An Open Framework for Smartphone Evidence Acquisition

Lamine Aouad, Tahar Kechadi, Justin Trentesaux, Nhien-An Le-Khac

Chapter 11

AN OPEN FRAMEWORK FOR SMARTPHONE EVIDENCE ACQUISITION

Lamine Aouad, Tahar Kechadi, Justin Trentesaux and Nhien-An Le-Khac

Abstract The forensic processes and procedures for performing a bit-for-bit copy of computer memory and media are well established and documented. However, most commercially-available tools for smartphone data acquisition and analysis have unspecified implementations and do not provide technical documentation about the acquisition process. A detailed understanding of the acquisition process is important in criminal cases because the technique that was used may be questioned in courtroom proceedings. To address the gap, this paper proposes an open framework for documenting and implementing the acquisition and analysis of digital evidence from smartphones.

Keywords: Mobile phone forensics, evidence acquisition, open framework

1. Introduction

Smartphones have become an integral part of the daily lives of an increasing number of people from around the world. Smartphones are essentially pocket computers and, as such, contain a large amount of information. Information stored on smartphones such as the details of communications (calls, text and email messages), user locations, contacts, photographs and videos are all potentially relevant in forensic investigations.

Given the variety of devices and the lack of standards, there is no unified data acquisition method for smartphones. Data acquisition is one of the most important steps in a forensic investigation. Unfortunately, the majority of tools retrieve the baseline of extractable evidence (e.g.,

address book data, call details, text messages, photographs), but omit important evidence, especially deleted data. Additionally, because of the variety of communication protocols and proprietary interfaces, acquisition methods have yet to be formalized for many models of smartphones.

This paper attempts to address these issues by proposing an open and flexible framework designed to facilitate forensic investigations of mobile devices while supporting add-ons and mappings to existing protocols. The paper also presents an implementation of the framework that supports the capture of bit-for-bit copies of the memory and media of several Android smartphone models.

2. Background

In recent years, there has been increasing interest in mobile device forensics. Several studies have evaluated techniques and tools for acquiring digital evidence from Android and iPhone smartphones [2–4, 6]. Given the variety of devices – the Android operating system for instance is compliant with more than 300 smartphone models – it should come as no surprise that no standardized or generalized methods exist for acquiring and analyzing digital evidence from smartphones.

The tools analyzed in the studies include .XRY, EnCase Neutrino, CelleBrite UFED and Oxygen Forensic Suite. The studies have considered tool performance in terms of the support for and the accuracy of targeted data such as call logs, text messages, email, multimedia, maps, cookies and deleted data. However, practically all the tools that have been analyzed are commercial products with unspecified implementations and little or no documentation of their logical and physical acquisition methods. One of the keys goals of this work and, indeed, current research in mobile phone forensics, is to document the fundamentals of acquiring and analyzing evidence from smartphones.

3. Evidence Acquisition Methods

There are two main evidence acquisition methods: logical acquisition and physical acquisition. Logical acquisition methods interact with devices to extract storage objects such as directories or files using protocols such as AT commands, OBEX (OBject Exchange) or vendor interfaces. These methods typically extract data that is accessible via the operating system and interfaces; they cannot extract deleted data.

Physical acquisition methods extract the memory contents via the bit-by-bit imaging of the flash memory of a device. The method used depends on the manner in which a device stores data in its memory structures and can be performed using system access to the flash mem-

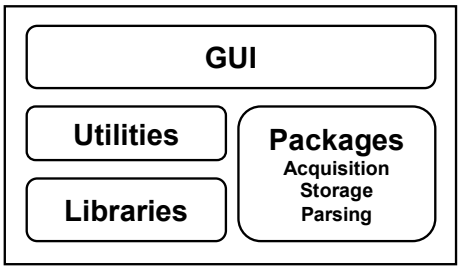

Figure 1. Open framework.

ory or using communications ports based on standards such as JTAG. However, the connections and interface specifications are not available for many smartphone models. Also, many vendors disable or lock access to device memory. Physical acquisition can also be done via direct memory chip access, but this requires specialized equipment and knowledge about the raw data structures. Also, the approach is very invasive and can damage the device and its memory.

