12-2019

Comparison of Forensic Acquisition and Analysis on an iPhone over an Android Mobile Through multiple forensic methods

Vikas Avancha
vavancha@stcloudstate.edu

Comparison of Forensic Acquisition and Analysis on an iPhone over an Android Mobile Through Multiple Forensic Methods

by

Vikas Avancha

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Information Assurance

December, 2019

Starred Paper Committee:
Mark Schmidt, Chairperson
Lynn Collen
Sneh Kalia

Abstract

Mobile phones are most widely used as mini laptops as well as personal digital devices one could have. The dependency on mobiles for every single person on every single aspect has increased day by day. Depending on the operating systems, storage capacity, user interface developed by various manufacturers, there are numerous mobile phones designed with diverse computing capabilities. Among all the distinct kinds of smart mobile devices that are available in the mobile market, iPhone became one of the most popularly used smart mobiles across the world due to its complex logical computing capabilities, striking touch interface, optimum screen resolutions. People started relying on iPhone by utilizing its functionalities including storing sensitive information, capturing pictures, making online payments by providing credentials. These factors made iPhone to be one of the best resources for the forensic department to retrieve and analyze sensitive information and provide supporting evidence. Thus, the rise of iPhone forensics took place where the data is retrieved and analyzed with the help of various iPhone forensic tool kits. The agenda of this paper is to give overview of iPhone forensics and mainly focuses on analysis done, and challenges faced while retrieving the sensitive information on iPhone by means of distinct forensic tools when compare to Android mobile device forensics.

Acknowledgments

I would like to thank Professor Dr. Mark B. Schmidt for supporting me to work on the paper, comparison of acquisition analysis of iPhone over android mobile forensics. Without his expert advice and encouragement, it would have been difficult for me to work on this research paper. I also thank Professor Sneha Kalia and Professor Lynn Collen for accepting to be my committee members and contributing their valuable time in correcting my paper and offering their individual suggestions.

Table of Contents

Chapter Page

List of Tables

List of Figures

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

List of Forms

Chapter I: Introduction

Introduction

Mobile phones took the space as one of important possessions a person would have in his life, transforming every little piece of information from simple to smart way. The enhancement of innovative technology is growing day by day so as mobile technologies. These mobile phones are being advanced from just picking up the calls to attending the conference through video calls from remote areas. At present smartphones became one of the traditional desktops with same functionality but different in terms of operation.

Further improvements in telecom technologies has given an immense change over through a period, where a smart mobile would serve all the possible works such as web surfing, finding location with the assistance of GPS system, scheduling meetings through emails, taking pictures, utilizing various applications depending on the operating systems, paying bills and lot more (Laboratory, 2015). Different telecom industries have launched smartphones with different user interfaces, processors, camera resolutions, logical structures that drags attention of the people by accompanying them with various needs.

Along with all these facilities, a smartphone appears with huge gigabytes of memory allocations that are assigned for processing various functionalities. In addition, it holds enormous gigabytes of storage capacities where it can store, download and upload the information directly and save the data at its internal location. Due to rapid usage of these smartphones in regular activities, making these devices to capture and store sensitive information about the person such as contact logs, call lists, recordings, login credentials for online payments, bank transactions etc. which may lead to misusage if it’s been in wrong hands.

This sensitive information available in a mobile phone can become the best evidence for a convicted crime using digital media as one of the resources (Mona Bader, 2010). This leads to the rise of mobile forensics which is a part of Digital forensics of acquiring and analyzing data retrieved from a digital device such a computer, laptops, mobiles to utilize the information as evidence for criminal activities which aids in solving a criminal case and it consists of four areas such as collecting the data, examining the data, analyzing the data and reporting the data.

Among all the distinct kinds of smart mobile devices that are available in the mobile market, iPhone became one of the most popularly used smart mobiles across the world due to its complex logical computing capabilities, striking touch interface, optimum screen resolutions, huge storage capacities. Thereby analyzing and retrieving the data from an iPhone will be a great resource and beneficial to the forensic investigators.

Problem Statement

The key Agenda of this paper is to indicate different methods and tools applied on an iPhone and android mobile to obtain forensic evidence by analyzing the retrieved data. In this research, to apply forensics on an iPhone or iOS device when compared to android devices, a forensic investigator needs to understand the details about device logical operations internally.

In addition to that, the forensic investigator needs to figure the access points for the device to retrieve the data using forensic tools. Some of the challenges that needed to overcome while acquiring the information are as follows:

- Unlike digital forensics, iPhone forensics requires more effort and time for gathering the information since the data in the iPhone can be accessed from multiple devices as well as modified from any other iPhone if it is initiated with same backup process.

- Enhanced security features including encryption techniques available in an iPhone increase difficulty in obtaining sensitive information.
- Unlike digital forensics, write blocking techniques for iPhone forensics are unavailable that makes forensic investigators more complicated to retrieve the data.

Nature and Significance of the Problem

Forensics techniques used for the digital forensics on computers are not similar when working with iPhone forensics, since there are several challenges that fall in when working on iPhone forensics such as the device is designed with solid-state flash memory and there is no inbuilt slot to insert external memory card, it can be accessed from multiple devices and thus the sensitive data can be modified from remote locations.

This paper illustrates some of the techniques and computations used for iPhone forensics to overcome all the challenges faced when performing forensics on iPhone to retrieve data.

Limitation of Study

The objective of this paper is to analyze and compare the different methods used to retrieve sensitive data from an iPhone when compared to android device. This study does not make any attempt to propose new concept or modifications in the existing methods to extract the data but suggests how these existing methods individually are not sufficient to acquire results and examine the different techniques applied to gather forensic information by comparing with Android mobile device forensics.

Chapter II: Background and Review of Literature

Introduction

Mobiles forensics is a category of digital forensics identified with the retrieval of evidence which are forensically acquired from mobile devices. The fundamental guideline for a forensic analysis of the digital evidence is that the original information in the mobile devices must not be altered. This is a challenging task to perform on mobile devices. Depending on the type of extraction, some of the forensic tools need to analyze the data by removing a chip or placing a boot loader on the mobile device before retrieving the data.

For some cases where there is a possibility of changing the configuration of the device where the procedures must be verified, approved and reported to follow the proper approach and rules to analyze the forensic information from these devices. Failure in reporting the data without following proper procedure may lead to violation of forensics guidelines and result in damage of evidence gathered or disapproval of information in the court. The mobile forensics methods are basically categorized into three main categories i.e.

- Seizure
- Acquisition
- Analysis

While collecting the mobile devices at the crime scene, there are many challenges that need to be encountered for a forensic investigator. If the device is switched off at the location, it should be covered in a faraday bag to stop the alterations if the device gets automatically switched on.

Faraday bags are mainly designed in a way to prevent any network incoming or outgoing from the mobile and by this the information in the mobile cannot be altered from remote locations using the network as a medium.

The below figure depicts the typical EDEC digital forensic faraday bag.

Figure 1: EDEC digital forensic faraday bag (Adorama, 2018; Sutton, 2011).

If the mobile device is already inactive state, there is a possibility that the information from the phone can be erased by the criminal from a remote place using different wipe commands. To acquire unaltered data from the mobile, the forensic investigator should turn off all the mobile sources such as WIFI access, network, Bluetooth and turn on flight mode before placing the device in the faraday bag.

Confirming that the mobile is collected in the right way by placing in faraday bag, the forensic investigator may need different forensic tools to examine the data present on the mobile. Depending on the distinct kinds of mobiles manufactured from diverse brands, their functionalities and user interface may vary. These versatile features require various kinds of tools

to perform forensics on the device and gather the sensitive information that can be represented in the court as the forensic evidence (Sutton, 2011).

Forensic Tool Leveling System

There are five different classifications for a mobile forensic tool to be categorized based on the examination methods used by the tool. Any Forensic tool for extracting the data from the mobile device will have pros and cons and it is necessary to know for a forensic investigator that no tool is perfect with having only pros. With the help of these classifications, a forensic investigator can understand and analyze these tools with proper forensic acquisition.

The five classifications are:

- Manual Extraction
- Logical Extraction
- Hexadecimal Dump/JTAG
- Chip Off
- Micro Read

The below figure illustrates the classifications of mobile tool leveling system.

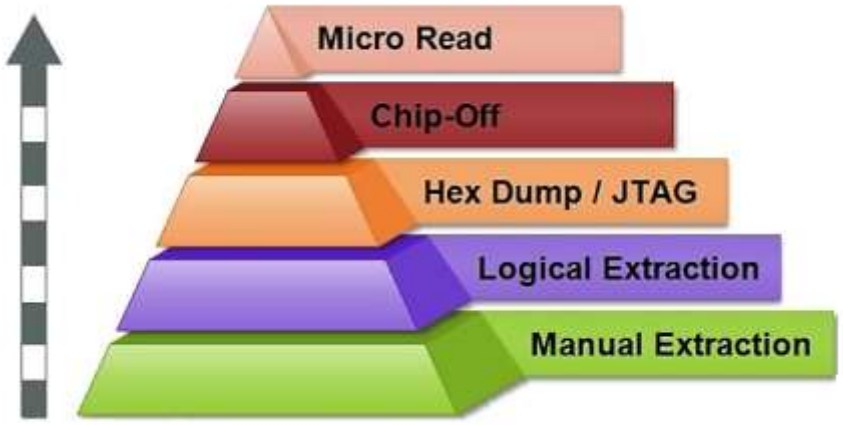


Figure 2: Forensic tool leveling system (Dimitar, 2018).

Without the utilization of these methodologies on a forensic tool before implementing, examiner may destroy or lose evidence while retrieving from the mobile phone. The level of difficulty inclines gradually when you start from the bottom. To overcome this risk, forensic examiner should have proper training before working on the mobile device to achieve optimum results in extraction.

These five levels in mobile forensic level tooling system are discussed below:

Manual extraction. Manual extraction is the simplest method that involves gathering sensitive data by simply using the phone directly with touch interface or with the help of keypad. The information gathered from the mobile device is documented through screenshots and with the help of capturing photos. This process is the easiest and works for almost any mobile. But these methods of extraction have lot of concerns such as making errors in finding the sensitive data due to lack of proper knowledge on interface. Another concern is that viewing the unread text messages on mobile devices leads to modification of data (i.e., SMS notification varies from unread to read messages).

Logical extraction. In this method, the mobile device is connected to a forensic medium through a USB cable, Infrared, or Bluetooth. After connecting both the device, the data is transferred to the forensic tool after all the required command process is accomplished. Deleted files on mobile devices can be retrieved at this level. This level of extraction has the probability of altering the information in the mobile device leading to integrity issues (Kostadino, 2018).

Hexadecimal dump or JTAG. This level of extraction is considered as physical extraction, where the data present in flash memory of mobile device is fetched by forensic tool which retrieves majority of data access. The deleted data from the mobile phone is accessed at

this level with the help of Joint Test Action Group (JTAG) .copying the data and converting them as image file formats from the devices which are locked does have minor damages.

Chip off. The name of the classification indicates that it is a physical extraction of the data by physically removing the chip from the device and analyzing the chip data using various forensic tools. Various models do have distinct memory chips and does require technical expertise. The information retrieved from memory is stored in the form of raw format where examiner is required to decode and translate it into understandable information. To perform this level of extraction, forensic examiner must be well trained regarding the process of isolating the chip and may have the chance of damaging the device by breaking the chip. The below figure depicts typical chip removed from a mobile device (Iris, 2016).

Figure 3: The chip of a mobile device (Zentek, 2004).

Micro read. This method is the toughest of all the methods since the examiner must observe the logical operations on the gates on a microchip with the help of electronic microscope to note down all the binary information and convert into ASCII characters. This method needs highly qualified examiner who have in-depth knowledge of the microchip architecture along with logical operation that a microchip can perform. This process of retrieving the data takes lot of time and even it's costly when compared to all the other methods. There is no commercial microchip tool-related this type of method to analyze the data.

Data Acquisition Techniques

Data acquisition techniques are mainly used to extract the data from the mobile devices without modifying the device data and functionalities. There are three types of data acquisition classifications: i.e., Physical Acquisition, Logical Acquisition and Manual Acquisition. All these methods are further discussed in detail in Acquisition methods used in iPhone as well as android devices.

Overview of iPhone

To perform forensics on iPhone, any forensic examiner needs to know all the details about the iPhone device such as different models released over a period, internal components of the iPhone, knowledge about operating system and touch interface etc. The knowledge on these elements helps forensic investigators to identify where the data is stored,what type of data can be extracted, and which forensic tool is required to work on iPhone model.

iPhone models. The below figure illustrates the specifications of all iPhone models that are manufactured by Apple company starting from year 2007 to 2017.

Table 1: iPhone models (Axon, 2017).

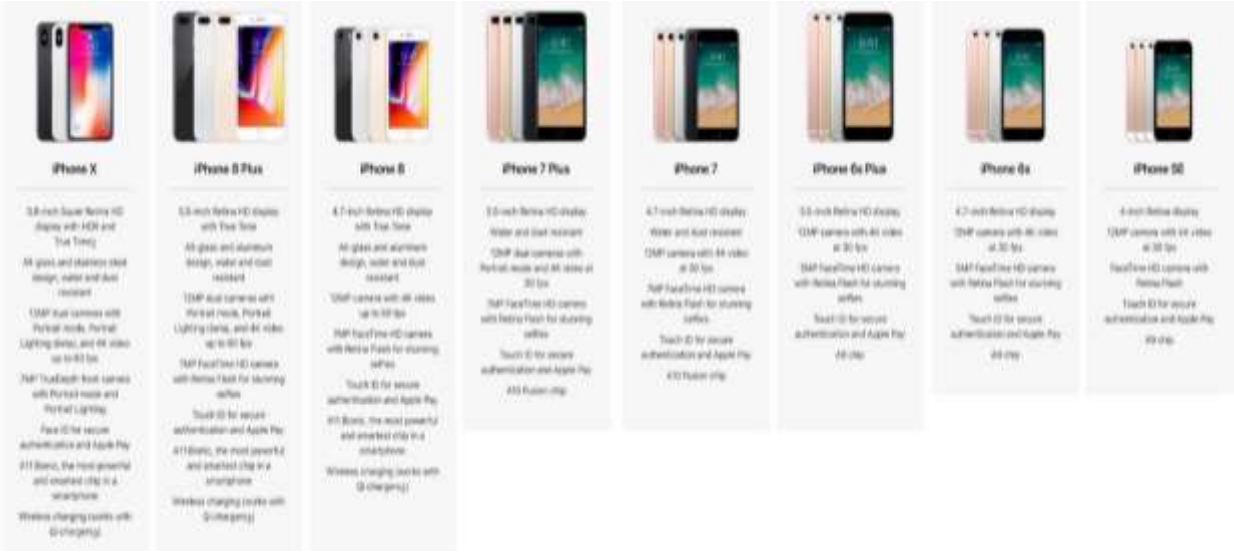


Table 2: iPhone evolution chart (Morgana, 2009).

								
	iPhone	**iPhone 3G**	**iPhone 3GS**	**iPhone 4**	**iPhone 4S**	**iPhone 5**	**iPhone 5c**	**iPhone 5s**
Code Name	M68	N82	N88	N90	N94	N41	N48	N51
Model Name	iPhone 1,1	iPhone 1,2	iPhone 2,1	iPhone 3,1	iPhone 4,1	iPhone 5,1	iPhone 5,3	iPhone 6,1
OS	iPhone OS 1.0	iPhone OS 2.0	iPhone OS 3.0	iOS 4	iOS 5	iOS 6	iOS 7	iOS 7
Screen Size	3.5-inch 480x320 at 163ppi	3.5-inch 480x320 at 163ppi	3.5-inch 480x320 at 163ppi	3.5-inch IPS 960x640 at 326ppi	3.5-inch IPS 960x640 at 326ppi	4-inch 1136x640 in-cell IPS LCD at 326ppi	4-inch 1136x640 in-cell IPS LCD at 326ppi	4-inch 1136x640 in-cell IPS LCD at 326ppi
System-on-chip	Samsung S5L8900	Samsung S5L8900	Samsung APL0298C05	Apple A4	Apple A5	Apple A6	Apple A6	64-bit Apple A7, M7 motion c-processor
CPU	ARM 1176JZ(F)-S	ARM 1176JZ(F)-S	600MHz ARM Cortex A8	800MHz ARM Cortex A8	800MHz dual-core ARM Cortex A9	1.3GHz dual-core Swift (ARM v7s)	1.3GHz dual-core Swift (ARM v7s)	1.3GHz dual-core Cyclone (ARM v8)
GPU	Power VR MBX Lite 3D	Power VR MBX Lite 3D	PowerVR SGX535	PowerVR SGX535	PowerVR dual-core SGX543MP4	PowerVR triple-core SGX543MP3	PowerVR triple-core SGX543MP3	PowerVR G6430
RAM	128MB	128MB	256MB	512MB	512MB	1GB	1GB	1GB DDR3
Storage	4GB/8GB (16GB later)	8GB/16GB	16GB/32GB	16GB/32GB	16GB/32GB/64GB	16GB/32GB/64GB	16GB/32GB	16GB/32GB/64GB
Top Data Speed	EDGE	3G 3.6	HSPA 7.2	HSPA 7.2	HSPA 14.4	LTE/DC-HSPA	LTE/DC-HSPA	LTE/DC-HSPA
SIM	Mini	Mini	Mini	Micro	Micro	Nano	Nano	Nano
Rear Camera	2MP	2MP	3MP/480p	5MP/720p, f2.8, 1.75µ	8MP/1080p, f2.4, BSI, 1.4µ	8MP/1080p, f2.4, BSI, 1.4µ	8MP/1080p, f2.4, BSI, 1.4µ	8MP/1080p, f2.2, BSI, 1.5µ
Front Camera	None	None	None	VGA	VGA	1.2MP/720p, BSI	1.2MP/720p, BSI	1.2MP/720p, BSI
Bluetooth	Bluetooth 2.0 + EDR	Bluetooth 2.0 + EDR	Bluetooth 2.1 + EDR	Bluetooth 2.1 + EDR	Bluetooth 4.0	Bluetooth 4.0	Bluetooth 4.0	Bluetooth 4.0
WiFi	802.11 b/g	802.11 b/g	802.11 b/g	802.11 b/g/n (2.4GHz)	802.11 b/g/n (2.4GHz)	802.11 b/g/n (2.4 and 5GHz)	802.11 b/g/n (2.4 and 5GHz)	802.11 b/g/n (2.4 and 5GHz)
GPS	None	aGPS	aGPS	aGPS	aGPS, GLONASS	aGPS, GLONASS	aGPS, GLONASS	aGPS, GLONASS
Sensors	Light, accelerometer, proximity	Light, accelerometer, proximity	Light, accelerometer, proximity, compass	Light, accelerometer, proximity, compass, gyroscope	Light, accelerometer, proximity, compass, gyroscope, infrared	Light, accelerometer, proximity, compass, gyroscope, infrared	Light, accelerometer, proximity, compass, gyroscope, infrared	Light, accelerometer, proximity, compass, gyroscope, infrared, fingerprint identity
Mic	Single	Single	Single	Dual	Dual	Triple	Triple	Triple
Connector	30-pin Dock	30-pin Dock	30-pin Dock	30-pin Dock	30-pin Dock	Lightning	Lightning	Lightning
Size	115 x 61 x 11.6 mm	115.5 x 61.8 x 12.3 mm	115.5 x 61.8 x 12.3 mm	115.2 x 58.6 x 9.3 mm	115.2 x 58.6 x 9.3 mm	123.8 x 58.6 x 7.6mm	124.4-8 x 59.2 x 8.97mm	123.8 x 58.6 x 7.6mm
Weight	135 g	133 g	135 g	137 g	140 g	112 g	132 g	112 g
Battery	1400 mAh	1150 mAh	1219 mAh	1420 mAh	1430 mAh	1440 mAh	1440 mAh	TBD
Colors	Black (and aluminum)	Black/White	Black/White	Black/White	Black/White	Slate/Silver (2-tone)	Green/Pink/Blue/ Yellow/White	Gold/Silver/Gray (2-tone)
Price	$499/$599 on contract (no subsidy)	$199/$299 on contract	$199/$299 on contract	$199/$299 on contract	$199/$299/$399 on contract	$199/$299/$399 on contract	$99/$199 on contract	$199/$299/$399 on contract
Availability	4 countries, 4 carriers by YE2007	70 countries, 16 carriers by YE2008	80 countries by YE2009	90 countries, 185 carriers by YE2010	70 countries, 100 carriers by YE2011	100 countries, 240 carriers by YE2012	100 countries, 270 carriers by YE2013	100 countries, 270 carriers by YE2013

Internal components of an iPhone. iPhone 6s is used in this paper for performing forensic analysis. This device consists of Apple A9 processor and M9 motion coprocessor with storage variations from 16 Gb,64 GB and 128GB. This device holds 12MP Back camera with 4K video recording and a 5MP High definition front camera. This device does have Touch Id enhancing security features. Along with specifications it is important to consider hardware details of iPhone since the hardware parts present in iPhone are manufactured from different manufacturers. The internal components of an iPhone 6s are dismantled and shown in the figure below.

Figure 4: Dismantled components of an iPhone 6s (ifixit, 2015)

iPhone file system. Forensic analysis on mobile devices become struggle free if the knowledge on file system used in the mobile devices are understood by a forensic examiner. HFSX is the file system used in all the iPhones and Apple iOS devices. It is similar to HFS Plus file system with some differences. The detailed information about the HPS Plus file system is explained below.

