# Anexo I. English summary

## Introduction

This document contains the final year project developed by Lucas González de Alba, a student from Universidad Carlos III de Madrid. This research project analyzes from a digital forensics standpoint, the structure, behavior, and artifacts created by Microsoft’s Your Phone application.

### 1.1 Abstract

Microsoft’s Your Phone is a service that facilitates user access to phone devices by integrating its notifications, messages, photos, and calls directly into Windows. All this data can have huge value on forensic investigation, so the work we present analyzes Your Phone app on the look for digital artifacts. The project began with large research of existing knowledge; that is articles, publications, blogs, forums...etc. After a brief overview of the app's behavior, several digital samples were collected. With this in mind, a formal study of the app's behavior was conducted using process monitoring. Then both format and structure from any relevant system artifact were analyzed. With the obtained knowledge, a Python script to facilitate extraction and search analysis was developed. This program parses call logs, sms and mms messages, settings, and a list of installed applications and supports facial recognition. The identity verification module uses OpenCV and DeepFace and its objective is to detect one or multiple faces within an image, allow for attributes comparison, and support the search for similarities based on a descriptive profile. A hybrid between OpenCV and DeepFace was chosen for its favorable results and versatility. Finally, the quality of the developed software was evaluated using a large test set.

### 1.2 Motivations

This project was first conceived as the author’s personal research on analysis techniques used in cybersecurity and computer forensics. But, since it quickly grew in length and complexity, a larger goal was set. Why computer forensics? Because, unfortunately, it is a subject that is not taught in the computer science degree, but offers transversal knowledge to many areas such as cybersecurity, software engineering, and data science.

The analysis of Microsft’s Your Phone application can offer very positive benefits to the forensic community. A few are:

* Exposing the formal study of the structure and artifacts by determining what information can be extracted from the artifacts of the application.
* Creating a linked mobile and computer environment through Your Phone.
* Automating extraction of the images stored by the system. Avoid the tedious and repetitive task of searching, selecting, and saving each image in the application.
* Enabling "live" evidence analysis

In summary, studying Microsoft’s Your Phone application from a forensic perspective not only provides the advantages outlined above, but it also offers a unique opportunity to develop innovative research with substantial work.

### 1.3 Work methodology

The first stage, analysis, was dedicated to the study of the program behavior. To acquire as much information as possible about the program behavior, an incremental examination based on the following phases was established:

* Preliminary phase: the working environment and the analysis tools are set up.
* Functionalities exploration: getting acquainted with the app’s capabilities.
* Dynamic analysis: real-time monitoring to identify and collect system artifacts.
* Static analysis: study the main digital traces and artifacts previously discovered.

The second stage, development, focused on the implementation of a software solution dedicated to facilitating access to the previously analyzed information.

* Requirements identification: capturing the needs and constraints of the problem.
* Design: study and establish the best possible software architecture
* Implementation: develop the program for the selected architecture
* Testing: evaluate results and verify compliance with the requirements.

### 1.4 Goals

This chapter sets out the main goal of the work and establishes the various specific objectives that comprise it.

Primary goals:

1. Analyze and report Your Phone’s artifacts identifying any forensically valuable information they might contain.
2. Implement a software solution to collect, parse and export the information presented by the Microsoft Your Phone application

Secondary goals

1. Extract the multimedia content of the application and apply face detection and categorization
2. Implement a facial profile search system. Develop a software tool to locate people given the main characteristics of their faces.
3. Evaluate the correctness and accuracy of the software solution that was developed
4. Evaluate if there exists content safeguarded by the application
5. Extract deleted content from the application

