MOBILE FORENSICS APPLICATION

Research Manual

Student: Connor Scanlan – C00226867

Supervisor: James Egan

Cybercrime & IT Security – CW_KCCYB_B

Institute of Technology Carlow

Table of Contents

Abstract

The Increase in Mobile phone usage in our society grows daily, with ever advancing technologies related to mobile phones. These advancements have led to greater computing power, increased functionality, and longer battery life along with the size of the devices remaining small and easily portable. This has also led to mobile phones becoming portable data carriers, which in turn has increased the potential storage a device can have stored. In order to develop a single application which would allow the user to search a device and copy all of its data without damaging or losing any. During the process of data transfer, the application will allow the user to search for Keywords. If keywords are identified the application will make a second copy and transfer it into a separate folder for further investigation. There will be options to grab the Geodata from every image, or from selected images. Within the folder the application will place the image beside a text file which holds its Geodata for use in the future. There will be an option for the application to attempt to open any locked files using a Brute force technique on the copied files, keeping the original files intact for any further investigation that may be required.

The aim of this application is to be very user friendly and visually clear on its instructions.

Introduction

As of 2020 there are 14.02 Billion Mobile devices with this expected to increase to 14.91 Billion by 2021. In 2024, it is projected that the number of active mobile devices could reach 17.72 Billion. These numbers are huge when compared to the 1.5 billion mobile devices globally in 2005. In comparison to this there are only around 2 billion computers around the world. With these statistics Mobile devices outnumber computers 7 to 1, yet mobile forensics is still a bit behind computer forensics.

Research

In researching for this project, I found portions of interest in various areas that could be used for the application in which I envisioned. The challenge was compiling these in a compatible format which would create the desired result.

In using the following tools AnalogExif for Apple and Windows, JPEG and PNG Stripper for Windows only and ExifTool for Windows, Mac OS and Linux (is command line based), Meta data can be gathered. The challenge at present is that each application does not work across all devices seamlessly. The complexity of each tool is a determining factor in their usage. They each require you to do each imagine manually one at a time.

Examples of applications that can copy the contents of a device include XRY logical and XRY physical which allows for data collection, and has the capacity to find deleted data, ACESO which allows a user to collect data without leaving a trace or being noticed. The aim of this new application is to multitask by copying all of the devices data onto an external storage but also set aside any Keywords that's the user searched for simultaneously.

The use of the tools listed above can vary between devices, this new application is intended to determine which process to use itself upon identifying the target device.

The need for better Mobile Forensics tools

Use of mobile phones to store and transmit sensitive information

Mobile applications are being rapidly developed with thousands being released for download daily, these include: Social Media, Games, Educations and Tools for work. Many of these applications can help with daily tasks such as a better way to communicate, store data or even pass the time. The increase of connectivity through the development of 4G and 5G means that even more data will be passing through all these mobile devices. With this increased traffic comes the question of security, who can access your files and how are they doing it. For Law enforcement being able to find all the data on a suspect's device is getting increasingly more difficult as not all the data is stored on the device itself as Mobile phones have access to cloud storage.

Online transactions

Wireless Application Protocol (WAP) has enabled mobile devices to use online transactions which led to the development of digital wallets, these added convenience to online transactions which increased the amount of money passing through mobile devices. With companies such as Amazon being huge online stores and allowing a customer to order a product from the comfort of their own home with relative ease on their mobile phone, it has attracted many criminals to try intercept these transactions for their own personal gain. Other online transactions include mobile banking, stock trading, booking flights and making hotel reservations. This increased attraction means the mobile forensics industry as a whole needs to be constantly improving to keep up with the demand for security.