The HFS plus file system. With the help of HPS (Hierarchical File System) designed by Apple in 1996, the storage of datasets is increased to larger volumes. Each volume is classified into 512bytes of logical memory block and is equivalent to the size of a physical block of 512 bytes. All these individual blocks stacked together to get data in a more effective way. Since HPS file system uses 16-bit value to determine all these blocks located in allocation blocks which is a disadvantage leading to reduce the number of blocks to 65,535 (mahalik, 2016). To

overcome these disadvantage Apple introduced new file system for extension of old file system as HFS Plus file system where this file system uses 32-bit value to determine the location of these allocation blocks in which the 512bytes logical memory are stacked together.

With the Use of the HPS plus file system all the improvements in checking and allocating data in effective way got achieved. The method of monitoring each operation to the disk and avoiding the file corruption is known as journaling. This phenomenon is default in HPS Plus file system which is an advantage. The HFS Plus volume is classified into different internal assemblies to supervise the data in a proper manner. The volume structure has different sections and are described below:

- *Reserved file*. This file has a space of 512 Bytes and used by Apple while creating the file system process and cannot be utilized for other purposes.
- *Alternate volume header*. This section is related to the backup of volume header used when disk needs repair and it also works as a support file for the volume header to perform functions when the volume header is not in operation.
- *Startup file*. This file stores the information about the booting process and helps in starting the system when the system is not installed with HFS plus file system.
- *Attribute file*. This file stores the records of all the data attributes such as inclined, fork and extension data.
- *Extents overflow file*. This file performs three operations i.e. To find the data that is present in the allocation block, stores information about the allocation block when the size of the block exceeds its limit, records the block which is not performing in optimum condition.

- *Catalog file*. This file stores all the information about all the file location details and their functionalities in this volume of a file system.

The below figure illustrates the volume structure of an HFS Plus file system:

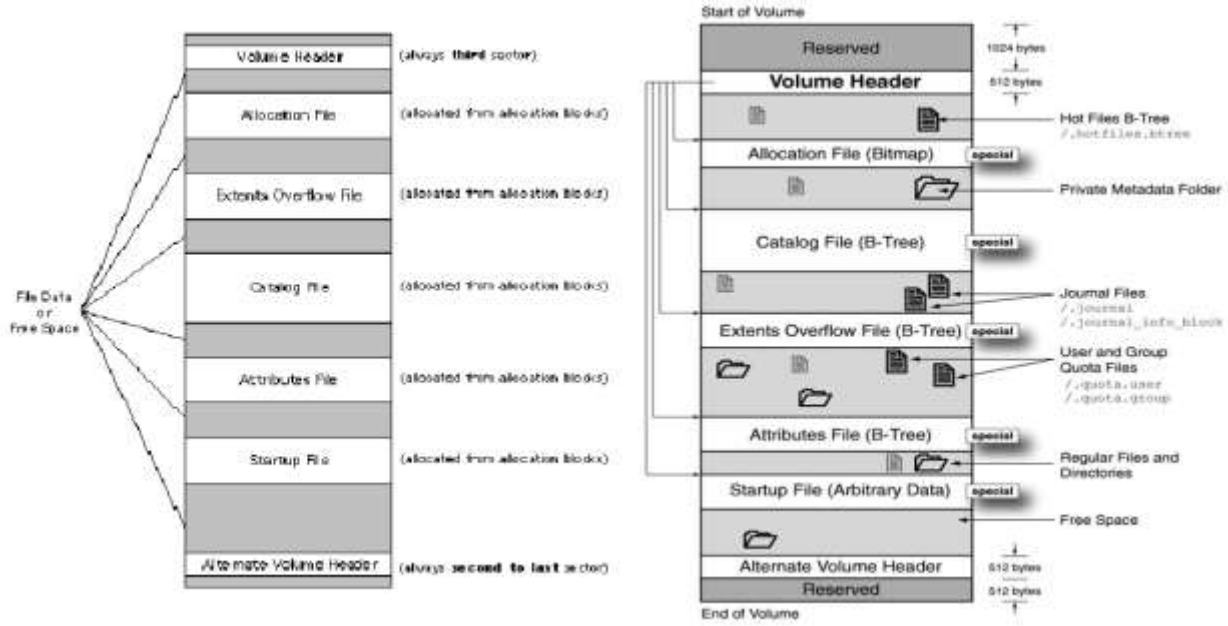


Figure 5: The HFS plus volume structure (ADC, 2000; Flylib, 2017).

- *Allocation file*. This file checks whether the allocation blocks are in use or free to process the data. It does this checking process with the help of placing each bit value to allocation block and if the bit value is high then the corresponding block arranged with the bit is in use.
- *Volume header*. It consists of all the information about the volume including the size of the blocks, details of creation of volume, data about the data of special files.
- *Reserved file (1024 bytes).* This file is utilized for storing information about the bootload process. Compare to space allocated for the manufacturing process at the bottom layer which has 512 bytes, this file has the size of 1024 bytes allocated for storing the information (Proffitt, 2012).

iPhone operating system overview. Operating systems in mobile devices are one of the important criteria where a forensic examiner should be familiar with working on iPhone forensics and retrieve the data. iOS is the operating system for iPhone which is created and developed Apple Inc. Firstly the operating system was named as iPhone OS, later it was renamed as iOS to indicate that some of the Apple devices run through same operating system and can be categorized as Apple iOS devices. Presently all apple products such as iPod touch, iPad, iPhone run through iOS operating system. More than 2.2 million iOS applications are developed and are available in Apple's app store in which 1 million of the applications belong to iPads. This operating system does have an additional feature named Voiceover which can read all the information present on the screen along with buttons, icons, links, the for people with vision and hearing disabilities (Wikipedia, 2018).

The Architecture of iOS

iOS operating system works as an interface between the hardware components and the applications that are accessible on the touch screen. The iOS architecture is classified into four layers.

The below figure depicts the layers of iOS architecture:

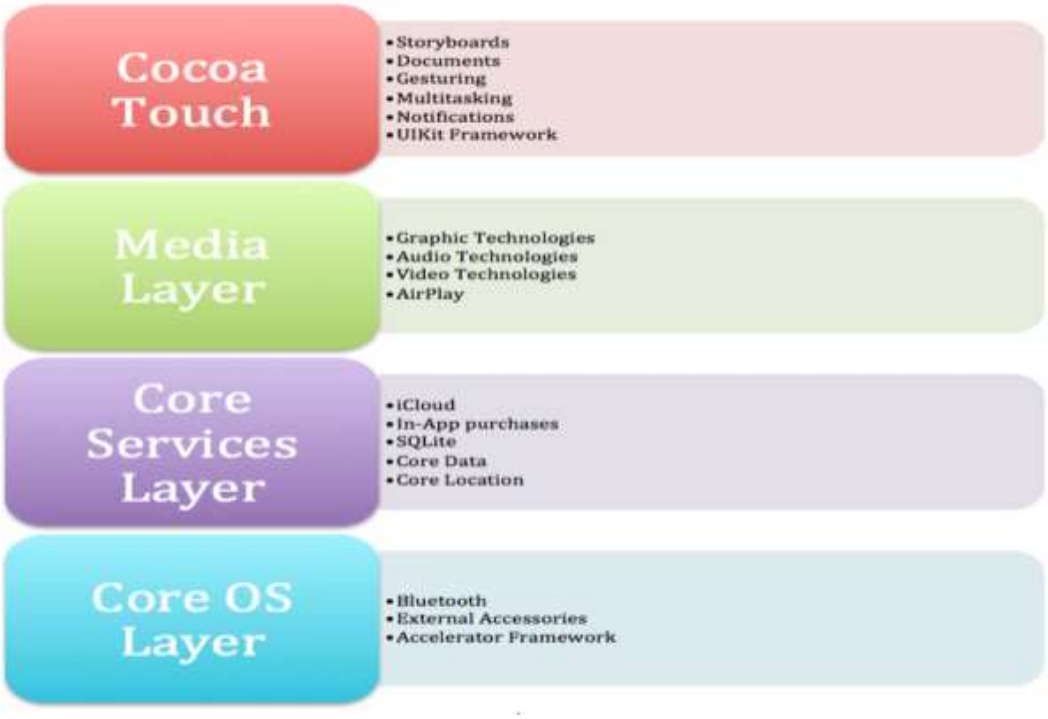


Figure 6: iOS architecture (Gondi, 2015).

- *Cocoa touch layer*. This layer consists of user interface framework for designing application-based programs to work on the operating systems such as storyboards, gesturing, multi-tasking, Notifications and UIKit frameworks. Cocoa Touch is built with main classes that work in Objective-C, an object-oriented language that runs at optimum speed.
- *Media layer*. This layer utilizes all the technologies including graphics, Audio, Video, Airplay to design the apps that provide good layouts, graphical designs and optimum sound qualities. Some of the technologies used are core animation, Text Kit and core text, OpenAL, Core Audio, UIImagePickerController, AV-Foundation (Gondi, 2015).
- *Core services layer*. This layer mainly focuses on providing system services for the apps and helps in supporting all the features such as location, iCloud, Social media.

Some of the features included in this layer are peer to peer services, SQLite, In-App Purchase, XML Support, etc.

- *Core OS layer*. Core OS layer is the fundamental layer and is the first layer that is designed and provides all the services such as memory management, external accessories, Bluetooth functionalities, handling of file systems and Accelerator frameworks.

Architecture of iOS Security

Unlike all the mobile devices, iOS security is designed at the core of the operating system giving one of the most secure mobile devices. Considering all the security risks that appeared in desktop versions, Apple-designed more secure systems in iOS mobile devices. iOS security works on both software and hardware at same time to give best security with simple user experience. Many of the security features cannot be configured by the user and are default so that the app developers or any technologies do not have to worry about configuring security separately and user cannot disable any kind of security feature (Inc, 2018b).

The below figure explains about the pictorial representation of the iOS security architecture:

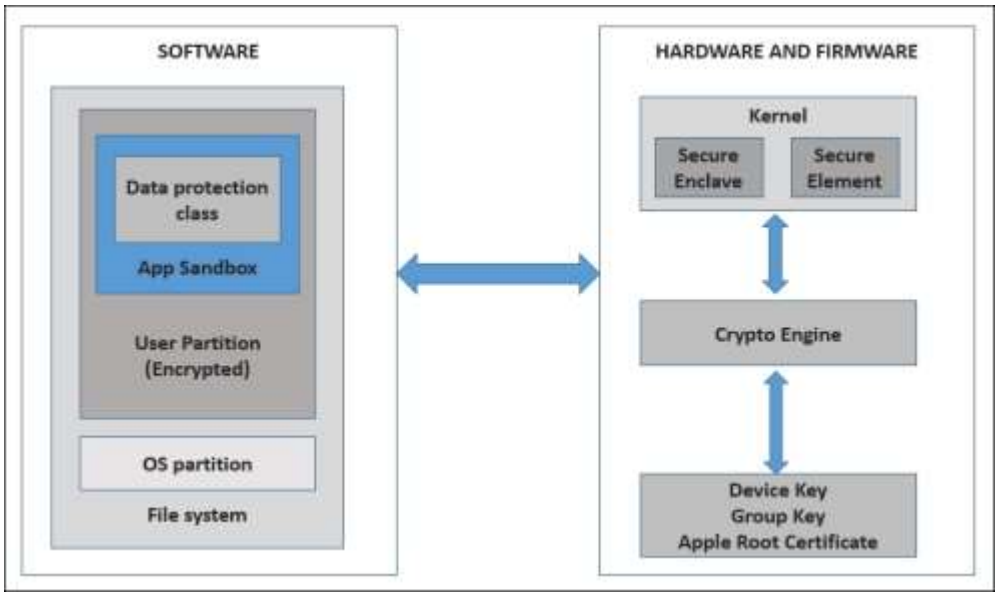


Figure 7: iOS security architecture (Velu, 2018).

Software section security. Security designed at the software section consists of series of layers wrapping one over the other starting with data protection classes, App sandbox, user partition, OS partition and file system.

Hardware section security. Security designed at the hardware section consists of series of layers binding one over the other including kernel (secure enclave and secure element), crypto engine, Device key, Group key and Apple root certificate.

Some of the security features are discussed below:

- ***Data protection***. Any new file created in the iOS Device is categorized into a class from the app that builds it. An encryption key is generated to connect with the user's passcode. When a user locks the device after few seconds the decrypted key used to load the file data or view the data will be deleted making inaccessible to anyone until the user opens with the help of passcode or uses his face id to open the lock of device. Thus, data can be protected from offline attacks.

- ***App sandbox***. Sandbox is the method of restricting access to files at a certain level. so, when all the applications installed on the iPhone are restricted to at some level thereby providing inaccessibility from app to others. It also restricts access to network resources.
- ***Encryption***. The encryption techniques used in iOS devices are different and unique from all other devices. Unlike other devices, iOS device uses its entire file system to be encrypted using an encryption key, which is present in between the OS and hardware level. This is the main reason where some of the techniques used in extracting the data such as JTAG and chip off are difficult to perform and retrieve data on iOS devices (Inc, 2018b).
- ***Passcodes***. Passcode is the basic security feature present in every mobile device. It does not allow the unauthorized access by any other person except the owner of the mobile device. iPhones provide simple to complex passcodes and now in newer versions of iOS, they provide touch ID and even fingerprints represent passcode in background. The latest iPhone X uses face ID as the prime security access for secure access to the mobile.
- ***Data wipe***. Data wipe procedure in an iOS device is designed in such a way that if the user wanted to delete data and reset the phone, there is an option provided in the setting and by clicking the option the data will be completely erased including the encryption key that is used to secure the data such as user settings and information. This makes impossible to retrieve the data even with the forensic investigation procedures (Inc, 2018a).

- ***Activation lock***. Activation Lock is the security feature that enables the lock for any iPhones with the operating system iOS 7 or later versions. This feature is enabled when the phone is enabled to "Find My iPhone" mode and it requires user's Apple ID and password to turn off this mode and even to perform any forensic analysis on the device to retrieve or erase the data.

Jailbreaking

This is a procedure to remove all the restrictions created by iOS device operating systems leading to install unauthorized apps breaking the rules provided by Apple's App Store. It also allows to run code on the device and achieve root access. This is also one of the methods that help in forensic investigation to retrieve the data but by jailbreaking the device, the user loses device warranty and the device will be jailbroken completely and cannot be restored to its original position. All the iPhones which are jailbroken will have higher chances of security issues and is also helpful a forensic examiner to perform Physical and Logical Acquisitions without having disturbances.

Overview of an android device. Android is open software platform for mobile development which is projected to be a complete stack that adds everything starting from the operating system passing along the middleware and finally ending to applications (Alison, 2019). Introduction to an overview of the architecture of the Android platform with some of the key principles that are underlying its design are discussed below.

Architecture of Android device. The below diagram depicts the android architecture covered with lot of boxes with various levels of design.

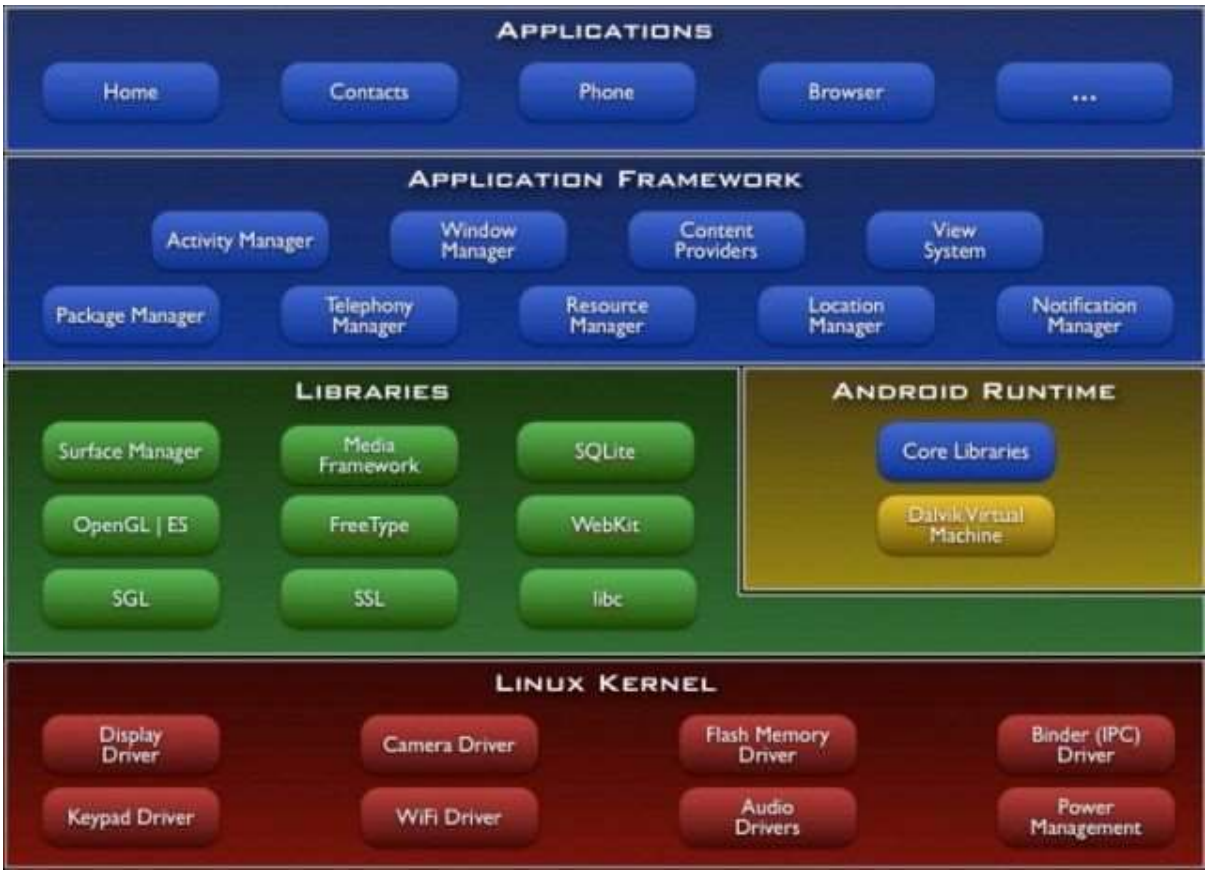


Figure 8: Architecture of android (John, n.d.).

Linux kernel. Starting from the bottom to up top, The architecture is based on the Linux 2.6 Kernel. It is used as the hardware abstraction layer where an Original equipment manager (OEM) starts building this layer first and then gets all the drivers required in the place.

Figure 9: Linux kernal layer (Sharma, n.d.).

The reason for using the Linux as the hardware abstraction layer is that it gives a successful plan of driver model established in existing android models. These successful existing

proven model runs through a long period of time that includes memory management, process management, along with concrete security model, well-designed networking combined to form one of the best-operating systems infrastructure (Dhinakaran Pandiyan, n.d.).

Native libraries. The next level up to design block is native libraries, where a lot of the core power of the Andriod platform comes from. Everything colored with green in the picture is basically written in C and C++. The below picture shows the architecture of native libraries consisting of all the blocks including Surface manager, medial framework, SQLite, OpenGL | ES, Free type, Webkit, SGL, SSL, libc.

Figure 10: Libraries in android architecture (Benny, 2014).

- ***Surface manager*.** The surface manager is used for composing different drawings surfaces onto the screen. In other words, it is responsible for combining different windows with the various applications connected to it in different processes performed during different periods of time providing a perfect level of pixels aligned in order.
- ***OpenGL/ES.*** it is one of the cores of graphics libraries which is a three-dimensional library with the proper three-dimension chip present could process the software implementation that is hardware accelerable (Dhinakaran Pandiyan, n.d.).

- ***SGL.*** The SGL graphics are for two-dimensional graphics where most of the application drawing is designed. One of the main benefits of android graphics is that both two dimensional and three-dimensional graphics can be combined in the same application (Dhinakaran Pandiyan, n.d.).
- ***Media framework.*** the media framework was designed by PacketVideo that combines the core of media experience. This adds all the audio and video codecs like MPEG 4, H.264, MP3, AAC, etc. to build an improved media experience.
- ***Free type.*** This is mainly used to create different types of fonts while typing in the keyboard layer of the android touch interface.
- ***SQLite***. The implementation of SQLite is used as the core of the most device's data storage.
- ***WebKit.*** it is an open-source browser engine, used as the platform for the browser used in both safari that's been in iPhone as well as android devices which can show well on small screens and on mobile devices.

Android Runtime.

Figure 11: Android runtime (Benny, 2014).

The Android Runtime is specifically designed for android to meet the needs of running in an embedded environment with limited battery, limited memory and limited CPU. There are two components which run in the Android runtime are described below.

- ***Dalvik virtual machine.*** The main component of the Andriod Runtime is the Dalvik Virtual machine that runs DEX files which are referred as the bytecodes that are the results of converting at build time which includes.Class and .JAR Files. These two files combined and converted to form .dex becoming much more efficient bytecode that can run very well on small processors.

  These .dex files use memory very efficiently along with the data structures designed to be shared across the processes whenever possible with a highly CPU optimized bytecode interpreter. The result of these capabilities make the possibility of running the multiple instances of the Dalvik Virtual Machine parallelly at the same time allocating with each of the many processes. (Dhinakaran Pandiyan, n.d.).

- ***Core libraries.*** The blue color indicates that is written in java programming language. This consists of all the utilities, tools, collection classes, IO, that are used in android devices.

Application Framework

Figure 12: Application framework (Benny, 2014).

The application framework is the tool kit that all applications use like the ones that exist in the phone application or home application (Alison, 2019). The entire framework is written in java programming language. These applications can be written by google or an individual developer. Whatever the applications developed using the same framework and same APIs for processing. Some of the main components are discussed below.

