolate from network?
yes
G) Turn the phone on
no
I) Phone is locked?
yes
K) Is it possible to unlock?
no
R) Is it possible to connect via ADB?
yes
S) Root shell via ADB?
no
T) Authentication may be ignored?
no
U) Extract accessible system info (dmesg,dumpsys,logcat)
no
yes
yes
yes
V) Memory card data already extracted?
yes
no
J) Memory card data already extracted?
no
L) Extract memory card data and consider its replacement
Q) Dump the memory
yes
X) Could memory card data be extracted?
no
yes
yes
M) Phone with root access?
yes
N) Mirror system partitions
O) Relevant applications running?
no
P) Extract phone data
Z) Document the data acquisition process
Y) Extract memory card data
no
Exam and analysis

Figure 1 - Workflow with the process of acquiring data from a smartphone with the Android operating system.

The next step is to isolate the telephone from the network. The ideal situation is to use a room with physical insulation from electromagnetic signals. However, when one does not have such an infrastructure, he should set the smartphone to flight or offline mode. From the moment the power is on, he must immediately configure it to such connectionless mode, thus avoiding data transmission, receiving calls or SMS (Short Message Service) after the equipment seizure time. If by any chance, before it is isolated from the network, the phone receives an incoming call, message, email or other information, the analyst should document and describe it in its final report, which will be written after the data extraction, examination and analysis processes.

With the smartphone isolated from telecommunication networks, the forensic analyst should check if the Android has been configured to provide an authentication mechanism, such as a password or tactile pattern. Afterwards, he should carry out the procedures described in the following sections, which depend on the access control mechanism which is configured on the device.

Smartphone without access control

The least complex situation that an examiner may encounter is one which the mobile is not locked and is readily able to have its data extracted. In this situation, one must first extract data from memory cards, if they have not been copied, and in case of removable memory cards, reinstall into the device the cards that have received the copies, preserving the original ones.

With the data from memory cards extracted and properly preserved, the examiner should check if the Android has super user privileges enabled. The application called "Superuser" can be installed to provide access to such privileges. From the moment the analyst is faced with an Android phone with super user privileges, he can gain access to all stored data in the device without any restriction. Using the USB debugging tool ADB (Android Debug Bridge), present in the Android SDK, one can connect to the device, access a command shell with super user privileges and make a copy of the system partitions stored in its internal memory, as illustrated in Figure 2.

```
C:\Android\android-sdk\platform-tools>adb devices
List of devices attached
040140611301E014        device

C:\Android\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
# mount | grep mtd
mount | grep mtd
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0# cat /proc/mtd
cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
# ls /dev/mtd/mtd*
ls /dev/mtd/mtd*
…
/dev/mtd/mtd6
/dev/mtd/mtd6ro
/dev/mtd/mtd7
/dev/mtd/mtd7ro
/dev/mtd/mtd8
/dev/mtd/mtd8ro
…
# dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/PERICIA/mtd6ro_system.dd bs=4096
dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/PERICIA/mtd6ro_system.dd bs=4096
44960+0 records in
44960+0 records out
184156160 bytes transferred in 73.803 secs (2495239 bytes/sec)
# dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/PERICIA/mtd7ro_cache.dd bs=4096
dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/PERICIA/mtd7ro_cache.dd bs=4096
27136+0 records in
27136+0 records out
111149056 bytes transferred in 41.924 secs (2651203 bytes/sec)
# dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/_PERICIA/mtd8ro_userdata.dd bs=4096
dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/_PERICIA/mtd8ro_userdata.dd bs=4096
50464+0 records in
50464+0 records out
206700544 bytes transferred in 74.452 secs (2776292 bytes/sec)
# ls /mnt/sdcard/PERICIA
ls /mnt/sdcard/PERICIA
mtd6ro_system.dd
mtd7ro_cache.dd
mtd8ro_userdata.dd
```

Figure 2 - Commands to list connected devices, display partition information, and generate the partitions dump.

It should be pointed that, by carrying out procedure described in Figure 2, the mirrored partition images will be written to the memory card which is installed in the device. In some situations, it may not be possible to replace the original memory card by an analysts one. Nevertheless, regardless of its replacement, removable media's data must have been mirrored prior to the system mirroring and copying. By doing that, data stored in the original memory card, seized with the smartphone, are preserved and the forensic expert should point that in his report that will be produced by the end of data analysis.