While the forensic tools mentioned above can extract a variety of data from smartphones, a bit-by-bit copy of the original media is the most accurate representation of evidence on a device [4]. This memory image can be analyzed by a forensic investigator using *ad hoc* and/or standard techniques and tools.

4. Open Framework

Figure 1 presents a schematic diagram of the open framework. The basic functionality is provided by the acquisition, storage and parsing packages, all of which are implemented in C++. The basic methods in the packages are defined in a manner that permits the extension and/or re-definition of the core functionality. This is motivated by the desire to have a generic framework that would allow other programmers to add support for new devices and data sources.

Currently, application databases and memory images from a supported Android smartphone can be extracted only after rooting the device. Depending on the operating system version running on the smartphone, various exploits, such as `psneuter` or `gingerbreak`, may be used to gain root access. After the data is extracted, it is parsed into an XML format in order to define standard metadata.

Future versions will employ content providers to extract databases without rooting smartphones. Note, however, that content provider

methods do not extract deleted data, so recovering deleted data would still require rooting a smartphone.

4.1 Logical Recovery of Databases

Depending on the operating system variant, the framework browses folders in its search for databases and proceeds to recover them. In the case of a Nexus One device running OS version 2.1, the list of the database files includes (among others):

- `/data/data/com.android.browser/databases/browser.db`
- `/data/data/com.android.browser/databases/webviewCache.db`
- `/data/data/com.android.browser/databases/webview.db`
- `/data/data/com.android.providers.settings/databases/settings.db`
- `/data/data/com.android.providers.calendar/databases/calendar.db`
- `/data/data/com.android.providers.contacts/databases/contacts2.db`
- `/data/data/com.android.providers.downloads/databases/downloads.db`
- `/data/data/com.android.email/databases/EmailProviderBody.db`
- `/data/data/com.android.email/databases/EmailProvider.db`
- `/data/data/com.android.providers.telephony/databases/telephony.db`
- `/data/data/com.android.providers.telephony/databases/mmssms.db`
- `/data/data/com.google.android.gm/databases/gmail.db`
- `/data/data/com.google.android.apps.maps/databases/friends.db`
- `/data/data/com.google.android.apps.maps/databases/search_history.db`

The current version of the framework is able to query seven of these databases to extract contacts, call details, text messages, emails and pictures. Deleted data corresponding to contacts and calls are also recoverable. Future versions will be able to extract evidence from additional databases, including SQLite.

4.2 Physical Recovery of Images

The Android filesystem has a number of partitions that potentially hold data of interest in forensic investigations. These partitions are located in `/dev/mtd`. For example, a Nexus One device running OS version 2.1 has six partitions:

- `mtd0`: "misc"

- `mtd1`: "recovery"
- `mtd2`: "boot"
- `mtd3`: "system"
- `mtd4`: "cache"
- `mtd5`: "userdata"

The current implementation images all six partitions. Most of the data of interest resides in `mtd3` and `mtd5`. Note, however, that the partition order can vary for different smartphone models. The current version does not support the mounting and reading of images. However, investigators can use commercially-available forensic tools such as the Forensic ToolKit (FTK) to mount and browse images.

4.3 Experimental Tests

The current implementation of the framework has been tested on several Android phones and emulators with different operating system versions. After setting up a smartphone and choosing the extraction mode (Figure 2), the framework displays a summary and the status of the extraction process (Figure 3). The results are then displayed as shown in Figure 4. Functionality for browsing the extracted memory images and searching for data of interest will be implemented in a future version.

4.4 Forensic Recovery of iPhone Data

As part of the implementation effort, we have also investigated the hardware and system specifications of the iPhone. The iPhone has two partitions, `root` and `user`, which are mounted at `/` and `/private/var`, respectively.

The `root` partition is 300 MB in size and stores the operating system and pre-loaded applications. It is mounted as read-only and remains in a factory state by default. No user data of value to a forensic investigation can be retrieved from this partition, but it can be used to safely install forensic toolkits without affecting user data. The `user` partition contains data of interest to forensic investigations.

Various utilities can be used to access iPhone data depending, of course, on the firmware version. For firmware versions 1.0.2 to 1.1.4, the `iLiberty+` utility may be used. This free toolkit includes a basic Unix world, `OpenSSH`, `netcat`, `md5` and `dd`. It can also be used to remove the operating-system-level passcode.