- ***Activity manager.*** The activity manager is responsible for managing the life cycle of applications and also keeps track of common back stack in order to maintain the applications that are running with various processes can have a smoothly integrated navigation experience.
- ***Package manager.*** The package manager will keep track of all the applications installed on the device and maintain the details of all the applications. It will be responsible for analyzing the capabilities of the new applications installed.
- ***Window manager.*** This will manage the windows platform when connected to Windows laptop or medium which is an abstraction from most of the java programming language platforms is connected to the lower-level services that are provided by surface manager (Dhinakaran Pandiyan, n.d.).
- ***Telephony manager.*** This manager contains the APIs that are used to build phone applications to improve phone experience. Some of the services that include are voice calls, sim details, phone id, etc.
- ***Content providers.*** These are one of the unique pieces of framework on the android platform that provides applications to share their data with other applications. For example it uses the contact information such as all the contact details, names linked to

the contact and addresses to be available for other application such as WhatsApp to use these details (Dhinakaran Pandiyan, n.d.).

- ***View system:*** It consists of all the building blocks of the User interface (UI) like buttons and lists. It also manages things like event dispatching, layout and drawing.
- ***Location manager.*** This is a part of the framework that gives you the services of geographic location. It is used in applications such as google maps to update the device location accordingly (Dhinakaran Pandiyan, n.d.).
- ***Notification manager.*** It collects all the notifications obtained from all the applications and place them on to the surface manager to get notified by the device user.

Applications.

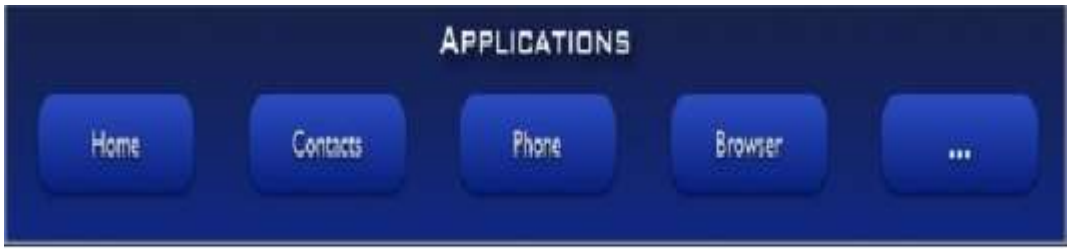


Figure 13: Applications (Benny, 2014).

Applications are written in this section .some of them include the home application, the contacts application, the browser, custom developer and everything at this layer is again using the same application framework provided by the layers discussed above (Alison, 2019).

Andriod Security model. Security Architecture of Andriod is the combination of both Linux kernel and android permission control models. The Linux security model is based on privilege control model where each application that runs on android is given a separate process id (PID) by Linux. So each and every application has its own process id and user id. Where the

Access to the user lever permissions and app permissions while installing from play store or downloaded from google are controlled by android permission control model. The below diagram shows the basic security design of two applications in android platform (Elenkov, 2014).

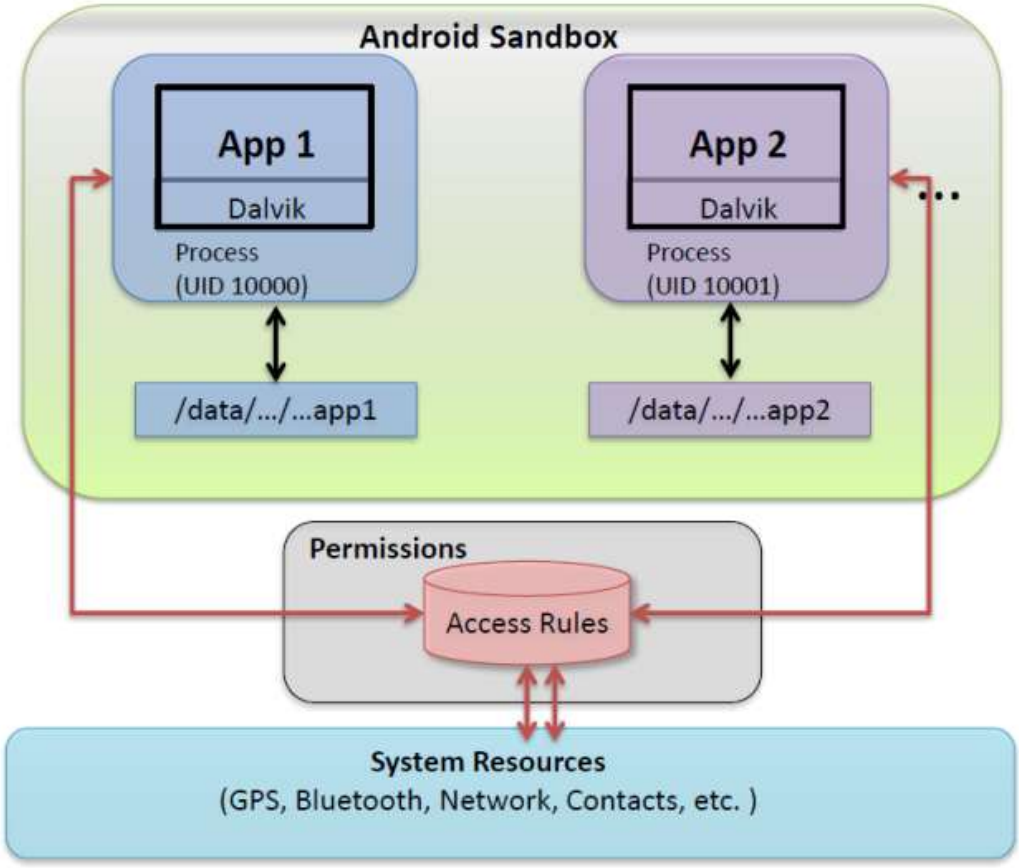


Figure 14: Android security model (Mahadewa, n.d.).

Kernel-based application sandbox. It is implemented by using sperate user IDs and group IDs to isolate applications and their data which is completely based on the traditional Linux's discretionary access control(DAC). It had been augmented recently with SELinux Mandatory Access policy-based control starting from the android version of 4.4 . So each and every application are packed with Dalvik VM and assigned with separate user id and they have

their own data directory for app storage which only the application does have the complete access to read the data (Elenkov, 2014).

Permissions. As the applications are allocated with different sandboxes they would be restricted to view their own app data and the external user can access these data. In order to provide more privileges of granting permissions to use hardware devices such as internet connections, use of camera, operating system level services to improve app technical aspects, Android can control both the hardware and system resources with the help of access rules (Elenkov, 2014).

Once the access for an application is allowed to use hardware resources, it cannot be prevented from stopping in middle or any point of time until the app is completely uninstalled.

Code signing and platform keys. Code signing of an application is to get the authorized signature from the developer when he develops the code. As the Android APK files developed in expansion to the java jar formats. It is signed using the jar packaging format. Whenever the application does have the new upgrade it must be released by the original author who signed while producing the first version of the application and that binds the trust level in between the applications. Platform keys are signatures of system applications that maintain and run android versions (Elenkov, 2014).

Chapter III: Methodology

Introduction

In this section, All the procedures used for research, requirements of the tools and techniques and concepts of Acquisition methods to analyze and compare the differences in extracting the data from an iPhone Vs an android device and focus on the various challenges to overcome while retrieving the data from an iPhone using various forensic methods are explained in detail.

Design of Study

This paper uses both quantitative as well as qualitative approaches for analyzing the data retrieved from the iPhone using various forensic techniques and categorizing the challenges faced during the forensic process over android forensics.

Tools and Techniques Required

The below tools and techniques are mainly used to retrieve and analyze the data on an iPhone and compare the experimental reports generated by various tools during forensic acquisition with results obtained when performed on android forensics.

iPhone Data Acquisition Techniques

Data acquisition techniques in an iPhone can be performed in many ways. Each method will have advantages and disadvantages. The primary principle of the acquisition is done by either bit by bit or directly copying the physical data available on the device. Some of the methods are discussed below.

Physical Acquisition

In every device such as mobiles phones, computers, laptop does have to kinds of memory, i.e., volatile and nonvolatile memory. In iPhone, RAM acts as volatile memory where

all the active applications are stored while processing any function and it is the important memory in which main functions of operating system will take place. Once the device is switched off and again rebooted the important data stored in the RAM such as usernames, passwords and encryption keys are lost and cannot be retrieved. Even though RAM stores important data, it is difficult to acquire data from RAM. Whereas NAND acts as the non-volatile memory where the data stored will present even after the device is rebooted. All the data present in the NAND flash consists of systems files and user's information. Here the physical acquisition is performed on the NAND memory by copying in a bit by bit procedure and copied memory is analyzed in the same way as the traditional methods used in retrieving the data from hard drive.

Compare to the modern hard disk used in the computers, NAND memory available in the iPhone is cheaper and can perform operations faster and even store copious amounts of data. With the invention of the latest iPhones, it is difficult to perform physical acquisition due to enhanced security features such as secure boot chain, encryption of storage and creation of passcode for the device (Mahalik, 2016).

File System Acquisition

The file system in an iPhone is encrypted to provide security to safeguard the information present in the device. The file system acquisition is only performed on the information of user section only and the system section is retrieved by using physical access. Some methods need the device to be jailbroken and some tools need the device to be in DFU mode to perform file system acquisition.

Logical Acquisition

Logical Acquisition is the process of acquiring the data from the unlocked iPhone. This acquisition mainly uses backup files of the iPhone stored in iTunes backup, SQLite Databases,

Property lists, and from other files such as keyboard cache etc. This Acquisition cannot be performed if the device is locked or the access to plist list file is unavailable. To do the extraction, the examiner needs to search for trusted computers in the lockdown file and bypass the passcode to trick the device that it is in unlocked state (Mahalik, 2016).

iOS Forensic Tools

There is a necessity for the forensic examiner to know about the different forensic tools and their features to perform forensic analysis. Every forensic tool does have limitations and backlogs that may not provide all techniques to retrieve the data from an iPhone. Some of the forensic tools that help in extracting the data from an iPhone are discussed below.

Elcomsoft iOS Forensic Toolkit

It one of the forensic tools that helps in retrieving the data for older iOS devices up to iPhone 4 through physical acquisition and can be to retrieve data for latest iPhone devices with 32-bit iOS operating systems if they are jailbroken. This tool can be performed in two ways, i.e., guided mode and the manual mode. The manual mode requires USB dongle purchased along with the toolkit must be connected to the computer whenever the toolkit is performing analysis. Some of the features of EIFT are:

- It can be used to perform physical and logical acquisition.
- The approximate time taken by the toolkit to perform system extraction is about 20-40 mins depending on the different versions of iOS devices.
- The toolkit can retrieve information from iPhone by decrypting the data present in the backups of Apple iTunes.
- It gives information stored in the raw disk image by extracting and gaining access through keys which are encrypted

- It also retrieves the details of keychain items by using the stored keys present in the keys.plist file.
- This toolkit does have the Zero-footprint operation which can retrieve the data without making any alterations to the device contents thereby maintaining integrity.
- This tool records all the steps performed to extract data along with the timestamps making easy for the forensic investigator to proceed with the investigation on the device (Mahalik, 2016).

Oxygen Forensic Detective

It provides advanced analysis compared to the oxygen forensic suite for mobile data extraction.it is available for windows platform, So in order to perform forensics on iPhone this tool needs the support of iTunes.

Some of the features are:

- Provides both logical and physical acquisition.
- Extracts passwords using key chain.
- Jailbreaking and rooting activities can be performed on the iPhone with the help of this tool.
- With the help of additional tool, extracts the deleted data from the SQLite databases.
- It is also featured with built-in formats such as Microsoft Excel, PDF, HTML etc. for creating reports after each analysis along with timestamps (Mahalik, 2016).

Cellebrite UFED Physical Analyzer

It provides support for major security organizations, antiterrorism, law enforcement by retrieving important evidence from mobile phones, PDAs, desktops etc. Same as the other tool kits, Cellebrite UFED (Universal Forensic Extraction Device) Physical Analyzer uses all the

techniques in extracting the data and analyzing the data from the iOS devices. One drawback of this forensic tool is physical extraction is not possible with the iPhone models having A5-A9 chips.

Some of the features include:

- Provides both logical and physical acquisition.
- With the help of file system acquisition, it provides the extraction of keychain items and images found in the root memory.
- Same as the Oxygen Forensic Detective, it is also featured with built-in formats such as Microsoft Excel, PDF, HTML etc. for creating reports after each analysis along with time stamps.
- Unlike all the tools, this tool can extract the data from the iPhone which are not jailbroken but do require unlocked mobile devices.
- Passcode recovery attacks can be performed with this forensic tool.
- The user interface of this forensic tool makes the examiner easy to analyze the data retrieved by reflecting both the physical and logical data on same screen.
- Easy to forward the data from this forensic tool to another tool for further analysis since it generates an additional binary image file (Cellebrite, 2018).

Along with above forensic tools there are many forensic tools available such as iPhone Analyzer, backlight, Magnet Acquire, NowSecureCE, iBackup extractor and the data recovery services including iMyFone D-Back iPhone Data recovery, wondershare Dr. Fone, iMobie Phone Rescue etc. help in extracting the data from the iPhone with various techniques (Hammond, 2017).

Andriod forensic tools.

Andriller. Andriller is a software application with combination of forensic tools that are useful in decoding the passwords and application data on Android devices. The formats andriller would generate reports are HTML and Excel.

Some of the features include:

- It extracts the data automatically and decode the information once the application is installed on the laptop and when the android device is connected without rooting.
- With the help of ADB daemon, andriller can extract the data by bypassing the passcode or the patterns.
- The social media applications such as WhatsApp with the encrypted archived databases can be decrypted and also opens the android backup files (Andriller, n.d.).

Ireparo for Android. It is an android data recovery software application that recovers all the media files such as photos, music, videos and also social media applications data such as WhatsApp, Facebook, Gmail, etc.

Some of the features include:

- It recovers the deleted data on the android devices and restores them on to the specified drive location.
- Whenever the root failure happens for an android device, it can be used to recover all the deleted data that happened during the process.
- It has the capability to even get back the data from when the device is set to the factory reset and the data on the device is completely wiped off (ireparo, n.d.).

Autopsy forensic tool kit. it is a platform that provides graphical interface along with the in-depth analyzer that could be helpful for digital forensics. This application is utilized by law

enforcement, forensic investigators and research scholar as it provides user-friendly application interface who would look for investigation on what could be deleted or archived from the systems, laptops, etc.

Some of the features include:

- It can perform analysis on multiple cases at once, so that examiner may perform forensics on different cases parallelly.
- With the backup file, the database will be created and it would search for all the web activities which help in filtering the user web activity and what are all the websites were used.
- With the help of the information obtained from pictures while taking from camera, it would search and extract the geographical location (Carrier, 2019).

Chain of custody form. Every forensic team in digital forensics will go through this form as it is a chronological document utilized for gathering information about the electronic evidence that covers all the details in sequence from the seizure to reporting the evidence are tracked down and registered with the timestamps. With the help of this form forensic team can maintain the integrity of the evidence collected and also can be prevented from any changes that can be made in evidence after the seizure of device from crime scene (Infosec, 2019).

Case/Incident: _____ Exhibit/Property Number: _____

Date Acquired (mm/dd/yy): ____ / ____ / ____ Time Acquired (24hr): ____ : ____

Location of Seized: _____ Seized By: _____

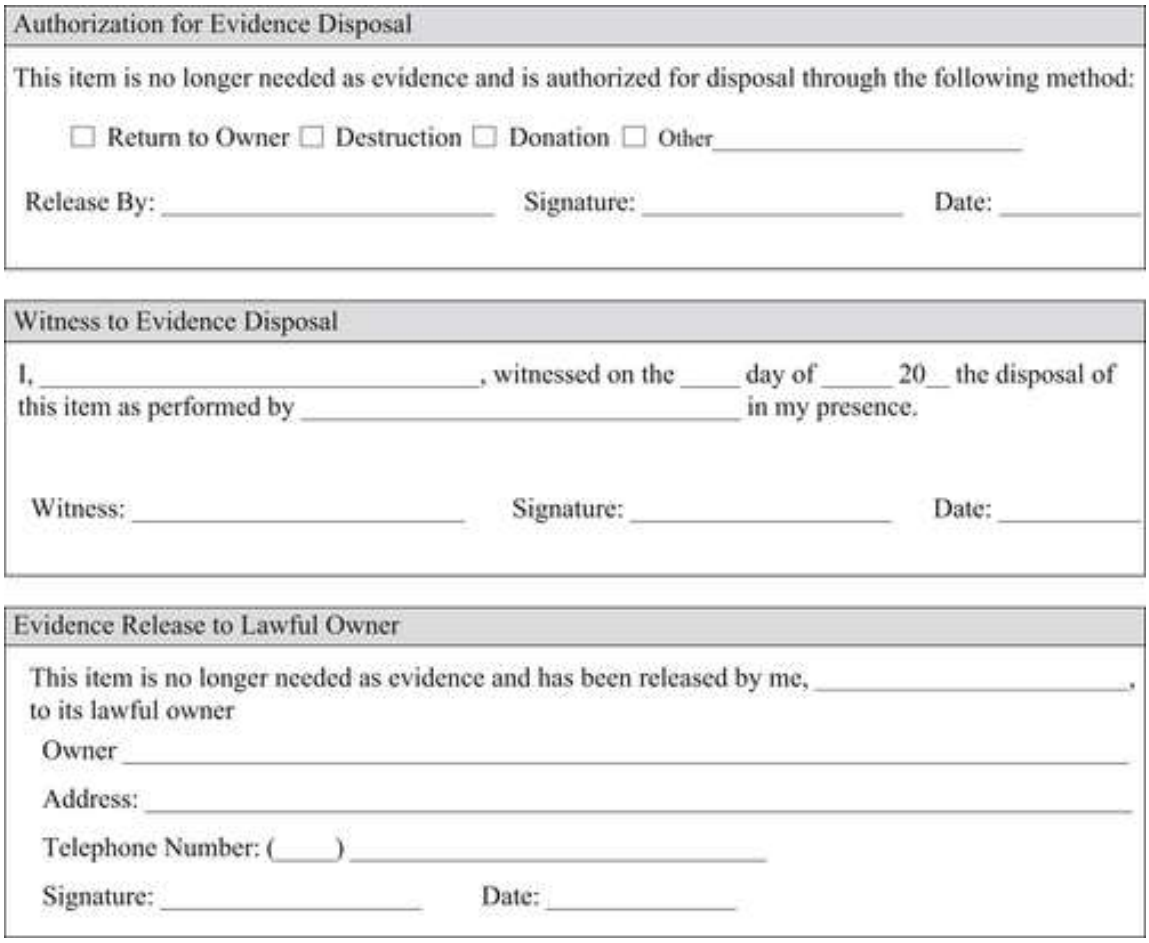
Description of Item (ex. Model, Quantity, Serial #, Condition, Markings, etc):

Notes / Additional Comments:

Chain of Custody			
Date / Time	Released By (Name & Signature)	Received By (Name & Signature)	Comments:

Form 1: Chain of custody form (Sachowski, 2019).

As a forensic examiner, we could also find where the evidence might lie and all the details about where it came from, who created it and what was the state of device when collected from the crime scene that would help us in evaluating the evidence.



Authorization for Evidence Disposal

This item is no longer needed as evidence and is authorized for disposal through the following method:

☐ Return to Owner ☐ Destruction ☐ Donation ☐ Other_____

Release By: _____ Signature: _____ Date: _____

Witness to Evidence Disposal

I, _____, witnessed on the ____ day of _____ 20__ the disposal of this item as performed by _____ in my presence.

Witness: _____ Signature: _____ Date: _____

Evidence Release to Lawful Owner

This item is no longer needed as evidence and has been released by me, _____, to its lawful owner

Owner _____

Address: _____

Telephone Number: (____) _____

Signature: _____ Date: _____

Form 2: Chain of custody form contd (Sachowski, 2019).

From the above screenshot of the form, we can get all the details about who authorized for evidence disposal, who witnessed the Evidence disposal and evidence release details to the owner of the device. It is also important to maintain the whole document with all the links from beginning to end as the court may disapprove if there is any missing link in the form (Infosec, 2019).

Hardware and Software Requirements

Hardware. Mobile devices - iPhone 6S, moto G. Supported USB data cables, pen drive. Laptop – Dell Inspiron 13 -5000 series with 32-bit Intel I7 Processor, Forensic bag.

Software. Operating system -Windows 10, Windows XP (os present in cellebrite touch) Forensic tools for iPhone : Imyphone, Cellibrate, gihosoft, Autopsy forensic tool, Plist editor. Forensic tools for android: Andriller, Ireparo, Cellibrate forensic tool, Autopsy forensic tool. The complete set up is shown in the below figure.

Figure 15: Complete setup for the research.

Budget Information. Total Cost for the project :350 to 400$(for the two phones, Pen drive and the Cellebrite Forensic tool kit). Most of the online applications used are free of cost.

Chapter IV: Data Presentation and Analysis

Introduction

In this chapter, we will conduct experiment with the process of recovering the data with different forensic methods using some of the above-discussed tools for both the android and iPhone. While performing the recovery methods, we would do the analysis how the tools help us recover the data without modifying any data that is captured and discuss the challenged faced while collecting the data.

Data Presentation

In this level, we focus on the gathering the evidence collected in forms of data while performing different methods with different tools such as obtaining information about the emails sent, WIFI passwords, photos, messages etc. all the evidences gathered are then submitted to the court for further legal procedures of the case. As a part of forensic investigation, While collecting the devices from the crime scene, we use faraday bags. The device is placed in a faraday bag so as to restrict from all the networks that device to connect temporarily until we start recovering the data. This would make the two devices hack free from outside world from remote locations. The below picture gives us the idea how the phones are kept in faraday bag.