After mirroring the partitions, one should observe the running processes and assess the need to get run-time information, which is loaded in the device's memory. Hence, it is possible to extract memory data used by running applications to access sensitive information, such as passwords and cryptographic keys. By using a command shell with super user credentials, the “/data/misc” directory's permissions must be changed. Afterwards, one must kill the target running process, so that a memory dump file for the killed process is created (Cannon, T., 2010). Data extraction of a telephone with available "super user" credentials may be finished in this moment. Figure 3 displays the technique described by Thomas Cannon (Cannon, T., 2010).

```
# chmod 777 /data/misc
chmod 777 /data/misc
# kill -10 6440
kill -10 6440
# kill -10 6379
kill -10 6379
# kill -10 6199
kill -10 6199
# kill -10 5797
kill -10 5797
# ls /data/misc | grep dump
ls /data/misc | grep dump
heap-dump-tm1303909649-pid5797.hprof
heap-dump-tm1303909632-pid6199.hprof
heap-dump-tm1303909626-pid6379.hprof
heap-dump-tm1303909585-pid6440.hprof
#
…
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /data/misc/heap-dump-
tm1303909649-pid5797.hprof
2206 KB/s (2773648 bytes in 1.227s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /data/misc/heap-dump-
tm1303909632-pid6199.hprof
2236 KB/s (3548142 bytes in 1.549s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /data/misc/heap-dump-
tm1303909626-pid6379.hprof
1973 KB/s (3596506 bytes in 1.779s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /data/misc/heap-dump-
tm1303909585-pid6440.hprof
1968 KB/s (2892848 bytes in 1.435s)
```

Figure 3 - Presents the commands to alter a directory's permissions, kill processes to create processes' memory dump files and copy those files to the analyst's station.

It is noteworthy that, in order to inspect the acquired data, the analyst should have an examination environment with tools that are capable of mounting images having the device's file system, which is in most cases the YAFFS2. The technique described by Andrew Hoog may be used to examine that file system (Hoog, A., 2011). Nevertheless, it is recommended that a logical copy of system files is made directly to the analyst's workstation, as shown in Figure 4.

```
C:\android-sdk\platform-tools> adb pull /data pericia/
Pull: building file list…
…
684 files pulled. 0 files skipped
857 KB/s (194876514 bytes in 226.941s)
```

Figure 4 – Copy of logical files stored in the device's "/data" directory to the "pericia" directory in the analyst's workstation.

The stored data in the "/data" directory, for instance, contain information regarding the installed applications and system configuration. Logical copy of files will create redundancy that may be useful during the examination phase, especially in situations when it is not necessary to delve into system partitions. In addition, some applications may be active in the system, in such a way that a simple visual inspection may provide information which would be difficult to access by means of analyzing the created image. Moreover, forensic extraction tools may be used to interpret stored data.

In situations when the smartphone "super user" privileges are not available, data extraction from its internal memory should be carried out by visually inspecting and navigating through the device graphic user interface. Alternatively, forensic tools and applications may be used to assist the analyst in extracting device's data. Nevertheless, it is important to check the information gathered by such tools, because the Android OS has different versions, as well as manufacturer and telephone carrier customizations, which may interfere in the automated tools' proper functioning. That are numerous applications that may store meaningful information for an investigation, whose data extraction is not supported by forensic tools. It is clear that the forensic analyst needs to have proper knowledge regarding the Android platform and its applications, since relevant information extraction should be conducted in the most complete way.

Some Android smartphones allow that their internal memories be copied using boot loader or recovery partitions vulnerabilities, without having "super user" credentials. It is up to the analyst to evaluate if it is possible and viable to apply such techniques for that kind of device. It is suggested that the investigation team should discuss the need of using such techniques and consider the risks and impacts to the exam results.

Regarding existing forensic tools, the viaForensics company developed a free tool to law enforcement agencies called “Android Forensic Logical Application” (AFLogical) (ViaForensics, 2011), whose goal is to extract information from Android smartphones. In addition, recently the commercial tool viaExtract was released and, according to viaForensics, it has more consistent and relevant features, such as generating reports. Another very useful tool is the "Cellebrite UFED", whose version 1.1.7.5, released in July of 2011, carry out physical extraction from a few models without the need of "super user" privileges. The same tool has a plugin to view Android's SQLite databases and other application files, such as Gmail, SMS, MMS and contacts.

Smartphone with access control

In the likely event the Android smartphone has access control, such as a password or tactile pattern, there are still techniques to be used to access the device.

According to NIST (Jansen, W., Ayers, R., 2007), there are three ways of gaining access to locked devices. The first one is the investigative method, whereby the researcher seeks possible passwords in the place where the smartphone was seized or even interview the its alleged owner so that he cooperates voluntarily by providing his password. Another way is to gain access via hardware, when the analyst performs a research on that specific given model to determine whether it is possible to perform a non-destructive procedure in order to access device data. In this sense, one may request support from manufacturers and authorized service centers. Finally, there are methods of software access that, even though it depends on the handset model and Android version, are usually the easiest way and can be applied in the forensic analyst's own test environment.