Law Enforcement and Criminals

The gap between law enforcement and criminals has always been considerable when it comes to the usage of mobile devices. In the 1980’s, criminals were very quick to release that mobile phones and pagers could be used as tools to evade being caught and even increase their profitability by keeping constant communication, it also allowed many of them to create wide reaching criminal empires as they could communicate further and faster. On the other hand law enforcement was severely lacking as they had never had to deal with this kind of criminal before, they were also at a disadvantage as they needed specialized tools which at the time were considerably more expensive. Even now days criminals are always looking for more ways to use mobile devices to their advantage with the use of mobile shopping and banking it has made a great opportunity to steal much more money in one go with a lot less effort and manpower needed. While law enforcement has improved significantly in the mobile forensics sector they are still lagging behind as they are restricted by funding and manpower.

Mobile Data as Evidence

Definition of Digital Evidence

Digital Evidence has been defined as "***information and data of value to an investigation that is stored on, received or transmitted by an electronic device***": *Electronic CSI, A Guide for First Responders, 2nd edition, National Institute of Justice, April 2008*

By this definition digital evidence is not only found on Computers and Mobile phones but also any device that operates on the Internet of Things (IOT), these other devices can include: Gaming Consoles, Smart Fridges, Smart TVs, Smart Watches and any other device that have the ability to connect to other electronic devices.

Principles of Electronic Evidence

The United Kingdom's Association of Chief Police Offices (ACPO) Good Practice Guide for Computer based Electronic Evidence (2003) says there are four principles involved when using digital evidence:

- Principle 1: No action taken by law enforcement agencies or their agents should change data held on a
computer or storage media which may subsequently be relied upon in court.
- Principle 2: In exceptional circumstances, where a person finds it necessary to access original data held
on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.
- Principle 3: An audit trail or other record of all processes applied to computer based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result.
- Principle 4: The person in charge of the investigation (the case officer) has overall responsibility for ensuring that the law and these principles are adhered to.

In regard to the recovery of digital Evidence, "The Guidelines for Best Practice in the Forensic Examination of Digital Technology" published by the International Organization on Computer Evidence

The following are considered:

- Guideline 1: The general rules of evidence should be applied to all digital evidence.

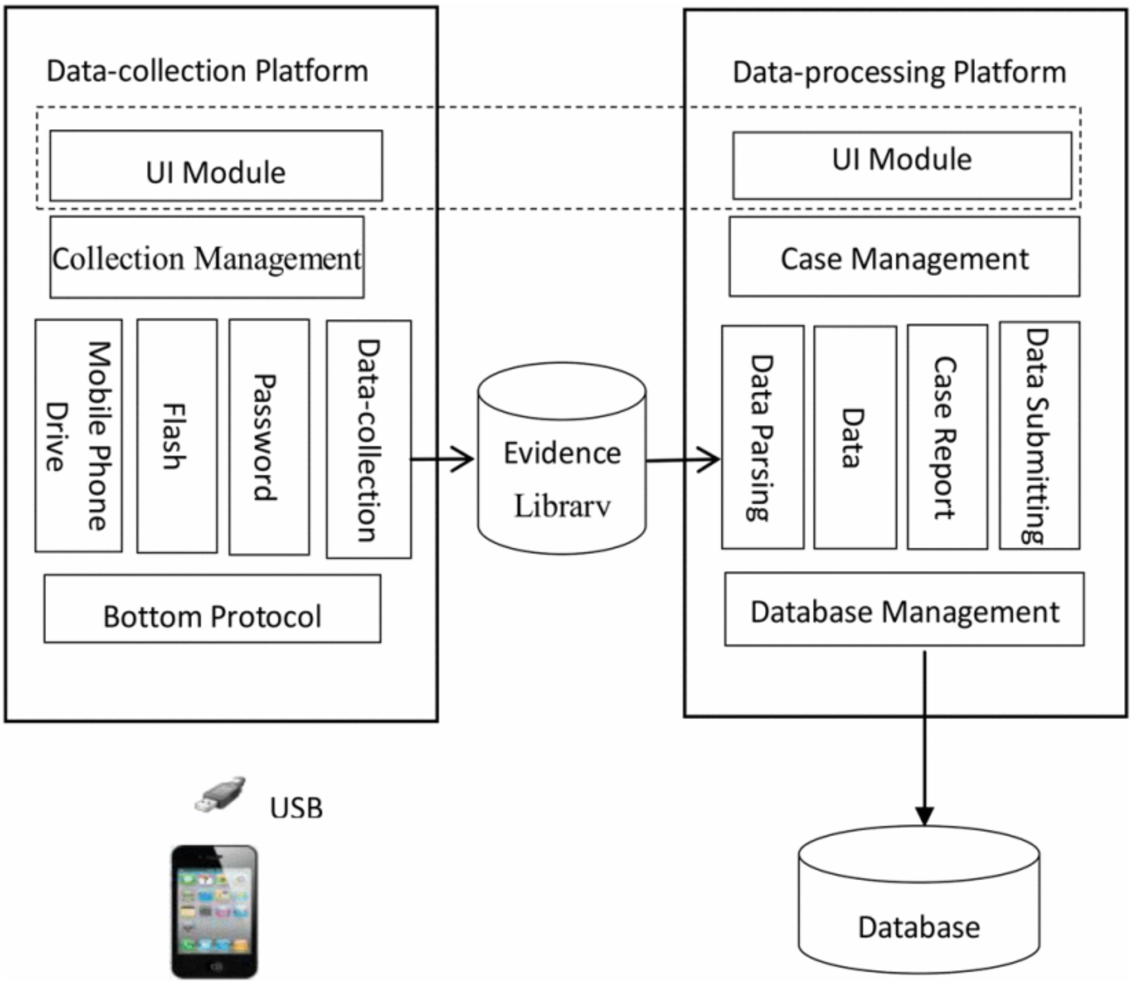
- Guideline 2: Upon seizing digital evidence, actions taken should not change that evidence.