Figure 16: Faraday bag with the devices.

Installation of Android SDK tools and platform-tools.

1. Download the android SDK platform-tools from the android developer's website by using this link https://developer.android.com/studio/releases/platform-tools

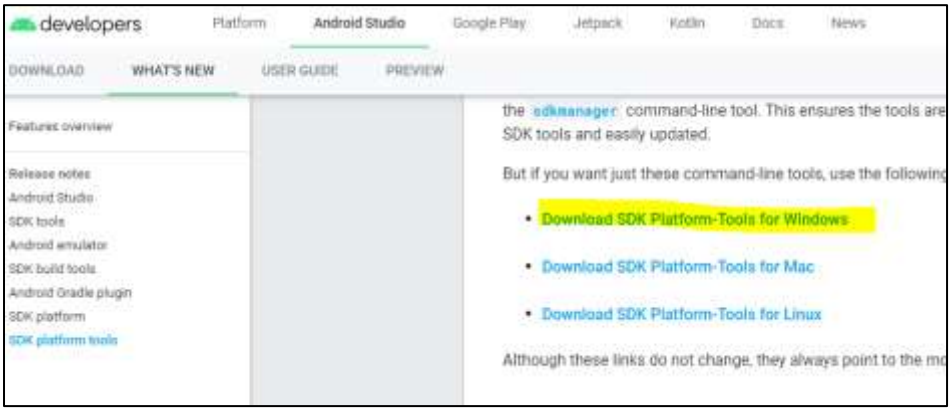


Figure 17: SDK platform tool download webpage.

2. Open the downloaded tools from the location and type left shift and right-click the mouse to open windows power shell.

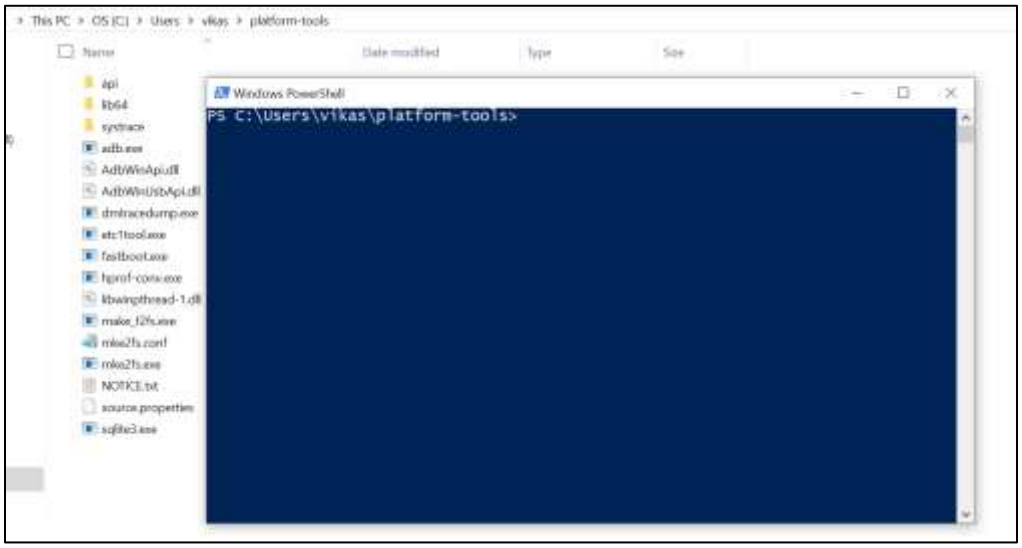


Figure 18: Launching Windows power shell.

Here we use power shell to write ADB commands to set the phone to debug mode.

Setting up the android mobile (Moto g) to recovery mode.

1. In order to set the android mobile to set to recovery mode, Press the volume down button and power button of the phone at the same time to go to boot options.

Figure 19: Boot options for android.

Now by using the volume down button move the selection bar to recovery option and then click the volume up button, the screen would go off and boot to recovery mode as shown in below picture.

Figure 20: Recovery mode selection.

2. To know about the information and details of the phone, press the volume down key for moving the selection bar until it reaches to barcode option and then click volume up button to select the option as shown below.

Figure 21: Barcode information about the device.

3. Connect the phone to laptop via supported USB cable.

Figure 22: Moto g connected to laptop.

Turning on USB debugging mode using Adb package command line. By using below ADB commands check the connectivity of the phone.

- **Adb help** : this command will give you the options about using different commands to perform the task.

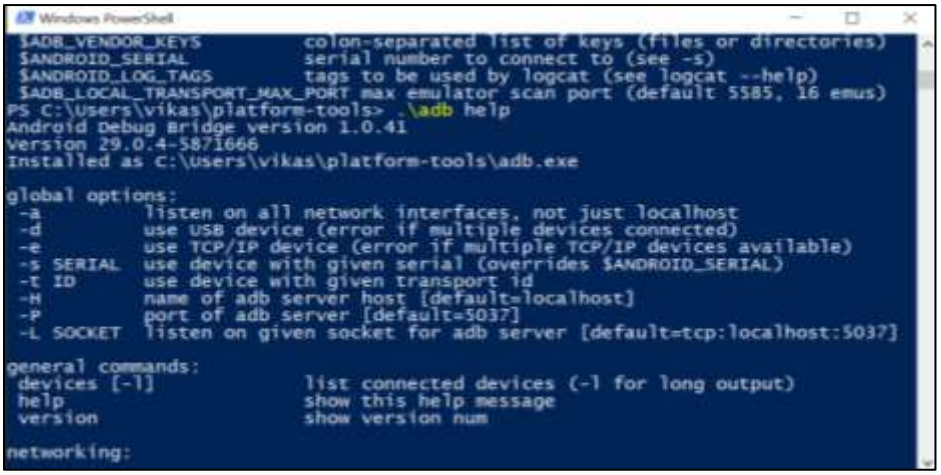


Figure 23: Information about ADB commands.

Now by using below commands enable ADB on the mobile.

.\adb devices: this command would start the daemon and look for the connected devices.

Figure 24: Restarting the daemon.

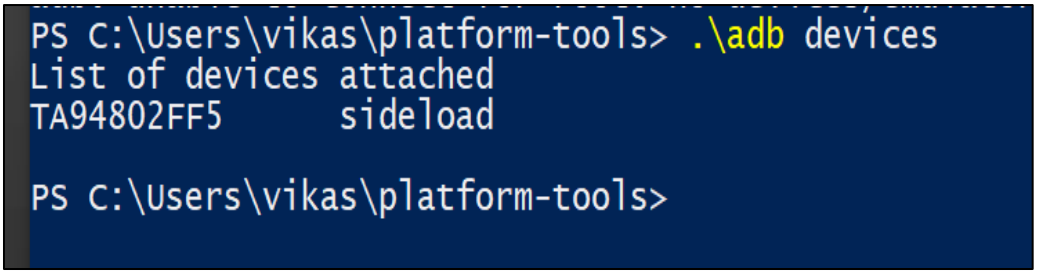


Figure 25: List of devices connected.

- Now use the below commands to enable ADB.

.\adb pull /data/property/persist.sys.usb.config c:\users\vikas\

This command would copy the usb.config file from the device to the location given in the command. Modify the config file in text editor by adding ADB to mtp present in the files and push back to the device by using the below command

.adb shell echo ‘mtp,adb’ > /data/property/persist.sys.usb.config

Push the file to the device to enable adb mode by using the below command.

.\adb pull c:\users\vikas\ /data/property/persist.sys.usb.config

The phone is set to send the package by selecting update package via adp mode. (Firelord, 2011).

Figure 26: Device in sideload mode.

Update the package with sideload command to apply for phone.

.\adb sideload persist.sys.usb.config

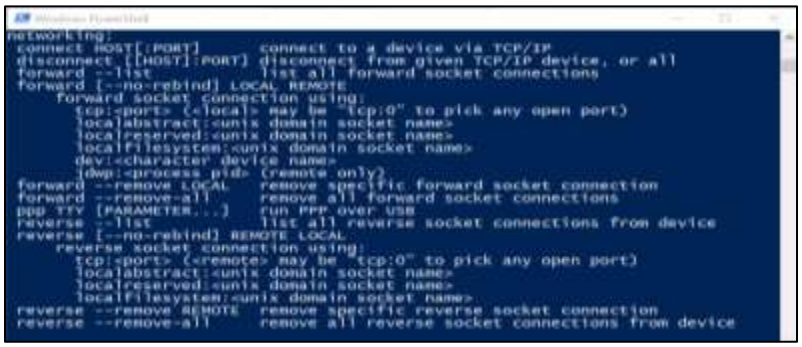


Figure 27: Device enables with ADB mode along with MTP activation.

The device is enabled with ADB mode and has USB debugging mode on. This may not work for the higher versions of the android from 5.0 due the upgrade that fixed the version. As a part of forensics the forensic team will reach out to the individual to ask for the phone password details with the warrant. The conventional way to enable debugging mode in an android phone is shown below.

Conventional way to enable debugging and MTP mode. Go to settings–open for the about phone option and click on the build version for seven times continuously that would release a new developer mode option back in the settings.

Now open the developer options and enable debugging mode on.

Figure 28: Procedure to set USB debugging.

Installation of Andriller.

- Now we download andriller tool from using this website.

  https://www.andriller.com/download/

Figure 29: Andriller setup file on webpage.

- Now we run the downloaded file.

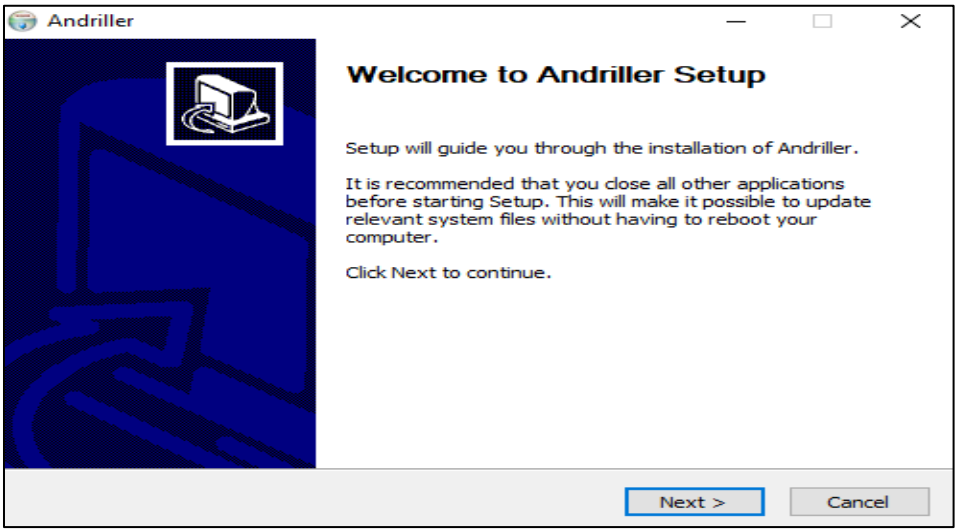


Figure 30: Andriller setup.

- After going through the agreement terms, click on the I agree option.

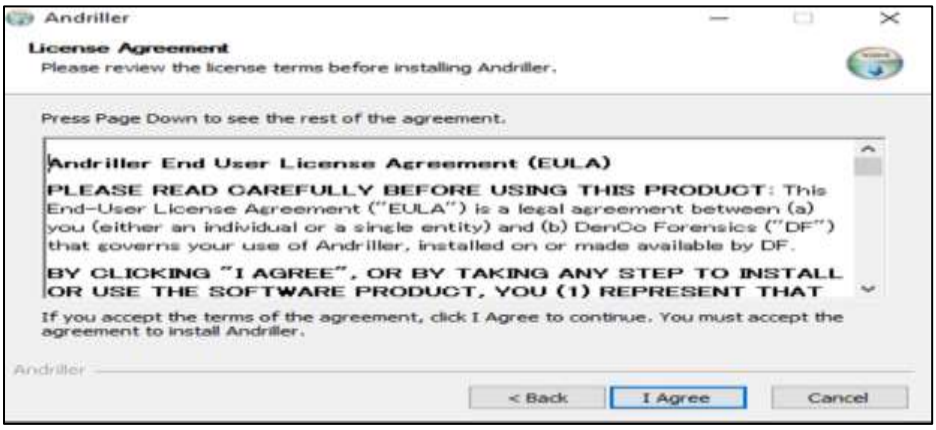


Figure 31: License agreement page.

- Click next to start the installation.

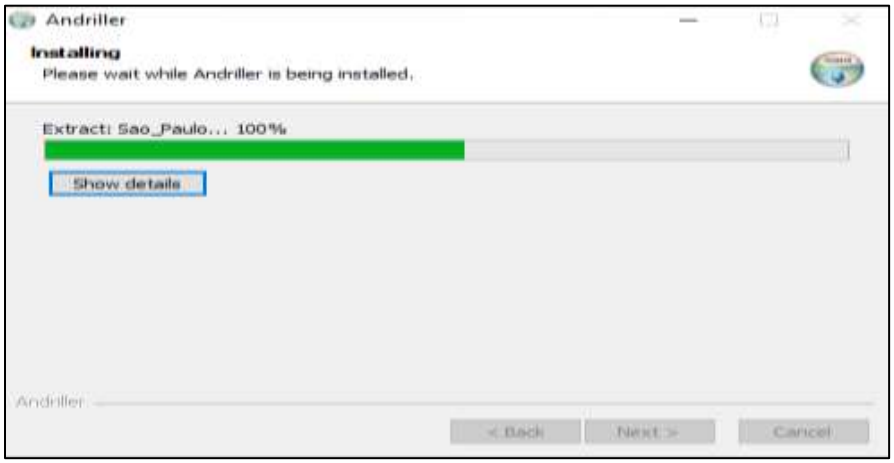


Figure 32: Installation in progress.

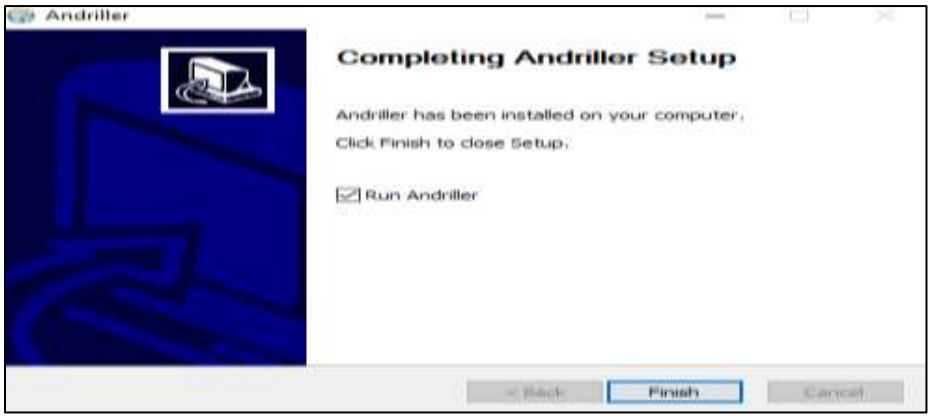


Figure 33: Final step of Andriller setup.

- Open the Andriller tool and connect the android mobile moto g in recovery mode.

Figure 34: Andriller not yet registered.

- As we see the tool is not yet registered with the license key. In order to get the valid license key, click on the help menu on the tool and open license update.it would redirect to the license details.
- Fill out the details in the license tab given on the website.

  https://www.andriller.com/license/get

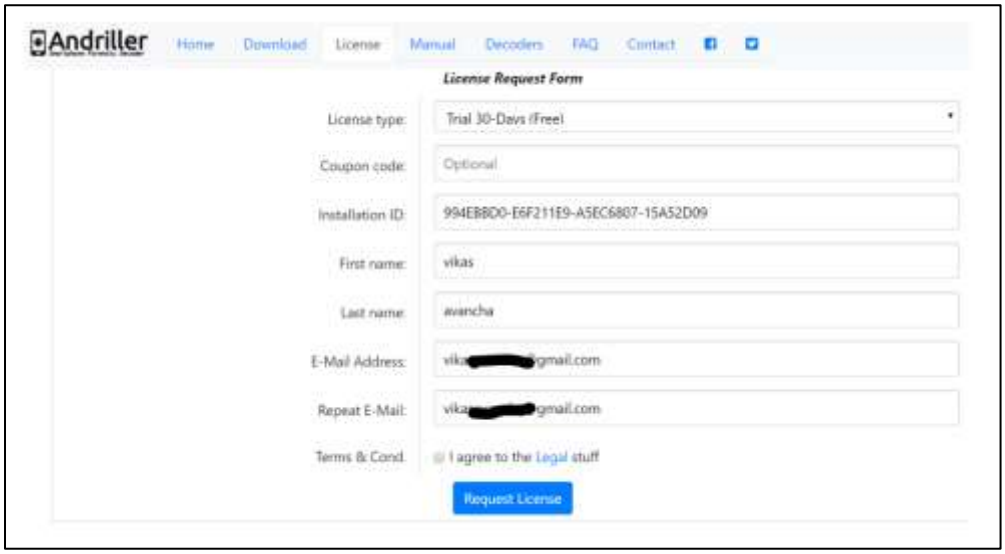


Figure 35: License request form.

- Now submit the request and we would get the license to our email.

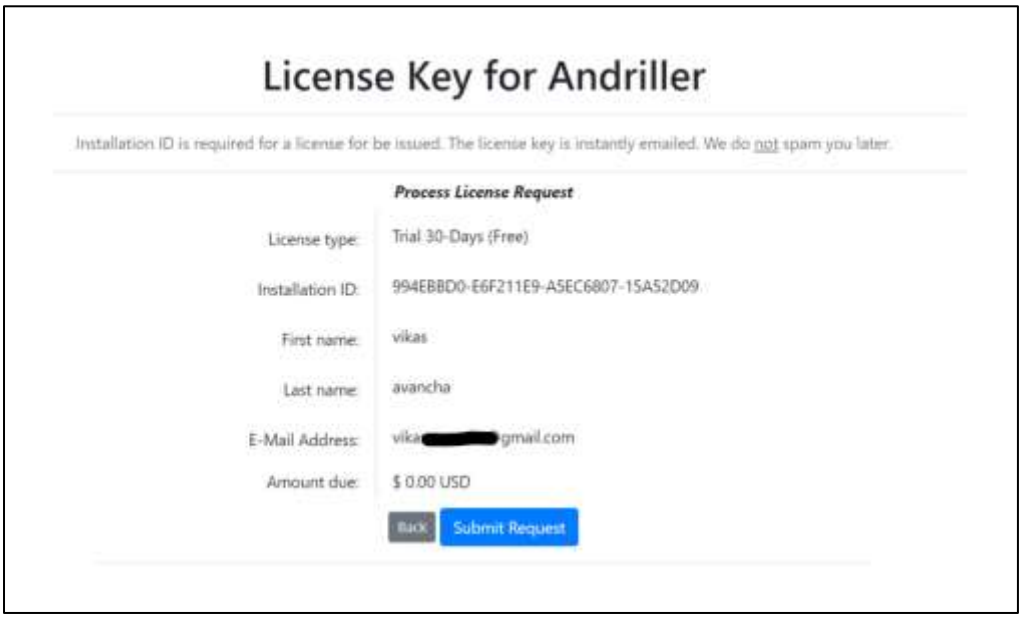


Figure 36: Process license request.

Figure 37: License key for installation.

- Open the email by logging to the email which is given in the request form and we see the email that is sent by the andriller team in below image.

Figure 38: License key sent in email.

- Copy the license key and open the andriller tool and save the license key in the give license key update menu.

Figure 39: Saving the license key on andriller tool.

- Check the connection whether the phone is connected to android tool.

Figure 40: Connecting the device to the andriller tool.

- We could see the serial number attached that confirms the connection is valid. Now select the extract shared storage and use all the method options and start the extraction process.

  Note: The android mobile must be in debugging mode before we perform extraction using the tool.

Figure 41. Detection of the device using serial id.

- The information about the mobile reported and extracted to the laptop as shown in the below images.

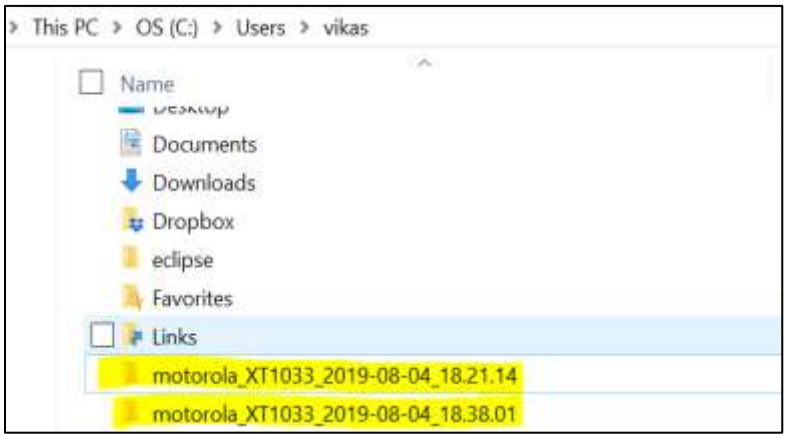


Figure 42: Extracted data from the device using andriller.

Installation of autopsy forensic tool.

- For analyzing the extracted files obtained from andriller tool, we use autopsy forensic tool.
- Use the following link to download autopsy for windows machines.

  https://www.autopsy.com/

Figure 43: Download page of autopsy.

Figure 44: Versions for autopsy download.

- Now open the downloaded file and start installing the autopsy.