## State of the art

### 2.1 External resources

In the field of computer forensics and cybersecurity, it is common to find publications related to the study and monitoring of free, corporate, and malicious software programs. Such studies usually apply both static and dynamic analysis and usually involve hash analysis, carving, network monitoring, process and thread recognition, Windows registry tracing, and artifact examination. As far as Microsoft Your Phone is concerned, only three publications have been found. These are Digital Forensics Tips&Tricks: "Your Phone" app Forensics [3], Digital forensic artifacts of the Your Phone application in Windows 10 [4] and its subsequent review Microsoft's Your Phone environment from a digital forensic perspective [5]. The first came just after Windows 10’s Insider Preview Build 18999 (20H1) was released and it briefly reviewed the functionalities of the newly introduced application. By contrast, in the second article the authors Patricio Domingues, Miguel Frade, Luis Miguel Andrade and Joao Victor Silva analyze versions 1.0.20453 and 3.4.4.4 of Windows 10's Your Phone and Android’s app Your Phone Companion respectively. In addition, they proposed a Python script designed to run on Autopsy. Finally, the third article examines updates 1.21011.127.0 (Windows) and 1.21021.81.0 (android) and follows the previous line of development, extending some aspects that were left out of the previous study and expanding the functionalities of the proposed program. Compared to the first analysis proposed by Panov the last two publications really expand the knowledge and explain with greater detail how the application was organized and how it stored the user's data. Overall, these three studies manage to meet some of the objectives of this paper, but nevertheless, leave others out.

Some of the technical issues they do not solve are process monitoring, Windows registry analysis, and description of artifact configurations, and that is without adding the challenge of new updates. The application has continued to renew itself, including new features such as the expansion of its gallery, instant messaging, and screen sharing which renders some of the previous work obsolete.

As far as photos and videos are concerned, digital image-processing services embedded in forensic applications are becoming the norm. Companies such as Belkasoft, Magnet, and Cellebrite with their consolidated products such as Platform X, Axiom Cyber, or Physical Analyzer are betting on programs with a variety of utilities, especially image recognition. This trend acknowledges the power of IA by incorporating it into the forensic tool kit to facilitate the work of analysts. Interestingly, none of the previous publications face this challenge, which opens the possibility for innovation.

### 2.2 Legislation

#### 2.2.1 Legislation and legal regulations.

In 2010 Spain ratified The Cybercrime Convention [6], drawn up in Budapest on 23 November 2001. Along with this convention and the Spanish code of law XXX crimes against privacy, espionage, theft, impersonation, fraud, forgery, embezzlement, manipulation of devices, damage, or alteration of data programs or files are contemplated. It is worth mentioning that the forensic profession relies heavily on ethical integrity. There are various codes of ethics and recommendations such as the ISCF code of conduct from the International School of Computer Forensics or the SANS institute work-ethic guidelines. These documents signify the importance of honesty, defense of intellectual property, confidentiality, and individual rights and freedoms In summary professionalism and commitment to safeguarding the truth. Likewise, it condemns any form of corruption (blackmail, bribery), prevaricating attitude, a premeditated attack against privacy, or discrimination based on sex, race, religion, age, ethnicity, politics, or any other condition.

#### 2.2.2 The figure of the computer expert in the courts.

The law defines this role as a professional specialized in computer science whose work consists on providing technical advice in judicial proceedings, as well as contributing to mediation and conflict resolution. He/she can exercise several roles, mediator, arbitrator, and auditor, all of which are overseen by judges.

#### 2.3.3 Chain of custody

When presenting a piece of digital evidence to the judge there are several prerequisites that must be met for admission. For any evidence collected, the original evidence must be preserved together with its chain of custody. The chain of custody is a control procedure that covers the process of obtaining, handling, transferring, assigning, and preserving evidence to rigorously ensure that the evidence has been delivered and remains unaltered (as demonstrated by hash compliance).

#### 2.4 Socio-economic environment

The economic impact is difficult to estimate given that in principle the product is oriented to a narrow niche, forensic investigations involving evidence containing Your Phone artifacts. According to vestigeltd [8], the average costs of a forensic investigation are usually around $5,000 to $15,000 on average. According to the agency, a cost of $250 per hour of effective work can be considered standard. In this sense, the developed project could have a positive impact since it would reduce the analysis time. The software solution developed could also be exported to different contexts such as image extraction functionality. Any application that seeks to interact (input or extract) multimedia content from Your Phone. The same would be true for the face comparator and face grouper, which if generalized and refined could be marketed as an add-on module for some of the forensic programs described above. As for the social and ethical implications, it is worth mentioning that since this is a program designed to discover and work with personal data (conversations, calls, images), strict confidentiality should be maintained

## Implementation

### 3.1 Planning

Following the chosen methodology, the resulting planning is as follows: information gathering, one and a half months, analysis and development, three months, documentation, one and a half months, and testing, fifteen days. The total estimated time dedicated to the execution of the project is 290 hours.