- Guideline 3: When it is necessary for a person to access original digital evidence that person should be suitably trained for the purpose.
- Guideline 4: All activity relating to the seizure, access, storage or transfer of digital evidence must be fully documented, preserved and available for review.
- Guideline 5: An individual is responsible for all actions taken with respect to digital evidence whilst the digital evidence is in their possession.

Due to the dynamic nature of a mobile device's memory ACPO's principles and Guideline 2 cannot be strictly followed. The aim would be to follow them as best as possible, but in terms of data such as constantly updated data that is done automatically by the system sometimes cannot be stopped.

Framework of Mobile Forensics

Mobile Forensics System Structure



The aim of a mobile forensics' application is to collect all/relevant data from a mobile device, store it on a Database and analysis it. (as seen in the Figure above) Many mobile forensic tools can begin working simply by connecting to the device via a USB cable. Different brands and types of mobile devices can have different ways to store their data so different plug-ins can be called to each data-collection sublayer to assist in collecting user specific data. Some tools may need to take control of the power of a device in order to gain access to the storage.

Mobile Forensics Processes

The collection of mobile data can be divided into three steps: collection of data, parsing of data and analyzation of data. For the application I am developing the process of collection will be done via USB this will connect the Mobile device to the System in which the application is downloaded. The application gathers the data by physical and/or logical protocols. The Parsing of the data involves saving the gathered data onto a database and into its relevant folders with in, this would include basic information and applications information divided into their own folders to allow them to be easily viewed. The Analyzation of the data will get additional and hopefully valuable data from the data gathered. Some errors may occur during the process, these could include communication may be interrupted, or same data may not be able to save onto the database. In order to prevent the application from crashing should these errors occur, regulations for dealing with the errors and a log of errors will be saved to analyse the reasons after.

Mobile Data Collection

There are two kinds of Data collection, one kind is to collect data by logical protocols or by a physical mirror protocol, the other kind is to use a card reader. Both of these ways have their own benefits and disadvantages, the first is much easier to do and can be done quickly by simply connecting the device via a USB connection. The second way is much more complex as it requires a trained user to do, but it can analyse the bottom information directly on the phone.