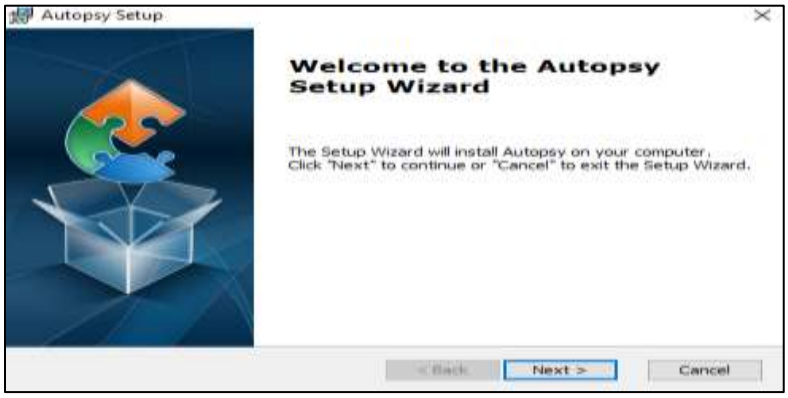


Figure 45: Autopsy setup wizard.

- Click next and select the location for the installation.

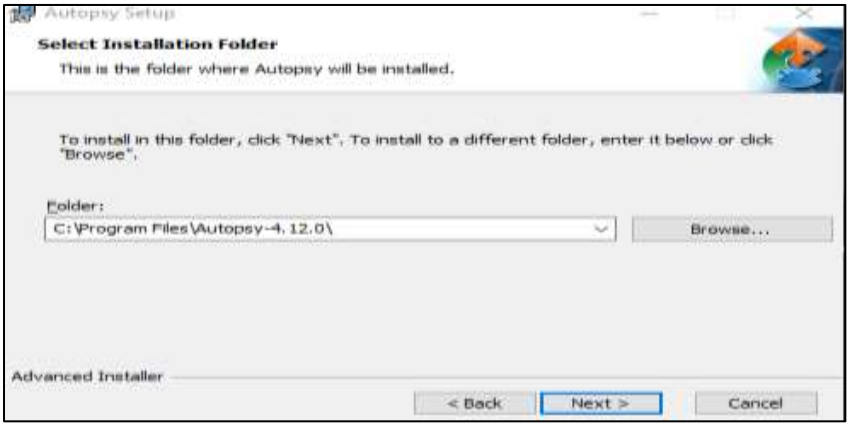


Figure 46: Selection of installation folder.

- Once the process completes, click the finish button and run the autopsy tool.

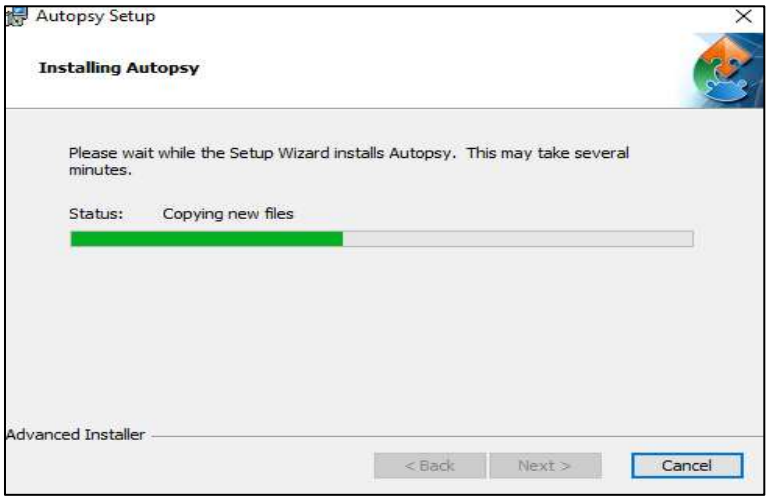


Figure 47: Installation of autopsy in progress.

- Open a new case and give the details about the case we are performing and click the next option.

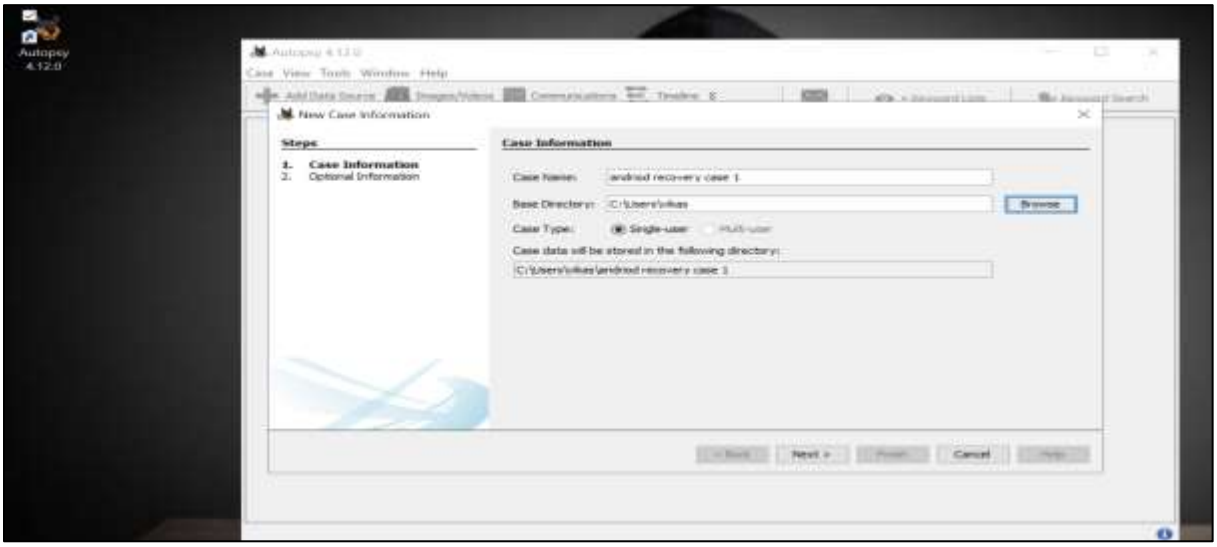


Figure 48: Creation of case in autopsy.

- Select the option as a logical file and click next.

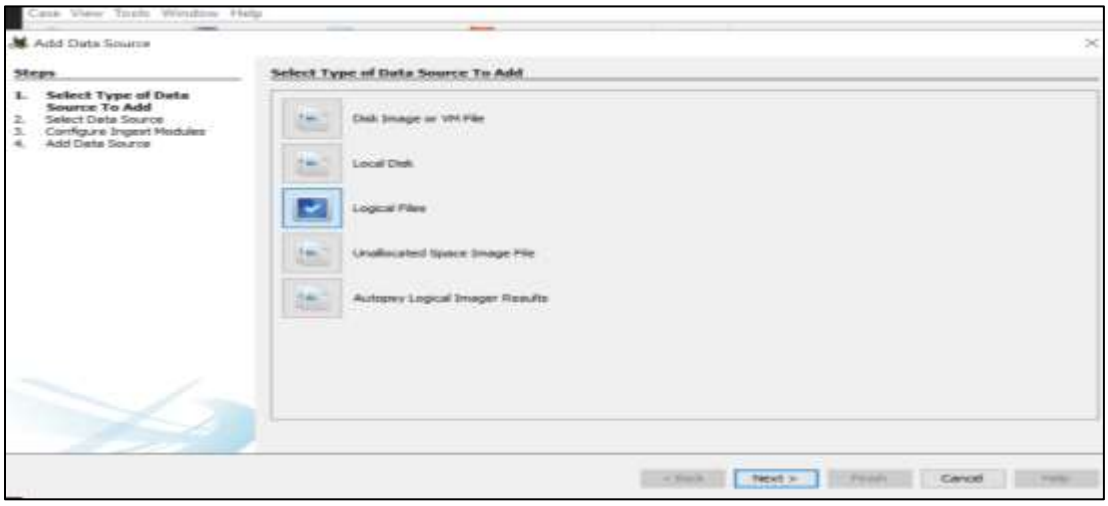


Figure 49: Selection of data source.

- Select the files that are extracted by andriller.

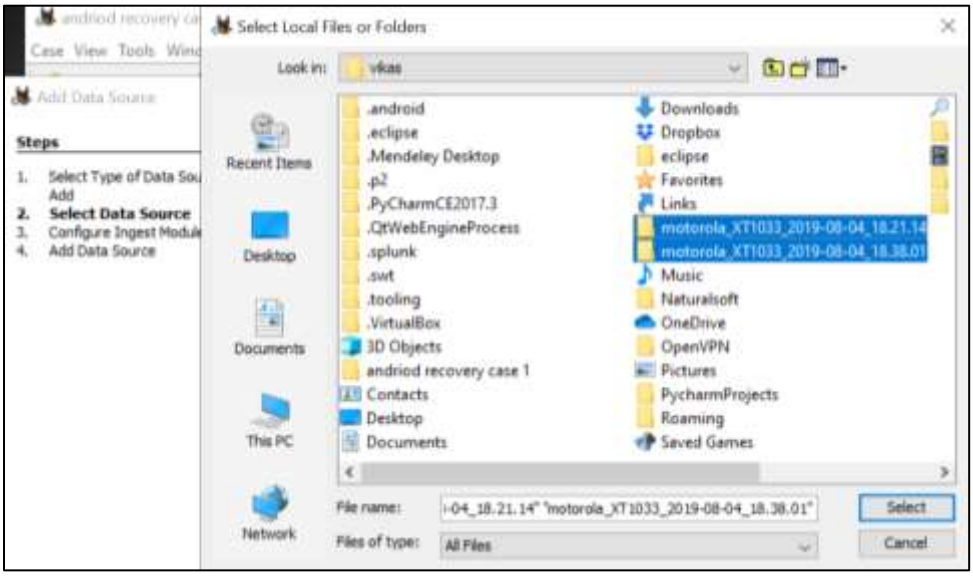


Figure 50: Add the extracted data from andriller.

- Configure all the ingest modules and proceed to further case and create the case.

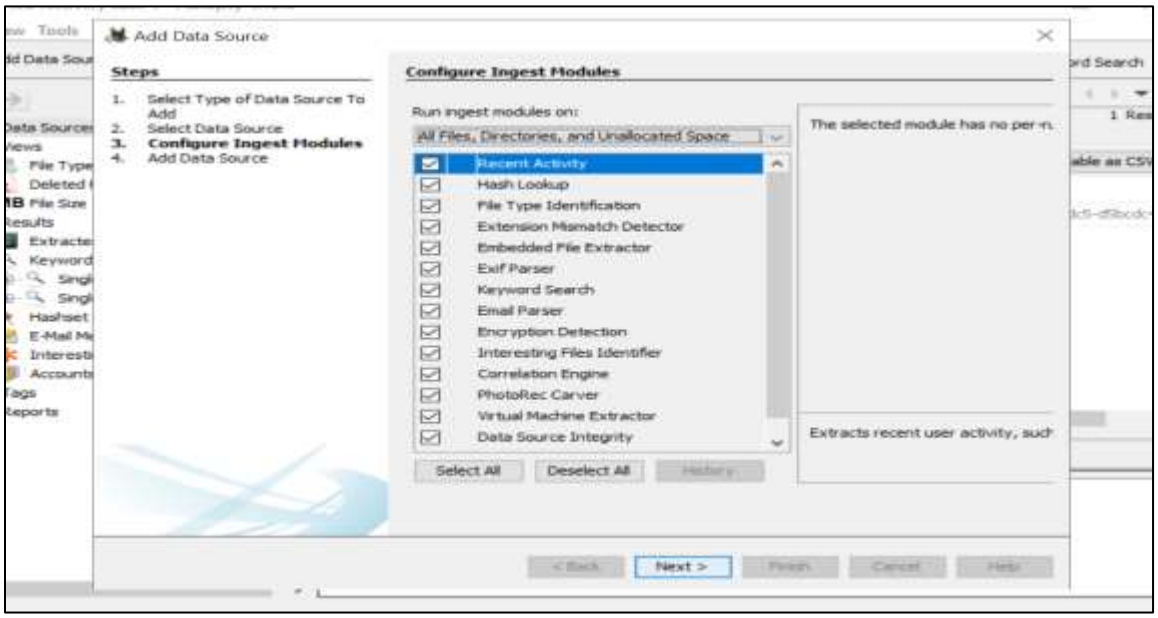


Figure 51: Configure ingest modules.

Installation of ireparo recovery tool for android.

- Download the package for ireparo tool from the website.

  https://www.androidrecovery.com/

Figure 52: Ireparo phone data recovery.

- Run the downloaded application and accept the license agreement and click next.

Figure 53: User license agreement.

- Go through the process and complete installation.

Figure 54: Installation of ireparo in progress.

Figure 55: Launch ireparo tool for android.

- Start the application and select the specified file types:

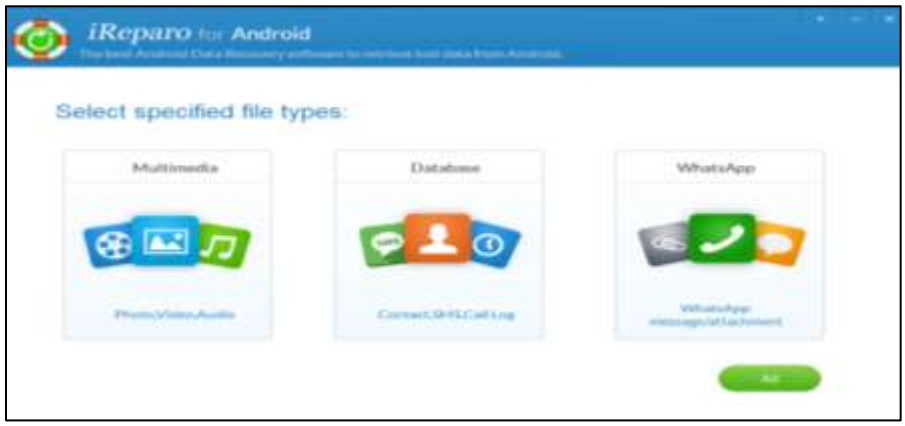


Figure 56: Selection of specific files.

Figure 57: File types to recover.

- Connect the Moto G device to the laptop with the supported USB cable and check whether the connection is valid.

  Note: the device must be in debugging mode in order to perform the recovery option.

Figure 58: Connecting the device.

Figure 59: Ready to scan.

- Start extracting the device and wait for the process to be completed.

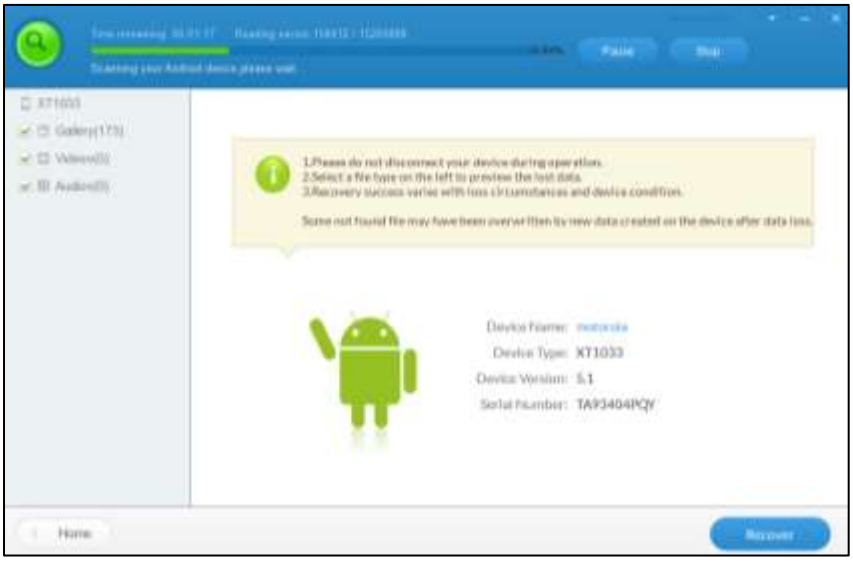


Figure 60: Scanning the data on the android.

Data collection through the cellebrite touch forensic tool.

- Start the Cellebrite tool and open the cellebrite touch application as shown below.

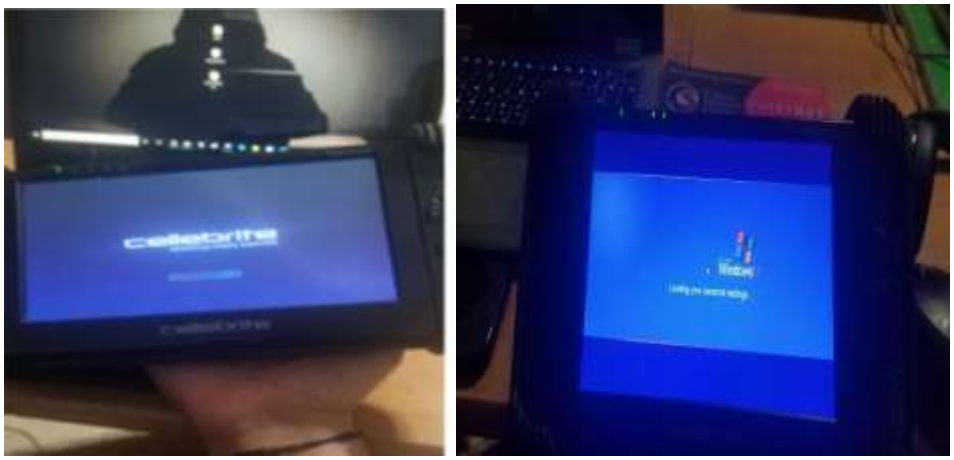


Figure 61: Cellibrate touch forensic tool.

- Click start menu and open the Cellebrite touch application:

Figure 62: Desktop view of the tool.

Figure 63: Launching the touch application.

- Connect the mobile to the Cellibrate touch forensic tool using the USB cable and keep the phone in recovery mode as discussed in the earlier topics.

Figure 64: Connecting device to Cellibrate touch.

- Choose the backup option and select the device name from the given list as shown below.

Figure 65: Choose the mobile vendor and model.

- Connect the USB device that we use as the output device from the right USB port given on the tool and click next to start scanning the device. Make sure that the device is set to debugging mode and we have access for reading the MTP connection is on.

- We could see the pop-up message on the tool saying to check for the MTP connection and then after accepting the alert we could see the connected device details along with the instructions to start the scan as shown below.

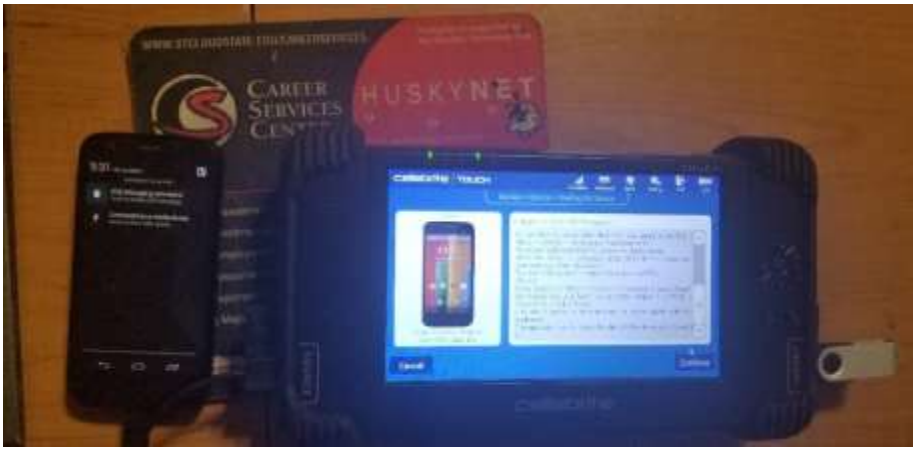


Figure 66: Mobile description and instruction on tool.

- Now read all the instruction and continue to scan the device connected.

Figure 67: Start the scan.

Here the Cellibrate tool will start the process by scanning the device including the information about the contacts, audio, pictures, ringtones, SMS, videos, etc.

Figure 68: Extracting the data from the phone.

- It would take a while to scan and extract the data from the mobile depending on the device. the data on the mobile is then copied to the pen drive and backup file along the report file is created.

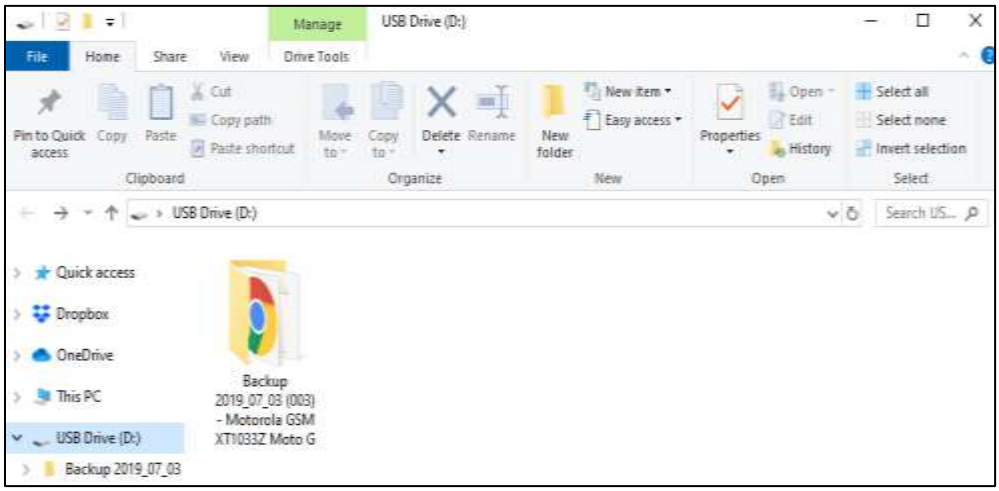


Figure 69: Backup of extracted data from Cellibrate tool.