To access the system, the analyst must do it the least intrusive manner possible in order to avoid compromising the evidence. If the password or the tactile pattern has been obtained when the device was seized, those should be readily tested. Alternatively, one may use the technique to find the tactile pattern by means of examining the smudge left on the device screen (Aviv, A. J. et al, 2010), before attempting any other way to bypass access control, preventing screen contamination.

If the analyst does not succeed, he should check if the Android is configured to accept USB debugging connections using a tool available in the SDK, the ADB. If he succeeds, he attempts to obtain "super user" access credentials to resume the acquisition process, the same way that it would be performed in cases where the mobile device was not locked, because with such permissions, one can get all the stored data in the device, as described previously.

Even when there is no "super user" access to the handset, it is still possible to install applications through the ADB tool to overcome the access control system. The technique described by Thomas Cannon (Cannon, T., 2011) is to install the "Screen Lock Bypass" application, available at the Android Market. In this technique, one needs that the Google account password be saved in the Android device, as well as Internet access be enabled, which is considered inadvisable. In this sense, it is recommended that the application is downloaded from another Android device and then installed via ADB on the examined mobile device. Thus, it is possible to perform the screen unlock using Cannon's technique without the need of having the device's Google account password or connecting it to the web. Figure 5 shows Cannon's application installation, as well as its activation, which depends on the installation of any other application, to perform access control unlocking.

```
C:\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
Permission denied
$ exit
...
C:\android-sdk\platform-tools>adb -s 040140611301E014 install screenlockbypass.apk
224 KB/s (22797 bytes in 0.100s)
        pkg: /data/local/tmp/screenlockbypass.apk
Success
C:\android-sdk\platform-tools>adb -s 040140611301E014 install AndroidForensics.apk
716 KB/s (31558 bytes in 0.046s)
        pkg: /data/local/tmp/AndroidForensics.apk
Success
```

Figure 5 – Connection via ADB, root access check and application installation in order to ignore access control.

In situations where it is not possible to bypass the authentication system or USB debugging access is disabled, it is left to the analyst to copy the data contained in the removable memory card that may be installed on the handset. In those situations, it is very important to report the impossibility to access the device with the used procedures. In addition, if there is another technique that may be applied, be it more invasive or complex, that fact should be informed to whom requested the exams. Consequently, the implications of applying such techniques should be discussed, considering the risks to the given situation, such as permanent damages to the examined smartphone.

Acquisition documentation

It is recommended that all the techniques and procedures used by the analyst be documented, in order to facilitate the examination and analysis of extracted data. Regardless of the path followed by the expert in the workflow illustrated in Figure 1, the process should be recorded, enabling auditability and reliability of the procedures performed by the expert analyst.

The analyst should be careful to register the hash codes of data generated and extracted during the acquisition process, as well as state in his report any caveats that he considers important to carry out the examination and analysis stage, like an e-mail or SMS received before the smartphone have been isolated from telecommunication networks or even the existence of applications that contain information stored in servers on the Internet, such as cloud computing.

The forensic expert, while executing his activities, should consider that the better reported the acquisition process, the more trust will be given to the examination results. The simple condition of the processes be well documented is the first step to conduct an impartial, clear and objective data analysis.

VALIDATION OF THE PROPOSED ACQUISITION METHOD

The proposed method was tested by using a sample containing six smartphones that had the Android OS. Among those handsets, four different scenarios were identified, summed up and presented in Table 1.

Table 1 - Scenarios used to validate the proposed method.

Scenario	Turned on	Removable card	Locked	Unlockable	Super user
1st Scenario (Motorola Milestone II A953)	No	Yes	Yes	Yes	No
2nd Scenario (Sony Ericson Xperia X10 miniPro)	Yes	No	No	Does not apply	No
3rd Scenario (Motorola Defy) (Samsung Galaxy S 9000 [a])	No	Yes	No	Does not apply	Yes
4th Scenario (Motorola I1) (Motorola Milestone A853)	No	Yes	No	Does not apply	No

[a] In addition to the removable microSD card, that phone has a built-in memory card which is not removable.

In the first scenario, as the device was turned off, first its memory card was removed and mirrored. Then, the memory card holding the copy was inserted into the handset. Subsequently, the smartphone was switched on and set immediately in flight mode. It was noticed that the cell phone was locked, but its USB debugging access was enabled. By using the ADB tool, a shell was obtained but that there were no "super user" permissions available, preventing mirroring system partitions. However, from the ADB, it was possible to install the "Screen Lock Bypass" application (Cannon, T., 2011), which was used to unlock the device, as well as the "Logical Android Application Forensics", a data extraction tool (ViaForensics, 2011). In addition, the extracted data were visually inspected.

In the following scenario, the smartphone was not locked and was put into flight mode in order to isolate it from the network. The device in this scenario had a memory card that was not removable. The card data were mirrored (copied entirely), and its own memory was used to extract its information by the "Logical Android Application Forensics" forensic software. Afterwards, data were extracted the same way as the previous scenario.