The different types of files a mobile device can generate include xml, journal, SQLite, wal, png, jpg, plist and many more. All of these files form the source of evidence.

The two most popular mobile operating systems on the market are IOS and Android, there are some smaller brands with their own operating system such as Windows mobile, Blackberry and so on. My application will use the first kind of data collection explained above. The IOS system supports Backup official protocol and AFC protocol from open source. BackUp2 supports data backup and data collection for Apple jail broken mobiles and AFC2 supports data collection and is very useful for extracting data from a specific directory at speed. IOS7 introduced Trust as a protocol which requests that any computer that wants access to the data must get trust from an Apple Phone. MTP is a protocol that can be used to extract data from Android.

Encryption and Decryption

Decomplication and AES

Decomplication is a direct method, it is used to decompile software to make an executable procedure in assembly language, it can then find method encrypted SQLite and any passwords by analysing the assembly language. This method will become much easier to do when any procedure executables do not add shell. For now, it must take full advantage of unshelling. Some mobile phone APP's only encrypt certain fields instead of encrypting the entire SQLite. Some APP's only change the field type for encryption, which makes them easier to get access to if you notice it. AES, DES and IDEA are examples of common symmetrical algorithms whilst RSA is an example of an unsymmetrical algorithm for encryption. Due to the limitations of hardware on mobile devices the encryption needed to be faster, stronger but also low in its complexity. That is why SQLite databases choose to use AES for its encryption.

The SQLite database has four main characteristics regarding encryption:

- Realize the function of encryption interface remained in SQLite server-side.
- Manufacturers of some APP's directly encrypt entire SQLite by themselves.
- Manufacturers of some APP's directly encrypt parts of the field themselves.
- Key used in encryption usually derives from IMEI of mobile phone, account of user or other special string.

Following these characteristics of encryption in SQLite, the following decryption method is proposed:

- Firstly, decompile the APP, reckon encryption algorithm and key.
- If the first step fails, then use tools of analyzation like WinHex to compare cipher and plaintext for discovering the encoding regulation.
- If the second step fails, use the algorithm of AES decryption to try again and again.

Application Specifics

Which Type of Application?

Having done a general research into Mobile Forensics, it was necessary to identify for elimination several areas of evaluation. The first area to focus on was whether the application should be Desktop or Web based. The pros and cons identified are listed below:

Desktop vs Web Application

Desktop Application

Pros

- Better Security
- No Question of Ownership
- Don't need an Internet connection
- No Questions of legality
- Rely on Computer Speed
- Cheaper in Long-term
- Can run older versions easily
- No 3rd Party Support

Cons

- Rely on Computer speed
- Users must Download Updates
- Only on the device it is downloaded

Web Application

Pros

- Rely on Internet Speed
- Works on all devices
- Cheaper in short term
- Easier for multiple users
- Less OS requirements

Cons

- Needs an Internet connection
- Questions of Legality
- Less Secure
- Rely on internet speed

- More expensive in the long run
- Difficult to run older versions
- 3rd party support

The findings of this research resulted suggested that a Desktop Application would better suit the project because it fulfils the requirements to produce a security-based application which should be more secure for the user. The Desktop application would be more reliable as it would not be influenced by fluctuation in Internet connection, which can vary considerable depending on infrastructure.

Although one of the serious pros for the Web application was the ability for multiple users to work in conjunction with each other, it did not outweigh the security benefits of a Desktop application.

With the Desktop application the user has much more control over which version of the application they use to best serve their needs and equipment, which influences the cost effectiveness of such an application within their workspace.

What Language?