### 3.2 Budget

The costs derived from personnel and work tools:

Human resources: Since the final year project depends on both tutor and student the average salaries of a junior programmer (novice) and a university professor in Spain have to be taken into account. According to glassdoor, the annual average salary is 19.745 € [10] and 33.862 € [11] respectively, i.e. approximately 1.646 € and 2.822 € per month. Working full time (8 hours) 21 days per month is equivalent to 10 and 18 € per hour. If we multiply this value by the number of hours of the participants, the result is 3.100 € and 144 €.

Working tools: the cost of licenses and hardware.

* Hardware tools: to calculate the total hardware cost, the percentage of usage (time used / estimated lifetime) has been calculated and later multiplied by the acquisition price. Tests were carried out on a DELL Latitude E7270 laptop worth 1,179.27 € and a Samsung A20e cell phone 145.20 € with a 16%, 7% of lifetime usage respectively, thus
* Software tools: since the project believes on open-source software, only none licensed products were used.

The project’s budget is 3.244 € + 200€ = 3500 €

### 3.3 Your Phone Analysis

DB Browser and DBeaver were used to explore the artifacts previously found. The study revealed that the database files were based on a specific Pragma configuration. Pragma is a SQL extension specific to the SQLite format that allows customization of databases configuration. To begin with, there was the “Auto Vacum” setting disabled, which indicates that the database does not reduce its size after delete operations. Instead, unused pages of the database file are added to a “free list” and reused for the following insertions. Likewise, the “Secure Delete” was also disabled so delete operations do not overwrite the content with zeros. Finally, there is “Journal Mode” which takes the value Write-Ahead. This explains why for each .db file two additional files get generated .db-wal and .db-shm files. These files are temporary files meant to support internal operations in case of failures, specifically, Write-Ahead Logs are logs/diaries intended for write conflicts on commits and rollbacks, while shared memory files are used when two or more connections share the same .db file and must update the same memory. With the given configuration configured it is possible to recover some records by carving tools such as Undark. However, these methodologies usually yield limited results since often only partial or corrupted information can be found. In the same way, it is not excluded that some records exist in unallocated space.

#### 3.3.1 SQL tables

Regarding the tables and their contents, it was found that:

The address book is stored in contacts.db, where the main table contact resides, which is related through a unique identifier contact\_id with other tables such as phonenumber, postaladress, emailaddresss, contactDate, contactURL. Tests carried out indicates that these are mostly empty, but this could be due to device conditions

The call log is located in call\_history in calling.db. This table provides useful information on phone call number, duration, type (incoming or outgoing), accepted or declined, and date.

Then there is phone.db where the device information is stored. On one hand, in subscription, we have the telemarketer settings, but on the other hand, there is the activity related to conversations sms, and mms messages.

There is also settings.db, which contains information about the device's applications, installed and recent, as well as phone\_request.

Notifications.db stores the applications messages sent to the user. At first, it was conceived as a valuable source of data, but the idea was soon dismissed as the analysis showed that only the phone's notification queue was stored. Your Phone updates the notifications screen every time an application sends a warning. If the user deletes these (either from the phone or PC) the database records are emptied, therefore, one can only access the last state of the notifications queue.

Lastly, photos.db and deviceData.db, two databases containing the image gallery (media, photos) and the device’s wallpaper. As the study by P. Domingues, L. M. Andrade and M. Frade [5], indicated the *photos* table is still obsolete. Instead, the application relies on the table media, which can store up to 2000 image records in the form of blob (byte string). Testing showed that Your Phone does not load the phone's entire gallery at once, instead the program offers images previews as the user scrolls throughout the view. Therefore, when studying a piece of evidence what images are stored will depend on the user’s usage. Regarding image metadata, persistence was examined using an online viewer called Jeffrey Friedl's Image Metadata Viewer [29]. Your Phone does not store the original image in its records but loads a thumbnail as a preview. This low-resolution copy dismisses the original EXIF metadata. In case the user selects a larger resolution is transferred and when a save operation is performed, an identical copy (except timestamps) of the original gets transferred.