In the case of iPhones with newer firmware versions, the `Pwnage` and `Xpwn` recovery tools may be used. The recovery is then performed using

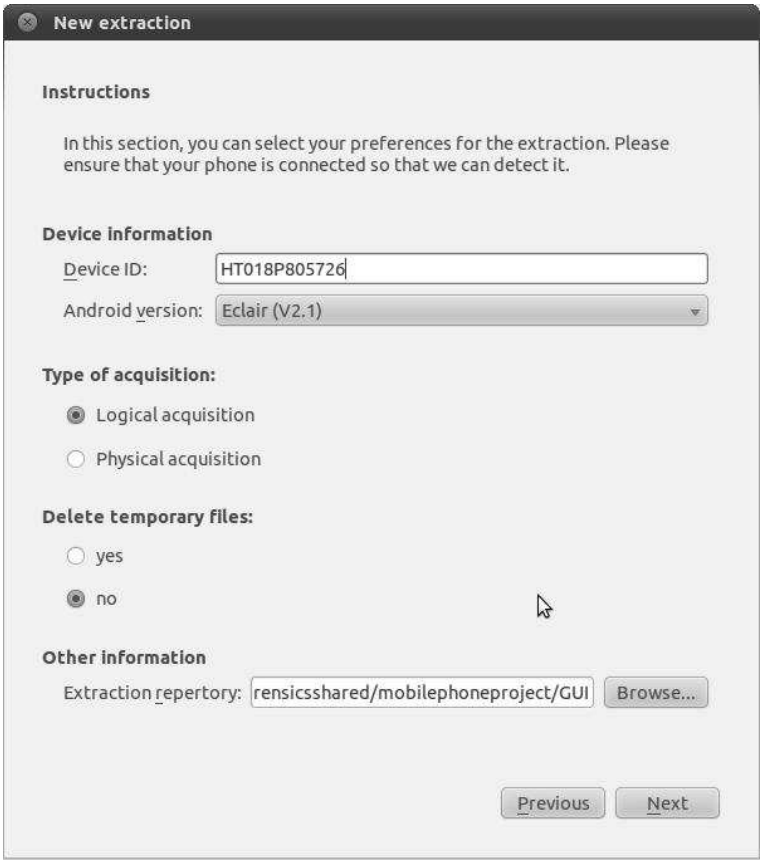

Figure 2. Selecting the extraction mode.

`ssh`, `dd` and `nc`. More details about recovering the media partition and deleted data can be found in [5]. Future versions of the open framework will incorporate these and other tools.

5. Conclusions

The open framework for evidence acquisition from smartphones is designed to facilitate forensic investigations of a variety of mobile devices while supporting add-ons and mappings to existing protocols. The implementation of the framework supports the capture of bit-for-bit copies of the memory and media of several Android smartphone models. Our future research will focus on implementing evidence extraction and anal-