Whilst Python is more built towards creating Web applications, there are numerus Toolkits and Libraires available for creating a Desktop application. Python is an open source language with great flexibility, but it requires external GUI tools to create Desktop apps. The following GUI tools were researched in order to identify their suitability for developing this project. The GUI tools are included below:

Python GUI tools for creating a desktop application

PyGTK

- Free software
- GTK Tool Kit and Libraries
- Works on Windows, Linux and OS X

QT

- For both GUI and non-GUI applications
- Use PyQT and PySide
- Supports Python 2 and Python 3

TKInter

- Most Popular

- Already a part of Python

WxPython

- Open source
- Works for Windows, Linux and OS X
- Used to create High Performing desktop apps

Other GUI tools that were researched include: Kivy, Camelot, Pyjamas Desktop, CEF Python, Cocoa. A lot of the elimination process included the ability to comprehend and understand their instructions and ease of usage.

The four GUI Tools in the shortlist at present include: PyGTK, QT, TKInter and WxPython. As this project further develops it will be necessary to reduce this list down to one GUI tool. At this time research is ongoing and such a decision needs further investigation.

What IDE?

Choosing an IDE for this project that best suited the requirements I set out, was very important. As a result, the research was compiled on the pros and cons of the following IDE's:

Python IDE's

General Editors and Ides with Python Support

- Eclipse + PyDev
- Sublime Text
- Atom
- GNU Emacs
- Visual Studio

Python Specific Editors and IDE

- PyCharm
- Thonny

Eclipse + PyDev

Pros:

- Eclipse is an IDE that I am already familiar with

- Adding PyDev will speed up the setup process
- A lot of informative support available

Cons:

- Steep learning curve using a new language
- A lot of unnecessary tools

Sublime Text

Pros:

- Huge community following
- Considered a fast code editor
- Small in size

Cons:

- Costs money
- No debugging within the editor

Atom

Pros:

- Works on all platforms
- Small in size

Cons:

- Does not run as a native application
- Add-ons required for debugging

GNU Emacs

Pros:

- College support available
- Current subject of learning
- Efficient
- Customizable

Cons:

- New subject

- Steep learning curve
- Customization can cause huge variation when looking for support

Visual Studio

Pros:

- Familiar with Visual Studio
- Quick addition of PTVS

Cons:

- No Linux installation

PyCharm

Pros:

- Designed for Python
- Support available
- Works out of the box

Cons:

- Can be slow

Thonny

Pros: Great for beginners

- Works out of the box

Cons:

- May be to basic
- New tool
- Limited support

The research has allowed me to narrow the IDE selection to GNU Emacs and Eclipse as a result of this process. My familiarity with Eclipse was a big selling point to choosing this

IDE and having a lecturer who is passionate and experienced with GNU Emacs means that I have access to support. The selection process in finalizing the chosen IDE will happen shortly.

External Programs for Integration

Volailty

This project will benefit from the integration of Volailty as it is one of the best open source programs for the analysis of RAM in both 32 bit and 64 bit systems. Volailty supports usage on Windows, Linux, Mac, and Android systems. It was designed to be used with Python. Its purpose is to analyse raw, crash, VMware and virtual box dumps.

GeoData

One of the features of this application will be grabbing GeoData and using it to find where images were taken. This will be done by using GPSphoto to collect metadata from jpegs. After which the longitude and latitude will be identified by using Selenium. Time will be used to slow down Selenium processes so that the data can be grabbed. Following which OS will create a folder, and Shutil will then move the files and their location into the newly created folder.

Brute Force

To open any encrypted folders/files this project intends to use gpgBruteForce program until such a time when a cracking tool can be developed alongside the project.

The reason that gpgBruteForce was chosen is its simplicity and its ability to attempt approximately 100 passwords per second.

Conclusion

While there are hundreds of applications available that will do individual parts of this application, there is very few that can do it all as one application and the few that do exist are incredibly complicated to use. My aim for this project is to create a easy to use and simple to learn application that will help users get the data they need quickly and have it ready to present. I also hope to be able to add more features to my application at a later stage and have it become a fully functioning multitask mobile forensics tool.

Bibliography

Real Python (2018). *Python IDEs and Code Editors (Guide)*. [online] Realpython.com. Available at: https://realpython.com/python-ides-code-editors-guide/.