#### 3.3.2 Image processing

Image processing is based on two models, OpenCV [30] and DeepFace [20]. The first one is used to detect human faces, while the second one is used to apply face recognition on the obtained faces. The second one is used to apply facial recognition on the obtained faces. OpenCV uses the haar-cascade algorithm, which is not based on deep learning techniques and is, therefore, faster, but its performance is relatively low. In contrast, DeepFace allows to correct this error by adjusting the model (VGG-Face, Facenet, Facenet512, OpenFace, DeepFace, DeepID, ArcFace, Dli) and comparing distance (cosine, euclidean, euclidean\_l2) so it was more beneficial to use it as a comparator. However, unlike DeepFace, OpenCV can detect multiple faces within an image. By combining both alternatives, it is possible to identify, isolate, crop, and evaluate each face present in Your Phone. Since the two models work together, it may happen that OpenCV first detects and cuts out a face, and then DeepFace does not recognize any face. In these cases, cropping is discarded since an image without a recognizable face is an image on which no comparisons can be made.

Regarding face recognition, the decision was made to offer three services to the analyst:

* **Suspect clustering:** group similar faces from the set of extracted images
* **Face comparer search engine.** Given a suspect’s face image, find similar faces within the set of extracted images.
* **Facial profile search engine.** Find faces that fit a certain description based on age, gender, gesture, and race of a given subject.

#### 3.3.2 Execution modes

The first step towards the implementation of yourPhoneForensicAnalizer was to develop the execution control. At a minimum, the script requires the path to the directory containing the databases (--i or --input). Optionally the user can indicate an output folder (-o or --output). The program continues to evaluate the execution modes provided by the analyst:

* Modes without input parameters.
  + Help mode: -h or -help
  + Verbose mode: -v or -verbose
  + Image export mode: -e or -export
  + Group similar faces mode: -gfi or -groupFaceImages
* Modes with input parameters.
  + Search by faces mode: -sfi -searchFaceImages pathToImages
  + Search mode using face profiles: -sfp -searchFaceProfiles pathToCSV
  + Phone search mode: -spn -searchPhoneNumbers pathToCSV

Once the execution parameters have been validated, the database files are scanned in search of the user's information. If one or more search modes have been set, the search set will be limited to the provided criteria. No amplitude crawl will be performed, as the search criteria will be imposed on the database instances that meet the provided parameters.

### 3.4 Algorithm design

The algorithm retrieves a contact identifier from each contact present in the address book. If the selected identifier is associated with a phone number every call or conversation related to it gets retrieved. Nonetheless, not every message or phone call is related to the address book. These are processed and output later as "Unnasociated phones, calls, sms, and mms".

Once the parsing is finished, the algorithm tackles images present in photos.db and deviceData.db by evaluating if the database rows contain any data stored in thumbnail and media fields. Then face detection is performed using OpenCV’s VGG-Face model. If the prediction is greater than 0.5 the module crops a rectangle around the analyzed pixels. Next DeepFace analyzes the cropped image using convolutional neural networks. These models use standard size inputs, so the library normalizes and aligns them before sending them to the model. Then they are propagated through the model, and either the library produces an exception when no face is detected, or returns a Python dictionary with the different categories identifying the face. These results are used for face clustering and image searching.

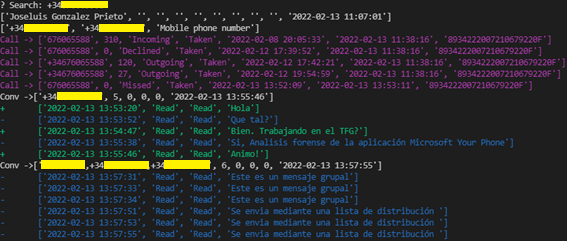
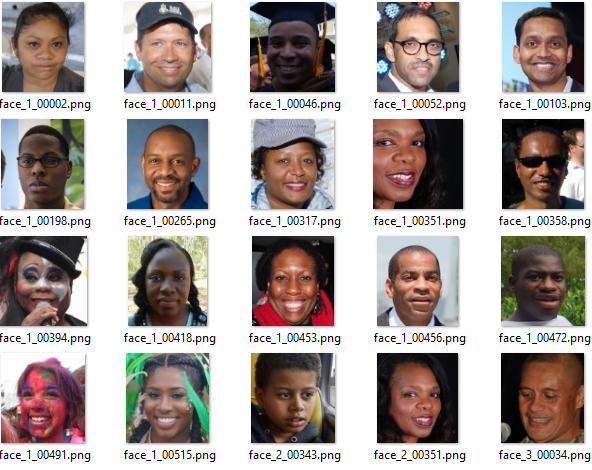
 