The same way as done in the first scenario, in the third, the memory card was removed and replaced by a mirror, since the device was received turned off. Later, the smartphone was turned on and immediately put into flight mode. It was observed that the mobile device was unlocked and also had a second memory card embedded. That memory card was also mirrored. The smartphone had the "Superuser" application, which provides super user credentials. Thus, the USB debug mode was enabled, an ADB was established, obtaining a shell with "super user" permissions to carry out the system partitions mirroring. RAM data were not copied, because the handset was received switched off and the analysts considered unnecessary to perform such procedure. Then, the Cellebrite UFED System 1.1.7 tool was used to extract forensic data from the phone, followed by visual inspection to complement the extracted data.

Last but no least, in the fourth scenario, the memory card was removed, mirrored and replaced while the device was still turned off. Then, the phone was turned on and immediately put into flight mode. The phone was unlocked. Thus, the Cellebrite UFED System 1.1.7 tool was used to extract forensic data from the phone, with subsequent visual inspection to complement the extracted data.

The procedures cited in the method could be directly translated into actions performed onto the examined devices. Thus, it was possible to perform data acquisition of every tested smartphone, demonstrating the suitability and validity of the proposed method for each encountered scenario.

CONCLUSION

The Android smartphone platform is already the most present among mobile communication devices. However, the existing approaches to forensic examine cell phones and computers are not completely adequate to the peculiarities of that class of devices. Moreover, the existing models of forensic analysis on cell phones do not consider the peculiarities of each platform.

A specific method was proposed to address data acquisition of devices that use the Android Platform, taking into account operational system characteristics, its most popular applications and devices hardware features.

By means of defining an Android system data acquisition method, it was possible to foresee the difficulties a forensic expert might face, preparing them to perform an entire evidence acquisition, given the situation the handset was forwarded, avoiding mishaps in the data extraction process and missing forensic evidence.

The method was proposed in a broad fashion, so that the techniques, procedures and specific tools chosen by the analyst during the workflow do not interfere with its application. So, as new techniques arise, with different approaches to perform a given task, such as unlocking the device, bypassing access control or mirroring partitions, they will be covered by the proposed method, which focuses on the result that each activity produces.

The proposed method was validated by its application onto the examination of six Android smartphones, which were grouped into four scenarios, involving different situations that an analyst may encounter.

For future work, it is proposed that the method is validated for the Android 3, evaluating its effectiveness in the Google system for tablet devices, making the adjustments that may be required. Another interesting work to be developed would be the creation of a forensic tool that supported the YAFFS2 file system, focused on NAND flash memory, facilitating data extraction and access and also mounting images from those storage media.

REFERENCES

Association of Chief Police Officers. “Good Practice Guide for Computer-Based Electronic Evidence”. Version 4.0. [S.l.]. 2008.

Aviv, A. J.; Gibson, Katherine; Mossop, Evan; Blaze, Matt; Smith, Jonathan M. “Smudge Attacks on Smartphone Touch Screens”. 4th Workshop on Offensive Technologies. Washington, DC: [s.n.]. 2010.

Canalys. “Android takes almost 50% share of worldwide smartphone market”. Canalys website, 1/8/2011. Available at: <http://www.canalys.com/newsroom/android-takes-almost-50-share-worldwide-smart-phone-market>. Accessed: 3 august 2011.

Cannon, T. “Android Lock Screen Bypass”. Thomas Cannon, 2011. Available at: <http://thomascannon.net/blog/2011/02/android-lock-screen-bypass/>. Accessed: 23 march 2011.

Cannon, T. “Android Reverse Engineering”. Thomas Cannon, 2010. Available at: <http://thomascannon.net/projects/android-reversing/>. Accessed: 23 march 2011.

Google Inc. “Android Fundamentals”. Android Developers, 2011. Available at: <http://developer.android.com/guide/topics/ fundamentals.html>. Accessed: 17 march 2011.

Hoog, A. “Android Forensics - Investigation, Analisys and Mobile Security for Google Android”. 1st. ed. [S.l.]: Syngress, 2011.

Jansen, W.; Ayers, R. “Guidelines on Cell Phone Forensics - Recomendations of the National Institute of Standards and Technology”. [S.l.]. 2007.

Rossi, M. “Internal Forensic Acquisition for Mobile Equipments”, n. 978-1-4244-1694-3. IEEE, 2008.

SQLite. “About SQLite”. SQLite, 2011. Available at: <http://www.sqlite.org/about.html>. Accessed: 5 april 2011.

ViaForensics. “Android Forensics Logical Application (LE Restricted)”. ViaForensics website, 2011. Available at: <http://viaforensics.com/ android-forensics/android-forensics-logical-application-le-restricted.html>. Accessed: 03 august 2011.