Kohli, A., Reierson, K.H. and Anderson, A.M., Microsoft Technology Licensing LLC, 2015. *Converting desktop applications to web applications*. U.S. Patent 9,176,742.

Grinberg, M., 2018. *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

Yamacli, S., 2018. Beginner's Guide to Python Programming: Learn Python 3 Fundamentals, Plotting and Tkinter GUI Development Easily.

Ruohonen, J., 2018, December. An empirical analysis of vulnerabilities in Python packages for web applications. In *2018 9th International Workshop on Empirical Software Engineering in Practice (IWESEP)* (pp. 25-30). IEEE.

Choudhary, S., 2019. pysradb: A Python package to query next-generation sequencing metadata and data from NCBI Sequence Read Archive. *F1000Research, 8*.

Bonzanini, M., 2016. *Mastering social media mining with Python*. Packt Publishing Ltd.

Rajabifard, A., KALANTARI SOLTANIEH, S. and Binns, A., 2009. SDI and metadata entry and updating tools.

Huang, Z., Ayday, E., Fellay, J., Hubaux, J.P. and Juels, A., 2015, May. GenoGuard: Protecting genomic data against brute-force attacks. In *2015 IEEE Symposium on Security and Privacy* (pp. 447-462). IEEE.

Bray, S.W., 2020. *Implementing Cryptography Using Python*. John Wiley & Sons.

Goyal, P., 2020. Brute Force Attacks. *CYBERNOMICS*, *2*(7), pp.17-20.

Sweigart, A., 2013. *Hacking Secret Ciphers with Python*. CreateSpace.

Annamaa, A., 2015, November. Introducing Thonny, a Python IDE for learning programming. In *Proceedings of the 15th Koli Calling Conference on Computing Education Research* (pp. 117-121).

Islam, Q.N., 2015. *Mastering PyCharm*. Packt Publishing Ltd.

Stallman, R.M., 2007. *Gnu Emacs Manual: For Version 22*. Free Software Foundation.

Cameron, D., Rosenblatt, B., Raymond, E. and Raymond, E.S., 1996. *Learning GNU Emacs*. " O'Reilly Media, Inc.".

Haughee, E., 2013. *Instant Sublime Text Starter*. Packt Publishing.

Burnette, E., 2005. *Eclipse IDE Pocket Guide: Using the Full-Featured IDE*. " O'Reilly Media, Inc.".

Vogel, L. and IDE, E.J., 2013. Eclipse IDE tutorial. *Vogella. com*.

Finlay, J., 2005. PyGTK 2.0 Tutorial. *ozone-friendly. ru, Published April, 13*.

Grayson, J.E., 2000. *Python and Tkinter programming*. Manning Publications Co. Greenwich.

Shipman, J.W., 2013. Tkinter 8.4 reference: a GUI for Python. *New Mexico Tech Computer Center*.

Moore, A.D., 2018. *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. Packt Publishing Ltd.

Summerfield, M., 2007. *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming (paperback)*. Pearson Education.

Rappin, N. and Dunn, R., 2006. wxPython in Action.

Curran, K., Robinson, A., Peacocke, S. and Cassidy, S., 2012. Mobile phone forensic analysis. In *Crime Prevention Technologies and Applications for Advancing Criminal Investigation* (pp. 250-262). IGI Global.

Al-Zarouni, M., 2006. Mobile handset forensic evidence: a challenge for law enforcement.

Lee, X., Yang, C., Chen, S. and Wu, J., 2009, August. Design and implementation of forensic system in Android smart phone. In *The 5th Joint Workshop on Information Security*.

Watson, S. and Dehghantanha, A., 2016. Digital forensics: the missing piece of the Internet of Things promise. *Computer Fraud & Security*, *2016*(6), pp.5-8.

Su, Q. and Xi, B., 2017, March. Key technologies for mobile phone forensics and application. In *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)* (pp. 335-340). IEEE.