All the extracted data from the android device (moto g) is discussed further in the data analysis section.

Installation of iTunes for iPhone 6s.

- Download the iTunes package using the following link https://www.microsoft.com/en-us/p/itunes/9pb2mz1zmb1s?cid=appledotcom&rtc=1&activetab=pivot:overviewtab

Figure 70: Downloading the iTunes package.

- Connect the phone using USB cable and check whether the iTunes recognizes the phone.

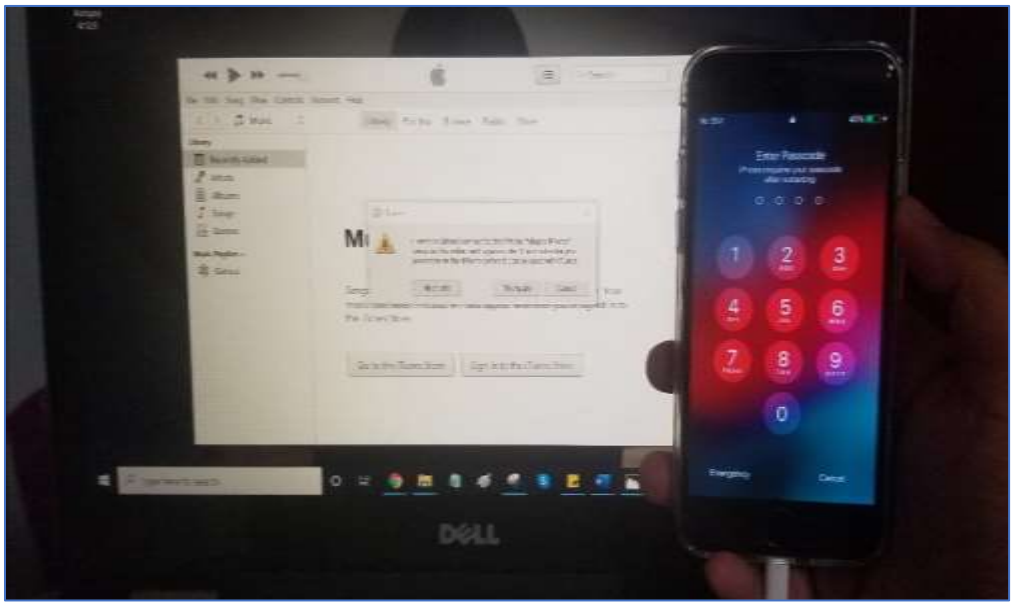


Figure 71: Connect the device to iTunes application in laptop.

The above picture shows that iTunes recognizes the phone but it is not connected to phone since the phone is locked with passcode. In order to use iTunes we need to unlock the phone. We need to set the iPhone to recovery mode and dfu mode in order to bypass the password and then connect to the tool to perform recovery.

Setting up the iPhone to recovery mode.

- Switch off the phone by pressing the home button for some time and the screen would show the power icon to drag right. Then hold the home button for few seconds and attach the USB cable to device while holding the home button until the itunes icon appears on the screen.

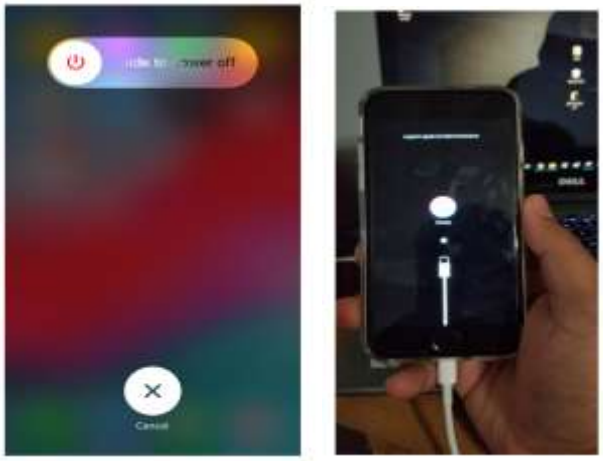


Figure 72: Setting the iPhone to recovery mode.

Now we open the iTunes in laptop and we could see that the iPhone is set to recovery mode as shown in the below picture.

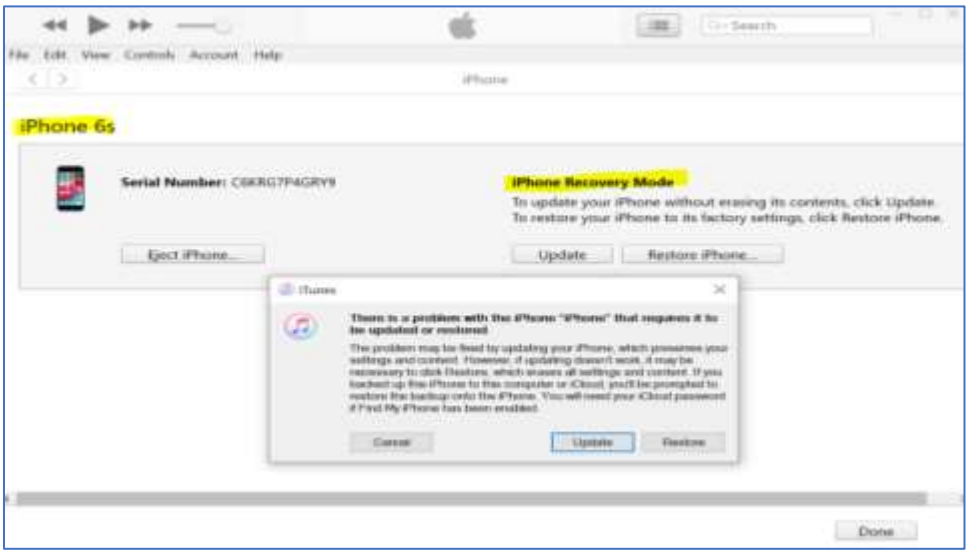


Figure 73: iTunes description about iPhone recovery mode.

Setting up the iPhone in DFU mode. In order to perform recovery without knowing the password to unlock the iPhone must be set to DFU mode. Steps to set the iPhone 6s to DFU mode are discussed below.

- Connect the phone to laptop with a supported USB cable. Open iTunes applications in laptop
- Keep holding the power key and home key at the same time for 10 seconds and release the power button.
- Hold the home button for few seconds until the phone shows a black screen.
- Now the phone is set to DFU mode though it looks like it got shut down.
- we could see the iTunes shows the pop-up message saying that the device is in recovery mode (Chan, 2019).

Figure 74: Setting the iPhone in DFU mode.

Backup using iTunes recovery mode.

Now with the phone setup in DFU mode, we could take a backup of the phone using the iTunes as shown below.

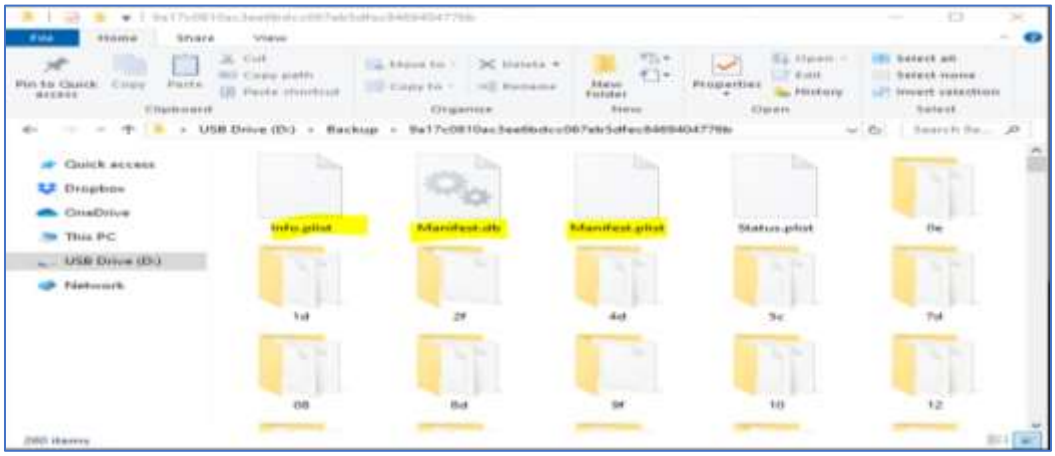


Figure 75: Device backup through iTunes when set to recovery mode.

Now by using the iTunes recovery methods and the following installations of tools, we should be able to recover the data from the phone as it is set to DFU mode. There are several

tools that could be useful in recovering the data and some of them include Imyfone D-Back for iPhone recovery, Gihosoft recovery tool, Cellebrite touch forensic tool, etc.

Installation of Imyfone D-back for iPhone recovery.

- We use the following link to download the package of imyfone data recovery.

  https://www.imyfone.com/iphone-data-recovery/

Figure 76: iMyFone D-back download webpage.

- Run the downloaded package and follow the procedure given in the user guide. Select the license agreement before starting the installation process.

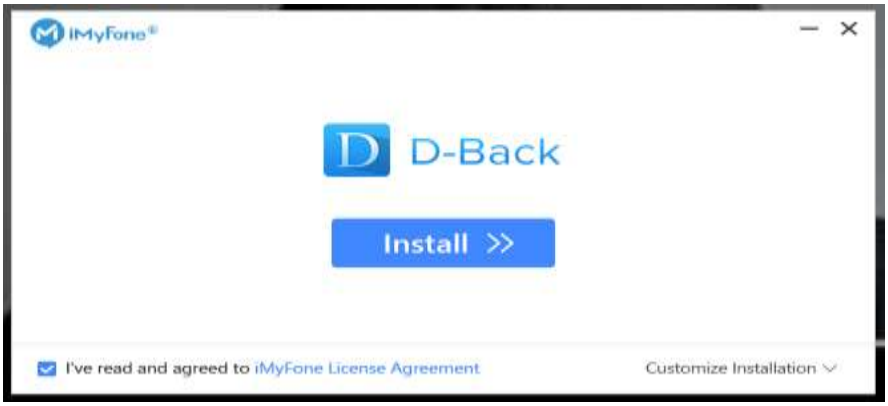


Figure 77: Installation of D-back tool.

- The tool would give us an alert popped out while installing the package regarding the user privileges,Allow user control permission by clicking yes option and proceed to the next step.
- The installation would take a while to complete the process.

Figure 78: Installing the package.

- There are different ways to recover the data, the tools give us a variety of options to recover the data from the device iPhone 6s.

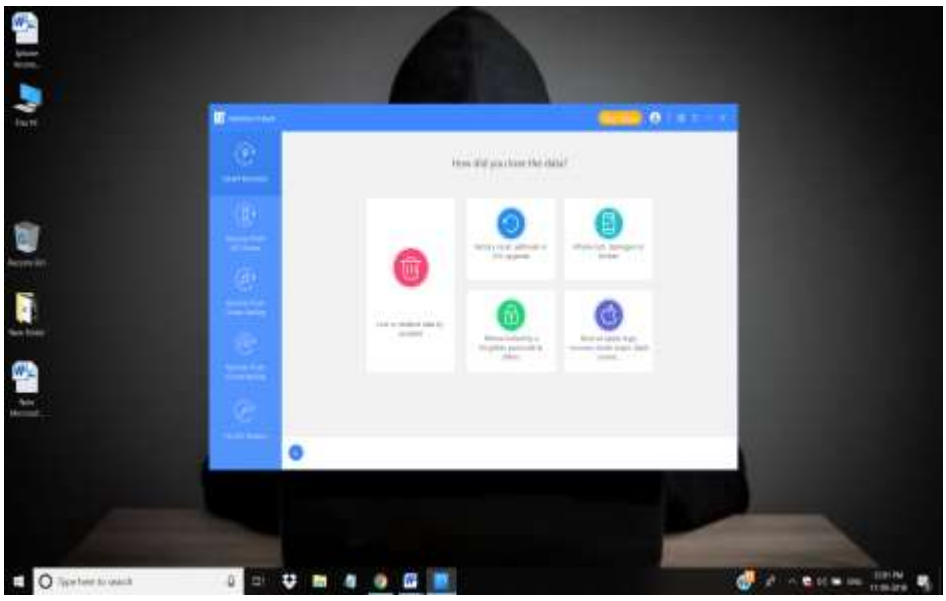


Figure 79: Choose the data type for recovery.

- Start the application and select the recover from the iOS device from the given left panel option.

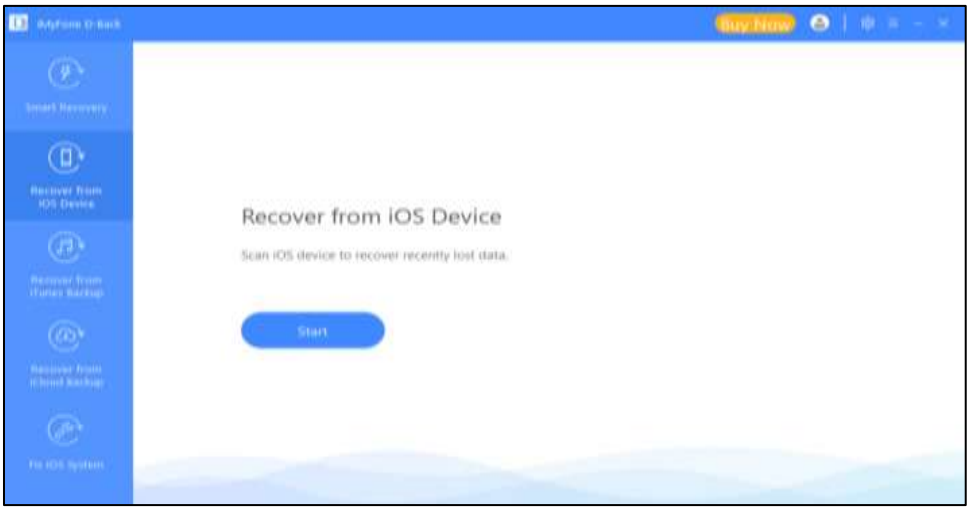


Figure 80: Recover from iOS device option.

- Now we connect the iPhone 6s to the laptop using supporting USB cable and continue through the process.

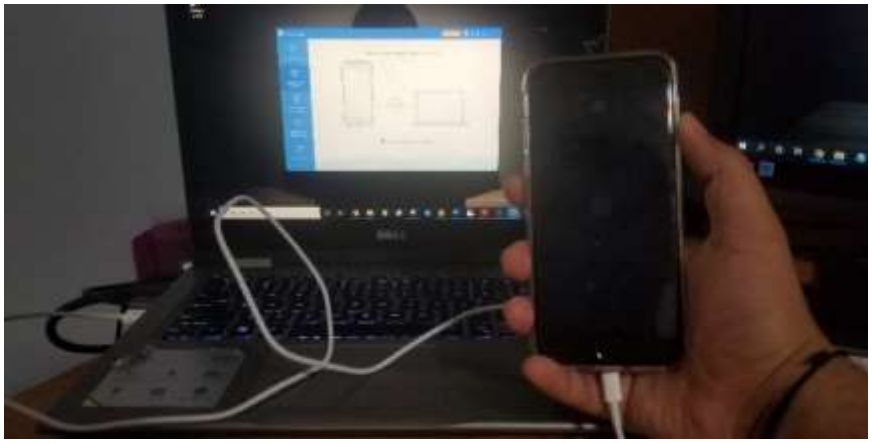


Figure 81: Connect the device to the application.

- We should see the connected phone showing in iMyfone D-back as shown below.

Figure 82: Device detection on the application.

- Now select all the data types we want to recover and click the scan option and continue to further steps**.** The scanning time may vary on the size of the data present on the iPhone and it would take a while to completely scan the data as shown in the figure below.

Figure 83: Scanning the data on the device using the application.

Installation of Gihosoft iPhone Data recovery.

- Download the package from this website https://www.gihosoft.com/iphone-data-recovery-free.html.

- Run the setup file to install the Gihosoft free iPhone recovery tool. Accept the license agreement and click next.

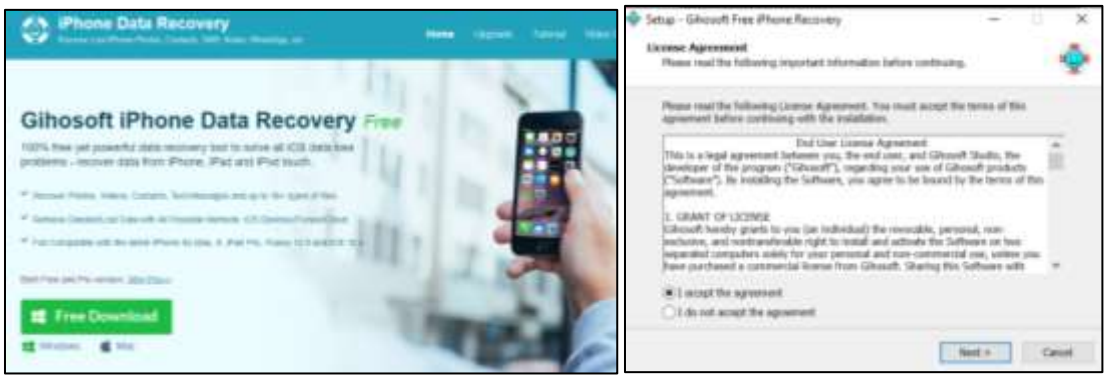


Figure 84: Installation and user license agreement of Gihosoft recovery tool.

- Now browse the destination location that we want to recover the data and continue clicking the next option through the process.
- The process will continue for a while and as it follows please continue to set the iPhone to DFU mode or recovery mode as described in earlier sections.

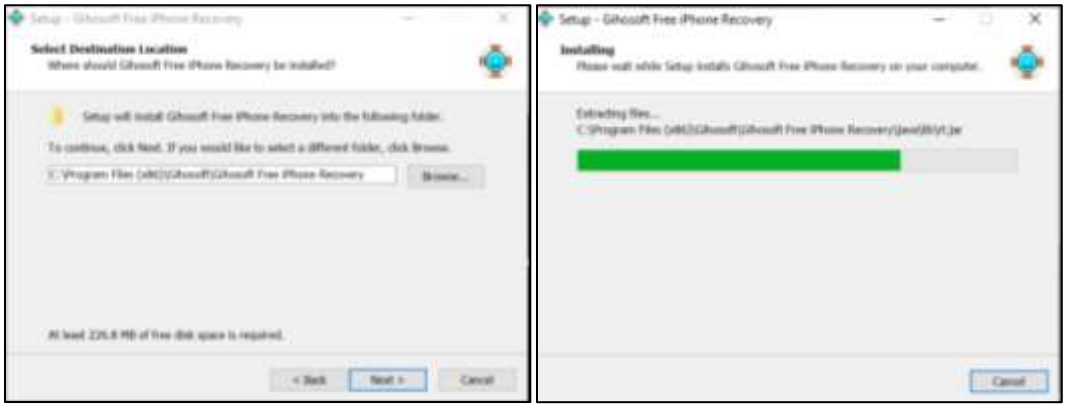


Figure 85: Selection of the installation package location and installing the package.

- Open the application and connect the phone to laptop with USB cable, so that we could connect the application to the device.
- Select all the file types we want to recover the data and,

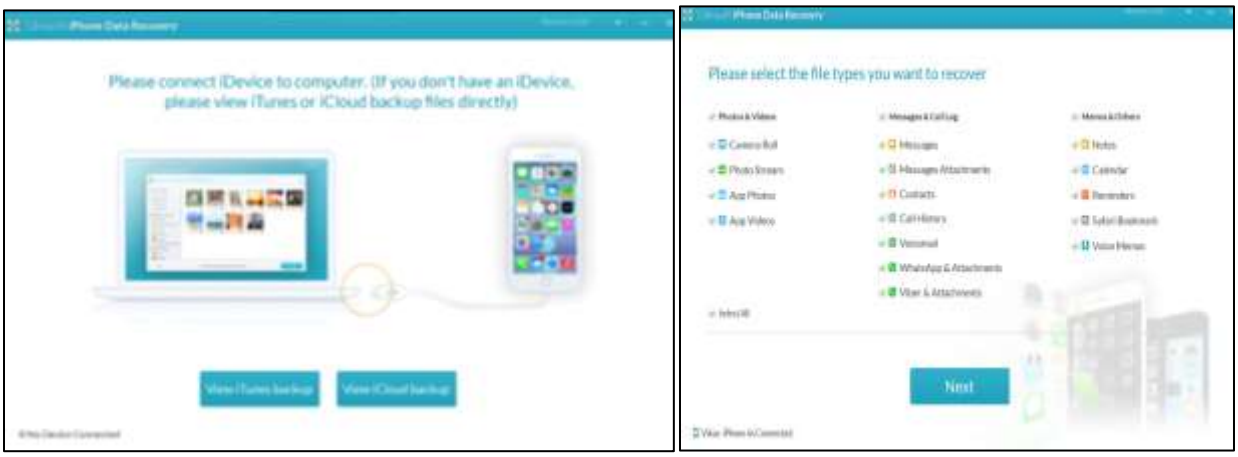


Figure 86: Connecting the device and select the data types.

- It would take a while to scan the complete data and recover them. The below figure shows the process of recovering of iPhone 6s connected.