Fig. 15 Associated phones, calls, sms, and mms". Fig. 13 Black faces

## Conclusions

### 4.1 Accomplished objectives

This project has fulfilled all of the goals proposed in section. Some of the challenges have been:

1. The image’s resolution in Your Phone. No image stored in the databases exceeds 1.5 MiB, which does not facilitate detection since the lower the resolution, the lesser the information available.
2. The number of recognizable faces is lower than the number of detected faces. Ideally, it should be possible to extract identifying characteristics of every detected face; however, since these tasks vary in complexity it is not always achieved.
3. The face clustering functionality produces ambiguous results. The proposed algorithm for this functionality is weak, as it does not produce reliable results. The task of clustering faces turns out to be much more complex than initially expected, so the selection and discard algorithm is not sufficient.

### 4.2 Future lines of work

Due to the limited access to resources and the time constraints, there are some aspects of the project left uncovered. Future development will likely have to work on face detection optimization as well as expand the forensic study.

On one hand, the face recognition module should improve the current accuracy in face recognition by extending the experimentation to all models with all distance measures supported by DeepFace. Moreover, a reliable alternative to the clustering algorithm must be found.

On the other hand, the study of Your Phone can grow even larger with the inclusion of the screen sharing service, only supported for the latest mobile models, and Android device artifacts. Similarly, just like this research has done, future updates of Your Phone will require similar analysis techniques since it is very likely that Microsoft will expand the synchronization of cell phones and PCs. Lastly, it would be neat to add new parsing modules to the current program so that extraction continues to be easy and accessible.

To conclude, one last improvement proposal. Given that criminal investigations often deal with images of child pornography, homicides, or domestic violence it would be helpful to introduce censorship systems for sensitive material. Software platforms apply blurring filters on this type of content, so it would be professional to add similar means of censorship to prevent overexposure of the analyst. Another aspect that could be expanded is image processing. Image recognition need not be limited exclusively to face detection and classification, but could also be extended to object detection.

### 4.3 Conclusion

The project covers multiple topics ranging from cybersecurity, software design, to artificial intelligence. The purpose was not to focus exclusively on one of the exposed areas but to offer a multidisciplinary comprehensive study of the defining characteristics of Your Phone. In addition, software solutions with multiple functionalities are presented as well as tested. In conclusion, it should be noted that the field of computer forensics is a vast, interdisciplinary, and constantly advancing field where new technologies emerge every day and those which already exist change over time. Hence, the challenges faced by analysts also change, so it is crucial to keep the discussion and research open by the community. As a result, this work will be available for free as an open-source resource at https://github.com/groongra/Microsoft-Your-Phone-parser.

Una vez hecho se realzó el siguiente estudio:

|  |  |  |  |
| --- | --- | --- | --- |
| TABLA 3. - Procmon - Procdot | | | |
| n | Descripción | Conclusión | Hallazgos |
| 1 | Vincular dispositivo | Emparejar movil y pc | REGISTRYKEY:\registry\a\{5472abfe-5b7e-89ff-d098-98132f6eed9d}\localstate\yppappprovidercontainer\0857d319-bd80-495f-a54a-34c472f351aa  \REGISTRY\A\{5472abfe-5b7e-89ff-d098-98132f6eed9d}\LocalState\ExpOverrides\ExpFileOverrideEnabled |
| 2 | Desvincular dispositivo | Desmparejar movil y pc | Hay registros LastSelectedView |
| 3 | Llamada telefónica desde el PC |  |  |
| 4 | Guardar imagen de la galería |  |  |
|  |  | Permite mostrar la última fecha en que se actualizó | \REGISTRY\A\{5472abfe-5b7e-89ff-d098-98132f6eed9d}\LocalState\Devices\0857d319-bd80-495f-a54a-34c472f351aa\PhoneAppsLastUpdatedTime |
|  |  | Muestra al usuario el nivel de bateria | \REGISTRY\A\{5472abfe-5b7e-89ff-d098-98132f6eed9d}\LocalState\YppTelemetry\LinkedDeviceBatteryInformation |