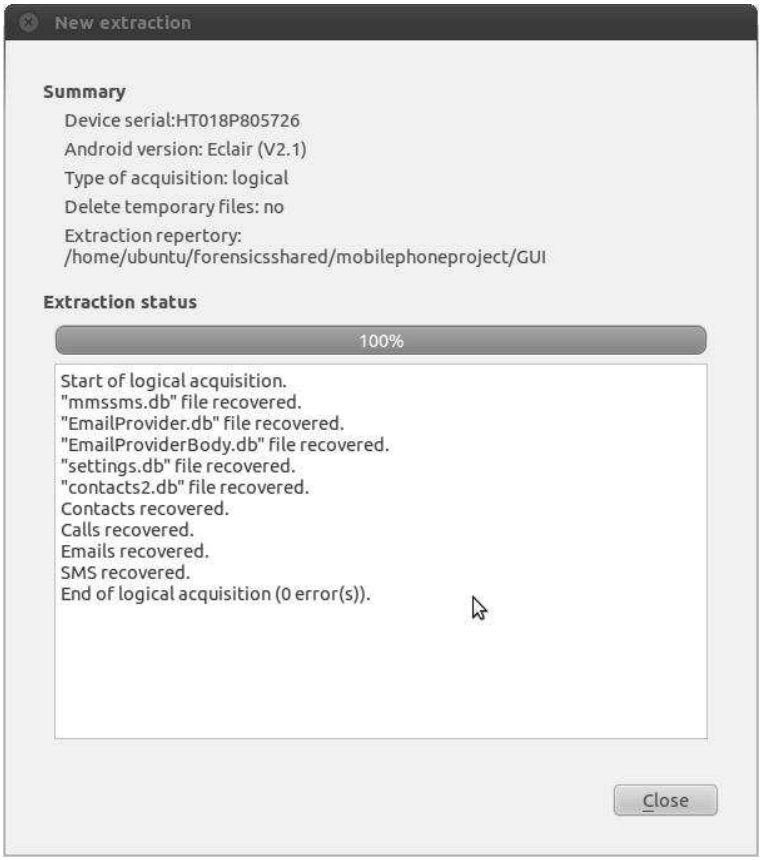

Figure 3. Performing the extraction.

ysis functionality for other Android models and other smartphone platforms.

References

[1] R. Ayers, W. Jansen, L. Moenner and A. Delaitre, Cell Phone Forensic Tools: An Overview and Analysis Update, NIST Technical Report NISTIR 7387, National Institute of Standards and Technology, Gaithersburg, Maryland, 2007.

[2] A. Hoog, Android forensics, presented at *Mobile Forensics World*, 2009.

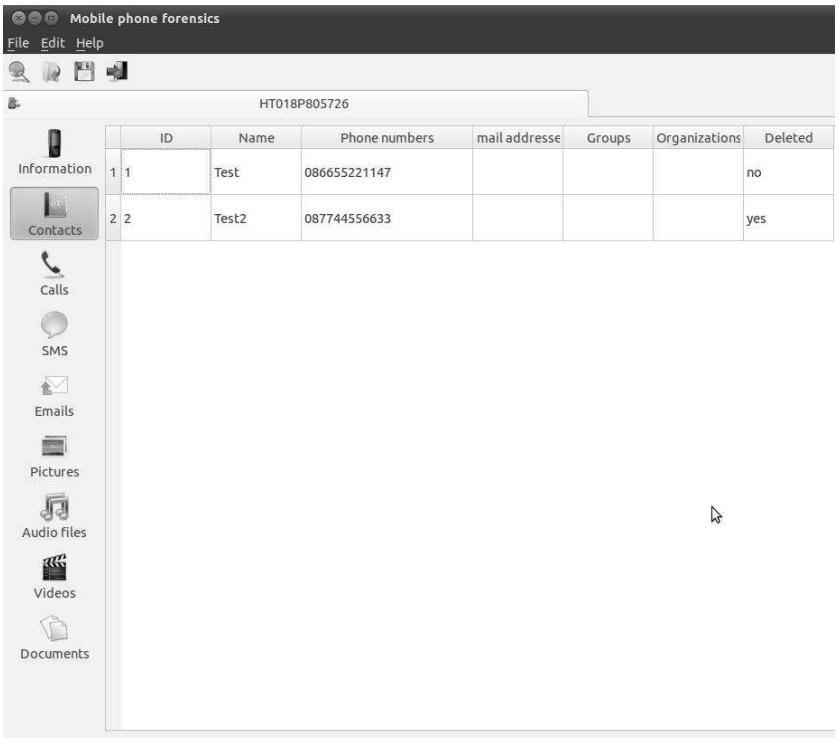

Figure 4. Displaying the results.

[3] A. Hoog and K. Strzempka, iPhone Forensics, White Paper, viaForensics, Oak Park, Illinois (`viaforensics.com/resources/white-papers/iphone-forensics`), 2010.

[4] W. Jansen and R. Ayers, Guidelines on Cell Phone Forensics: Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-101, National Institute of Standards and Technology, Gaithersburg, Maryland, 2007.

[5] N. Le-Khac, iPhone Forensics, Technical Report, School of Computer Science and Informatics, University College Dublin, Dublin, Ireland, 2011.

[6] V. Thing, K. Ng and E. Chang, Live memory forensics of mobile phones, *Digital Investigation*, vol. 7(S), pp. S74–S82, 2010.