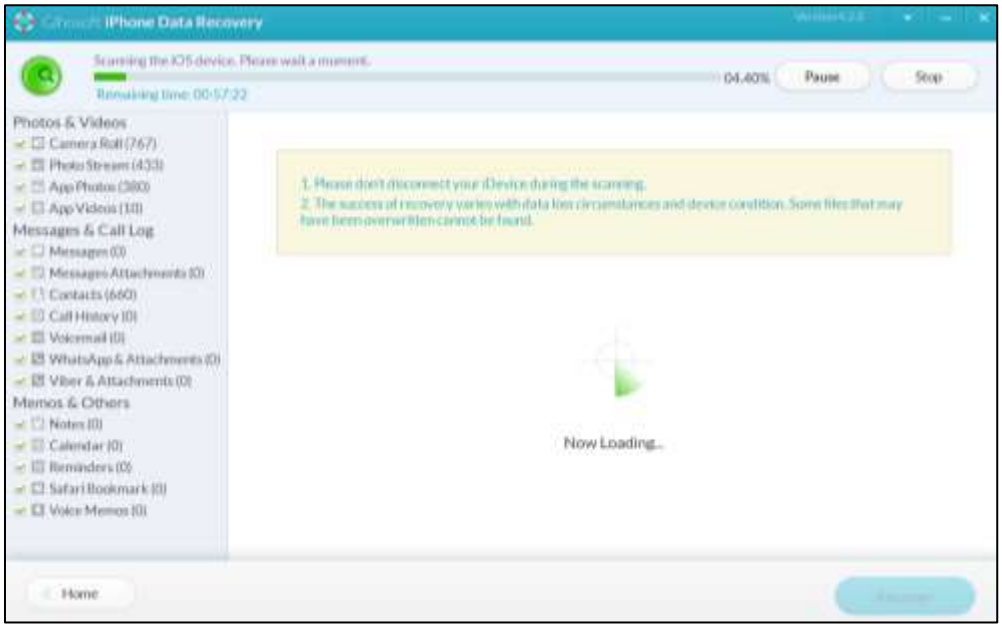


Figure 87: Gihosoft scanning data from the iOS device.

Data collection through the cellebrite touch forensic tool.

- Connect the iPhone 6s to the forensic tool as shown in the picture:

Figure 88: Connect iPhone 6s to Cellibrate tool.

- Click on the vender option and choose the iPhone brand from the list and click to continue
- Connect the output device when it prompts on lower right side of the screen. There is a slot for connecting SD card on the top left side of the device. In our case we use Pendrive as the output device as shown in the picture below.

Figure 89: Choose iPhone 6s from the vender Apple.

- Now the tools pops up for the warning message and sends the information regarding the alert to the device.
- Click on the trust option on the device and continue the process to recover the data.

Figure 90: Click on the trust option.

- The Tool will show up the information about the device and instructions to proceed before the recovery.
- We select the all the types of data that we need to recover like phonebook, Audio, SMS, videos, pictures, ringtones etc. on the given options and click start to scan and extract the data from iPhone 6s.

Figure 91: Select all types of data.

Data Analysis

Here we will discuss about the recovered data collected from Android device Moto G and iPhone 6s from using various methods. We also analyze the data recovered and the challenges faced while retrieving the data for both the devices.

Data recovered from an Android device(Moto G). By utilizing the andriller application we have extracted the data and took a backup file from the device. we have created the case in the autopsy tool and added the backup to the case as shown in the Previous sections at the data presentation part.

The metadata about the device details including serial number, IMEA number, user information and the tool used to extract data were found using the autopsy details as shown below.

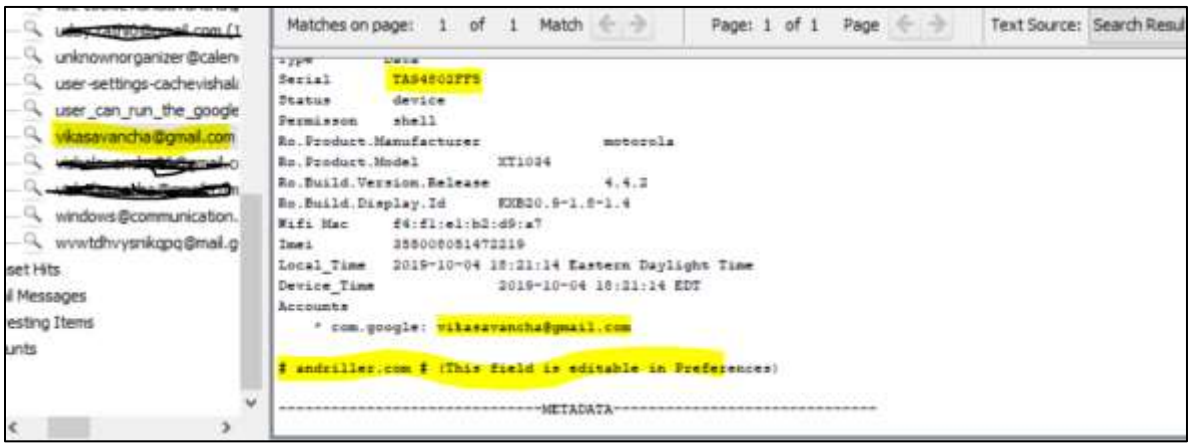


Figure 92: Device details seen in autopsy extracted from andriller.

From the above picture we could confirm that the connection from the device to the tool was successful and extracted data was properly recovered from the android device used.

- The data we retrieved from the android mobile through autopsy is shown below.

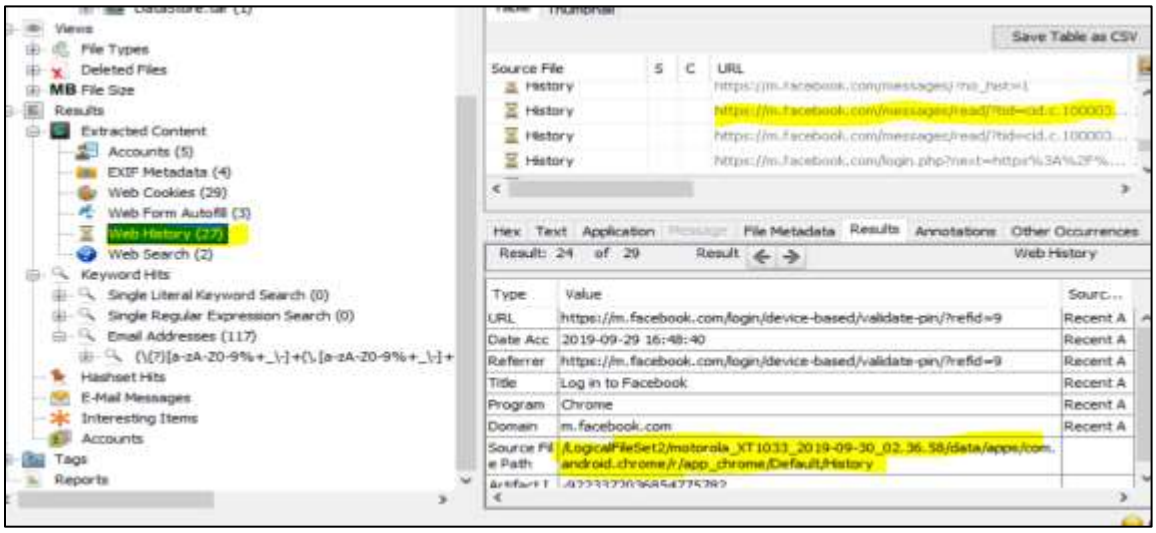


Figure 93: Web history recovered from android visualized in autopsy.

There are various files we could extract from the device such as user account details, metadata , web history, Wi-Fi passwords etc. From the above picture we could see the recovered details about the web history that were searched on the moto g and all the details about the link, log in details, the file path for the source, artifacts were found.

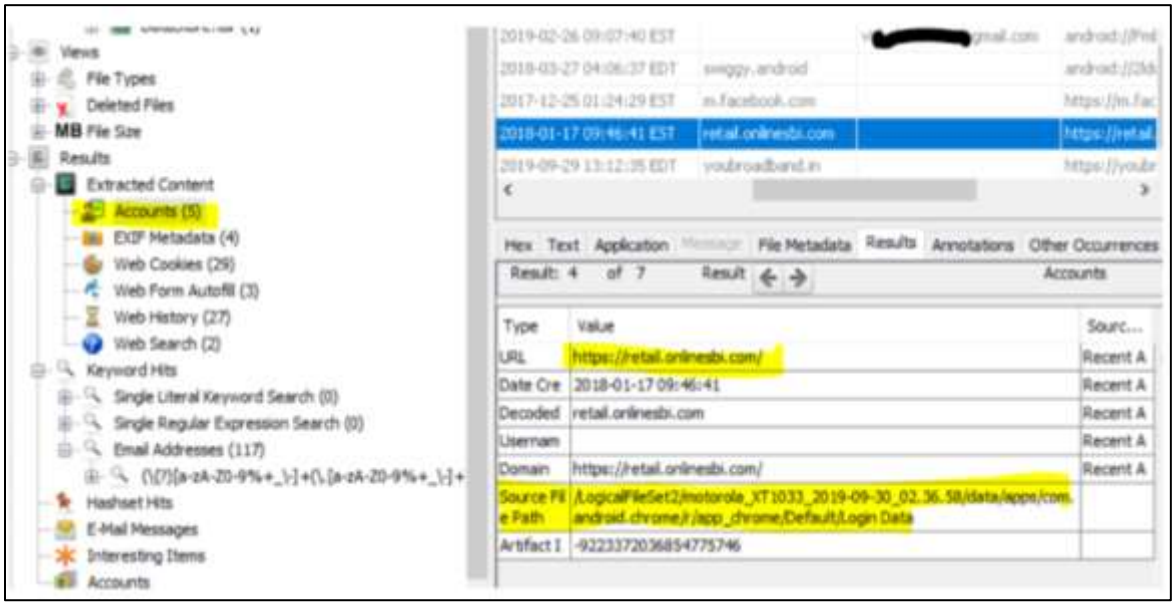


Figure 94: Bank login details recovered from the device.

- From the above picture we could see the account details of the sensitive information about the bank websites were found. All the details about login password used and transaction details were found. By using this information we could analyze the user may have used this bank details for transactions before making the crime.
- We see that there were 117 email addresses found in the email hits and all the information about the emails were captured in the metadata.
- We could also verify the details of the transaction that happen and confirm whether the transaction was made purposefully as a contract benefits for accomplishing crime.
- From the below evidence picture, we see that the information about the Wi-Fi details along with the metadata about the network details and location of IP addresses used are recovered.

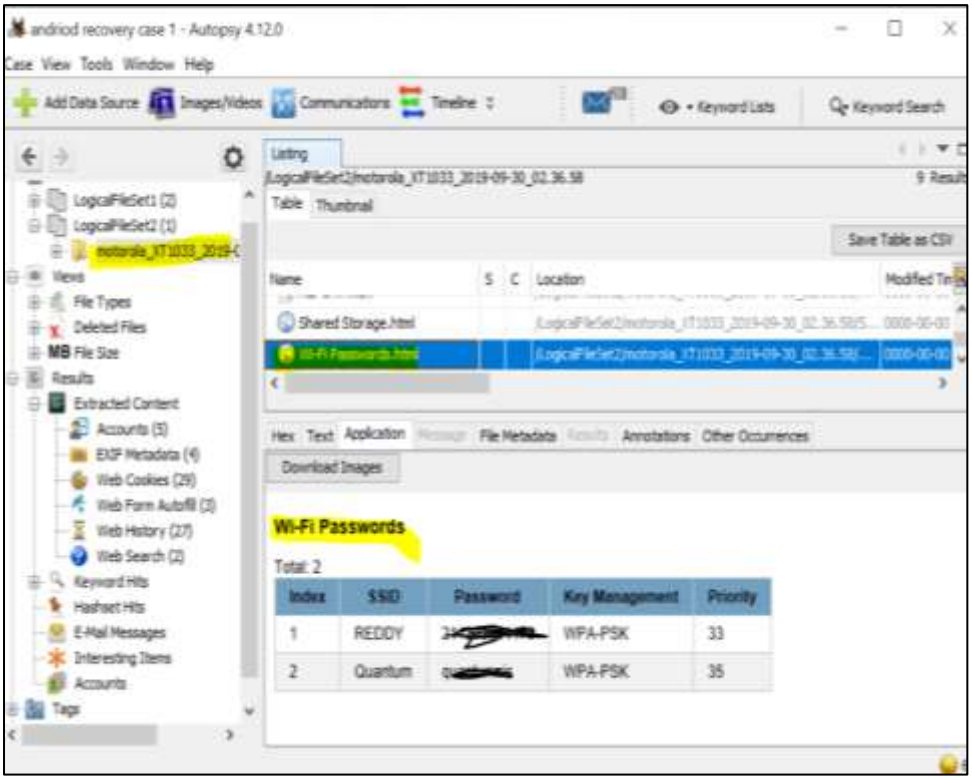


Figure 95: Wi-Fi passwords recovered from the device.

- The source of the file path is also found along with the details, that would confirm the evidence found from the device are authentic and using this details we could write a report supporting all these evidences that the user of the phone must have managed to use all these ways to perform cybercrime.

Data recovered from the Cellebrite tool is forwarded to the Pendrive. We connect the pen drive to laptop and we could see the recovered data from the below.

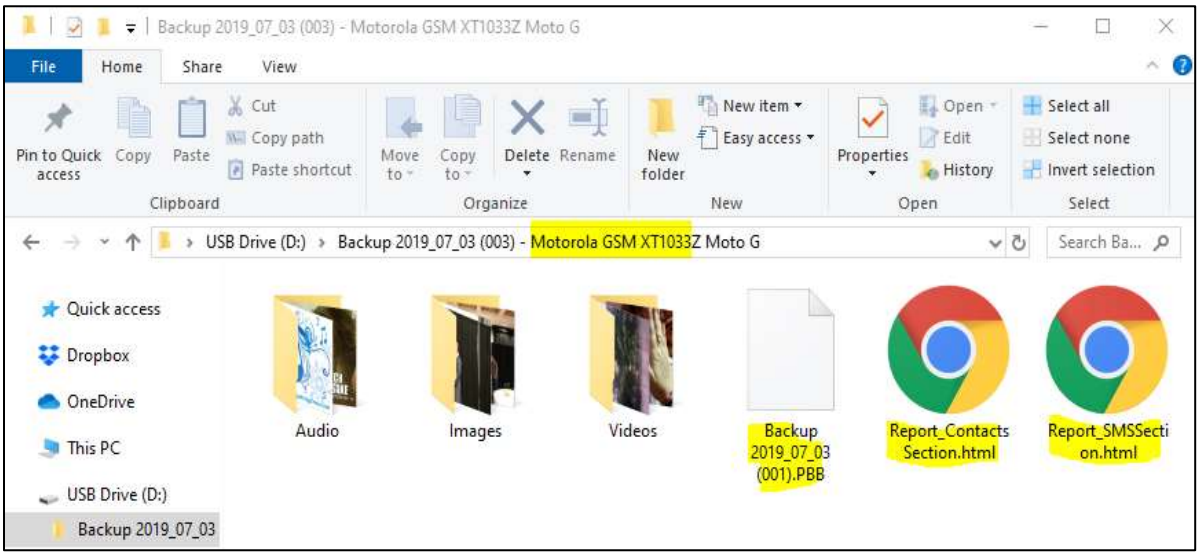


Figure 96: Backup from the device using Cellibrate tool.

From the above evidence screenshot we could see that the backup file has the details about all the audio files, images and videos. Now opening the weblinks created by cellebrite tool with reports of contacts and SMS details are shown below.

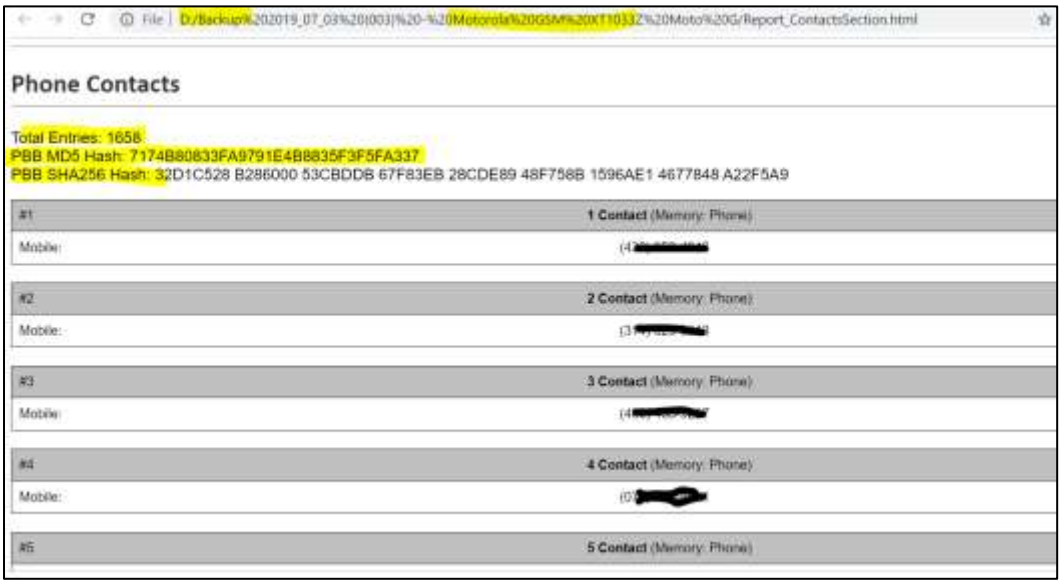


Figure 97: Report of contacts section recovered using Cellibrate tool.

The above web link gives us the contact details from the moto g that have been saved. Using the autopsy, we analyze the backup file created by Cellebrite tool, we have found the information about the contact details and email address of the contacts as shown in the below figure.

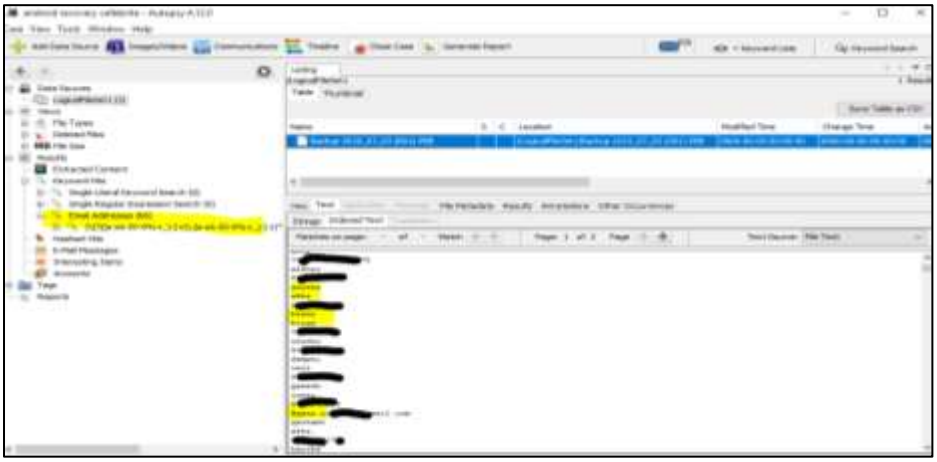


Figure 98: Email addresses recovered using Cellibrate tool.

Now from the recovery tool I-reparo we could see the details for images, pictures and contact information were found.

Figure 99: Deleted pictures recovered using ireparo.

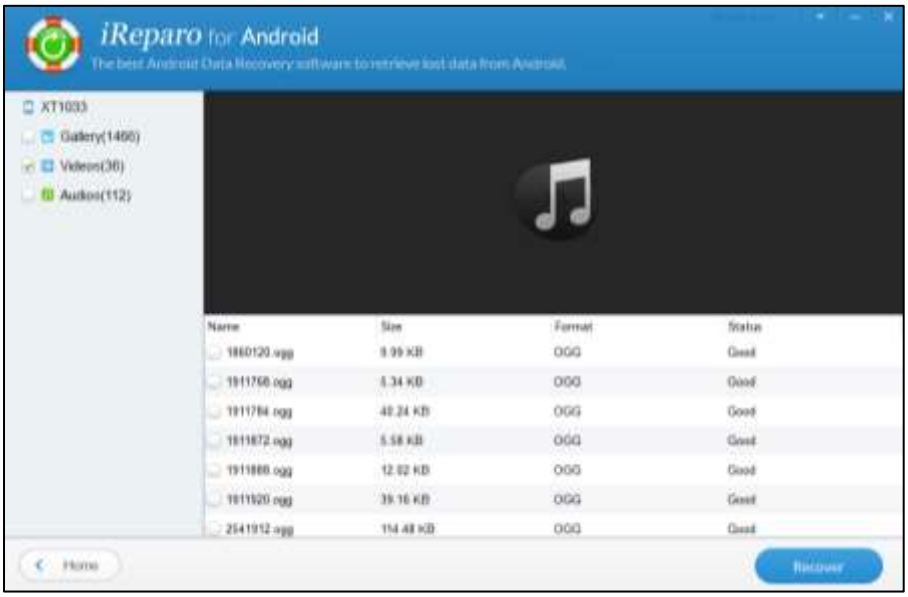


Figure 100: Recovered videos and audio files recovered using ireparo.

The above picture could give us the information about the details of all the deleted images and videos that were recovered. From all the above Evidence recovered from android mobile logically with the different tools, we confirm the logical acquisition of android mobile moto g was successful.

Data recovered from iPhone 6s. Here in this section we will see all the data that is recovered from the tools we used and the challenges that we found while extracting the data using the Calibrate touch too are discussed below.

From the iTunes backup, using autopsy tool, following data is recovered from the iPhone.

Figure 101: Analyzing files from iTunes backup in autopsy.

We see that the autopsy is analyzing the logical files and keyword searches such as email id, contacts, etc.

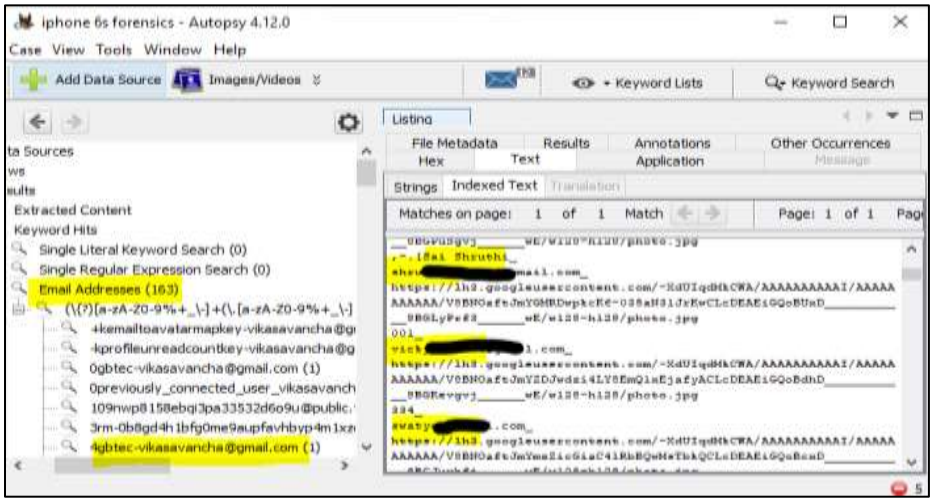


Figure 102: Email addresses and content recovered using iTunes backup.

From the above screenshot, we could see all the email addresses along with the content is recovered, with this data we could analyze and see what are all the communication that user made and we could figure out a lead from there.

With the help of plist editor pro 2.5.0 we could open the manifest , status plist files and get all the details about the phone details, Apple id of the user, to confirm that the mobile belongs to the user.

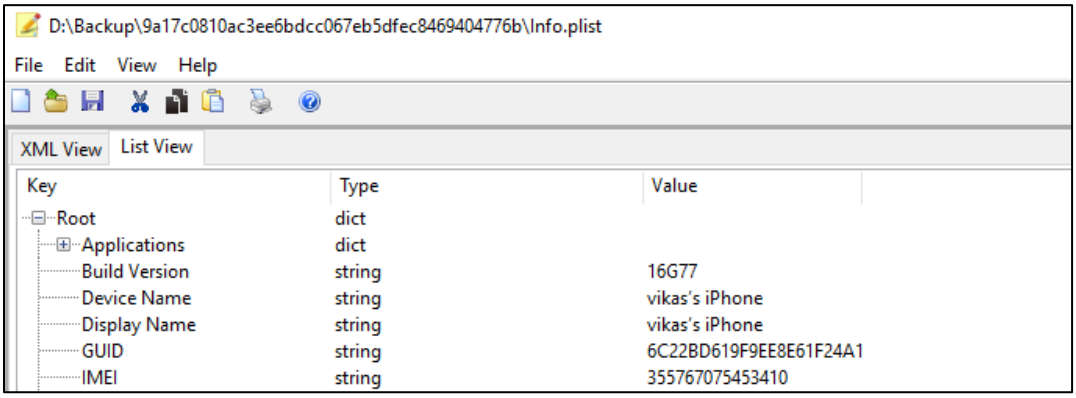


Figure 103: iPhone details recovered using plist editor pro.

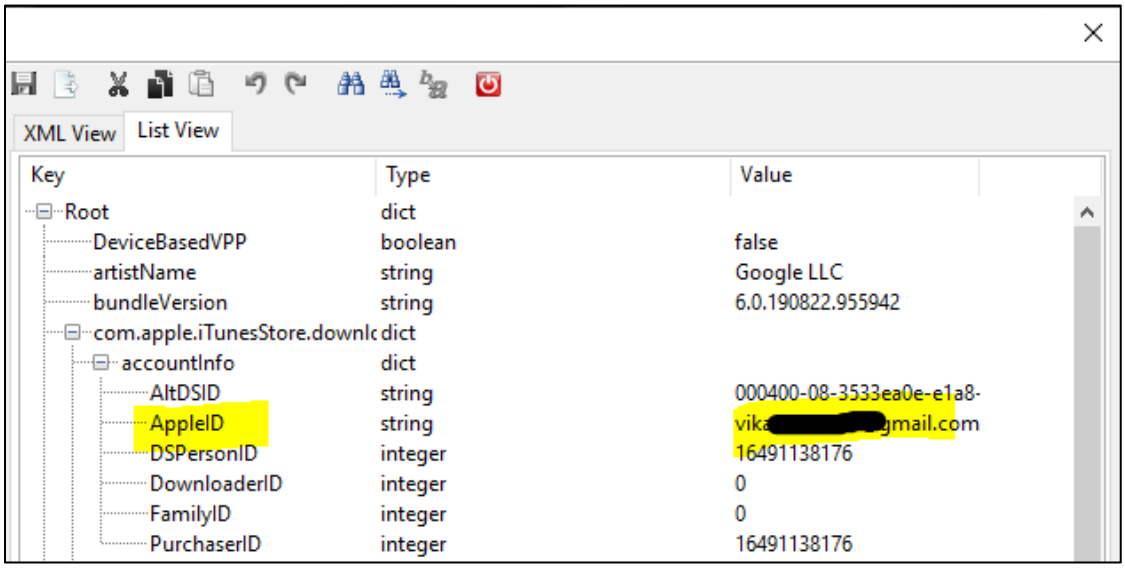


Figure 104: Apple id recovered through plist editor pro list view.

When we further search for the lockdown and deleted application strings under the list view we could figure out the metadata about all the details of contacts, Calenders, bookmarks as shown in the below screenshot.

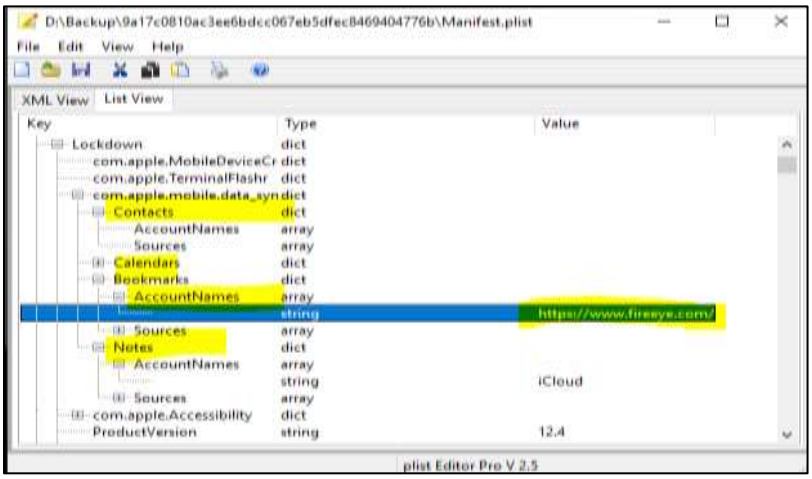


Figure 105: Bookmarks of websites recovered using plist pro.

We see that all the information about the bookmarks, contacts and notes are visible in plist editor, if we could use the links in bookmarks and further analyze what user is trying to acquire knowledge from these websites. We also could see the maps that might lead us to stronger evidence that summarizes the complete plan of the crime.

The data recovered using the application Gihosoft iPhone data recovery are shown below.

Figure 106: Data recovered from iPhone using Gihosoft.

We could see all the details of data recovery from the above picture and the connection to the tool and the device is successful.

Note: The results of the recovered data are possible only when the iPhone is set to recovery and dfu mode or unlocked mode with legal proceedings.

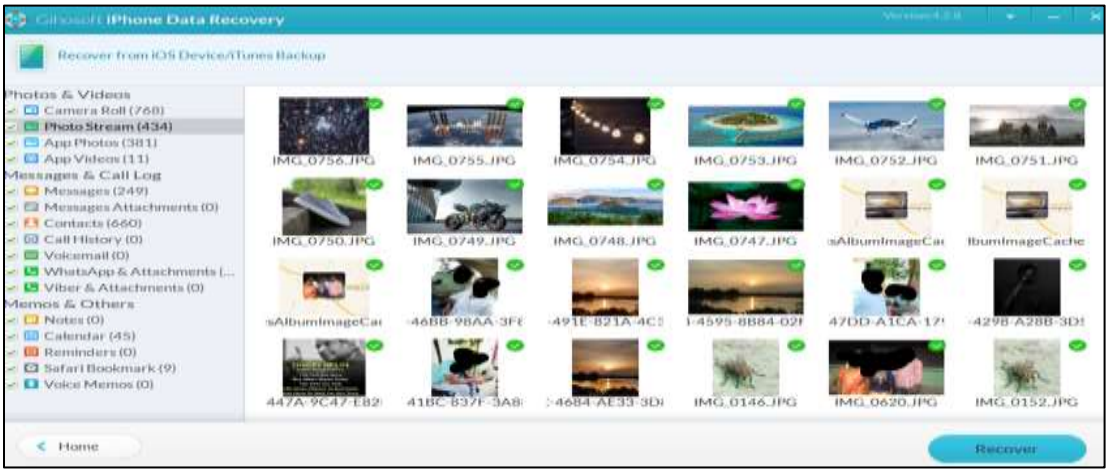


Figure 107: Deleted images recovered from the iPhone using Gihosoft.

From the above screenshot, we could see all the deleted data regarding the images were found, from we could analyze that the user of the phone may have deleted the files regarding images that he took as supporting files to perform the crime scene.

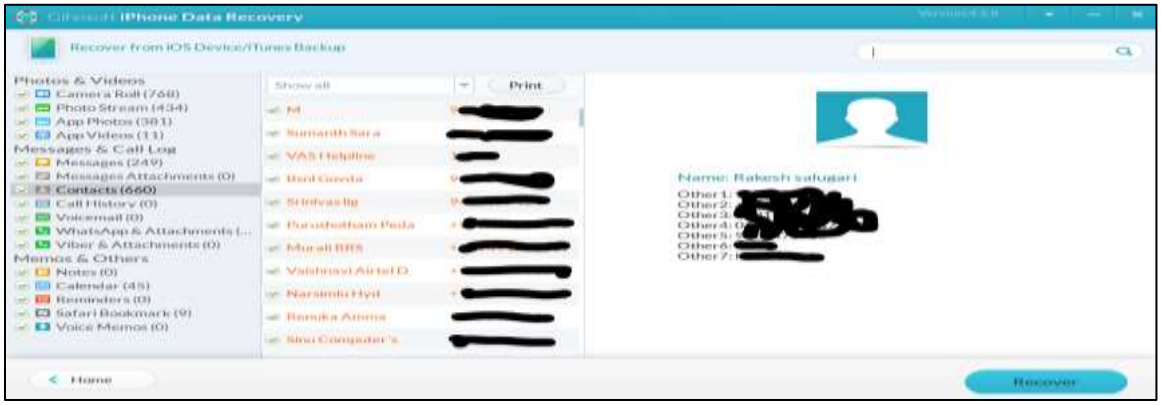


Figure 108: Contacts recovered from iPhone 6s using Gihosoft.

From the above screenshot we could get the recovered details about the contacts that the user has saved on his iPhone.

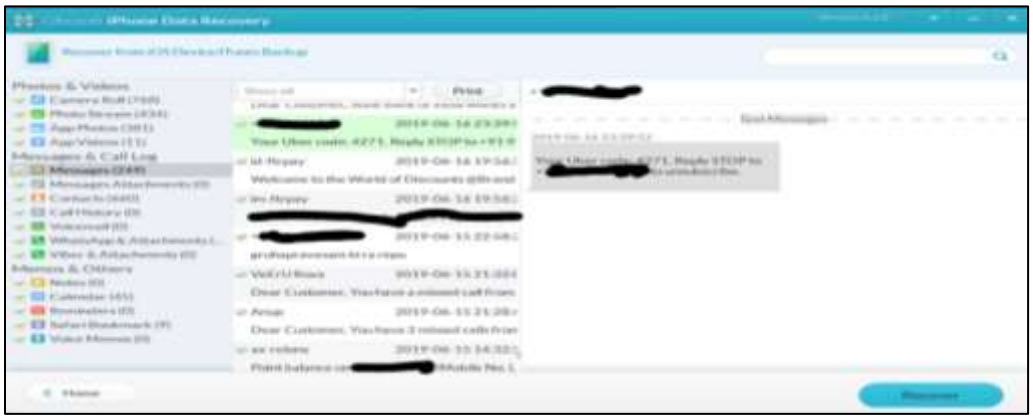


Figure 109: Messages recovered from iPhone 6s using Gihosoft.

We could see all the message details recovered from the iPhone 6s that user had conversation and also we could see the uber details that he traveled from place to place. This would be one of the best evidence for the forensic team to submit to FBI to proceed to further legal inquiries such as calling the uber number that was assigned to the phone and gathering the information about where the user was traveling during the specific period of time.

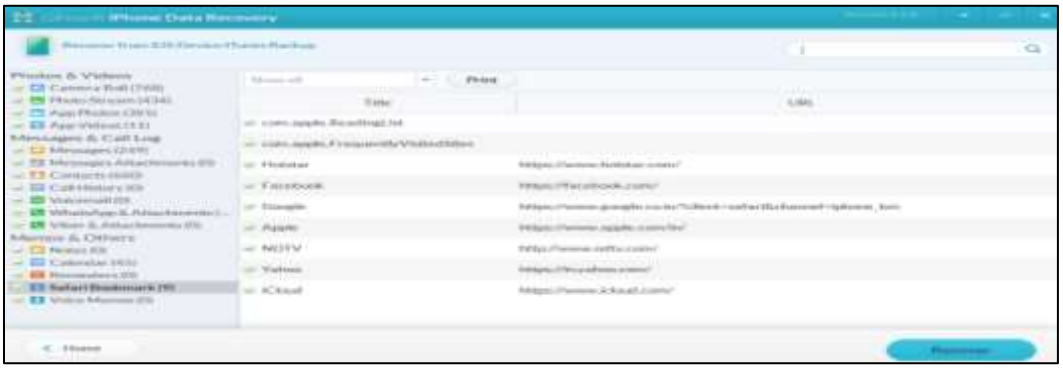


Figure 110: Safari bookmarks recovered from iPhone using Gihosoft.

From the screenshot, we could see the information regarding safari bookmarks, we could see that all the information users had constantly reached out to websites.We could recover only some part of the deleted data that was available on the device by using the above tool But as a forensic team, we need to recover deleted data efficiently from the phone that would be even

more helpful to analyze the data and produce the evidence to the court. For this purpose, we use another application iMyFone D-back. With the help of this tool we recovered the deleted data from the iPhone 6s.

The data recovered from iMyFone D-back are shown below.

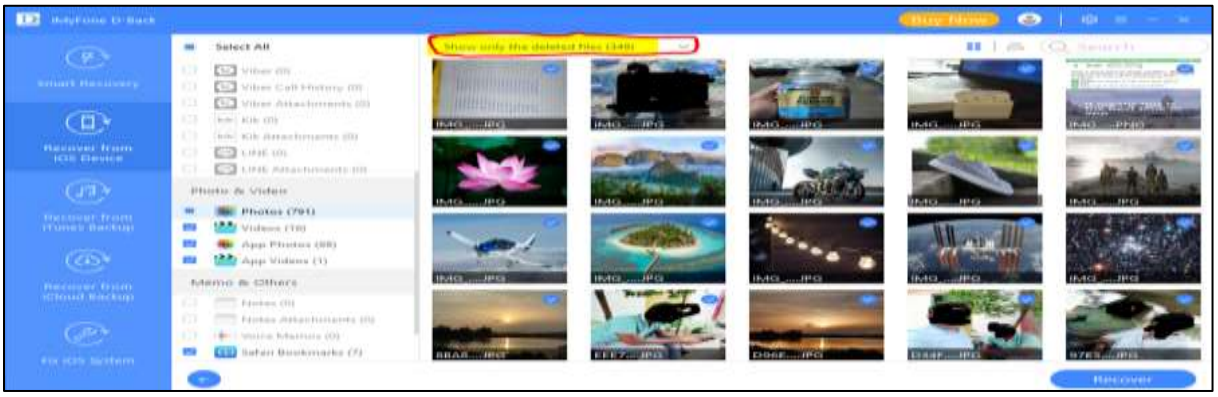


Figure 111: Deleted images recovered from the iPhone using iMyFone.

Here we could see all the deleted data of images from the device and the messages from external applications such as WhatsApp are shown below.

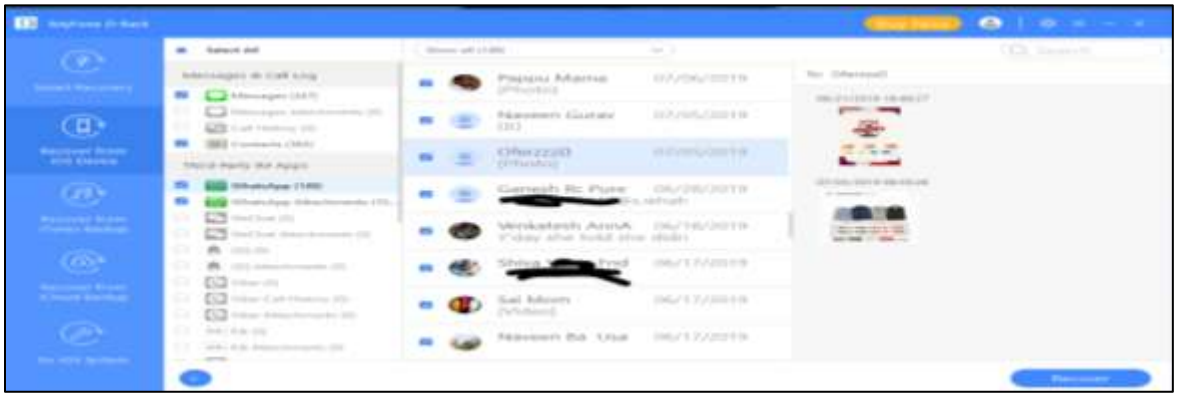


Figure 112: Whatsapp data recovered from the iPhone using iMyFone.

Data recovered from Cellebrite forensic tool is shown in the below picture:

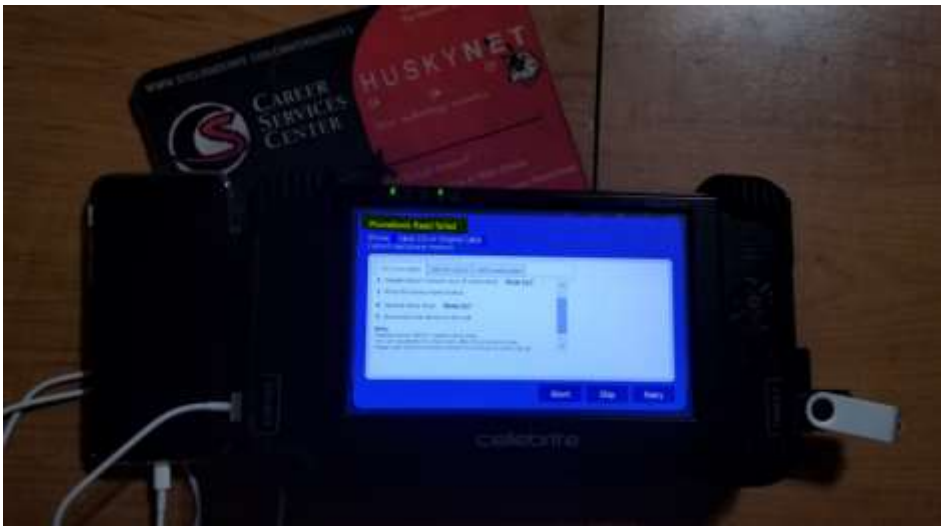


Figure 113: Details about the recovered data from the iPhone using Cellibrate tool.

We could not extract the results from the iPhone using the Cellebrite tool as we see that the device should be in Debugging mode in order to extract the data. We are successful in setting up the device recovery mode and DFU mode that solved our recovery collections to some extent but we could not break the Passcode or pattern on iPhone 6s in our experimental analysis, thus we could say that we are partially successful in recovering the data using logical acquisition.

Limitations

This project shows the experimental analysis of comparison and analysis of logical acquisition of the iPhone over an Android device with existing methods and will not prove any new methods achieved to recover the data. The experimental results and analysis are limited to above two phones (i.e., iPhone 6s, Moto g first gen) and the results may vary with different iPhones and android mobiles with various reasons depending on their Operating system versions, security design and the storage capabilities about how much data is deleted and set to factory reset mode.

Chapter V: Results, Conclusion and Recommendations

Introduction

In this chapter we will compare the achieved results on both the android and iPhone devices and conclude which one we could get complete success by using various forensic tools and techniques. We will also discuss the challenges that we would face with the updated version of operating system in both the android and iPhone forensics.

Results

For this project the devices utilized for forensic acquisition are iPhone 6s with iOS 12.4 and Moto G with android version 4.4.2. Successfully performed the debugging techniques, like using the PowerShell command line to enable MTP transfer functionality, we are successful in enabling the MTP along with ADB activation. With the help of the Andriller tool, we acquired the data including the emails, contact history, Wi-Fi passwords and metadata about the media files including videos, pictures that are deleted from mobile.

With the help of other tools like ireparo forensic recovery for android, Cellibrite touches forensic tool kit we successfully recovered the deleted data about messages, contacts, call history. Whereas we are for the iPhone 6s that we performed forensic analysis and recovery, we were partially succeeded in recovering the data using various acquisition techniques.

For the iPhone debugging, we used various tools but could not succeed in cracking the passcode, we did set the iPhone 6s to recovery mode and dfu mode, then with the help of forensic tools like Gihosoft iPhone recovery tool, iMyFone D-back and Cellibrate forensic tool kit we are successful in recovering the deleted data with logical acquisition methods. we are partially successful in recovering the data from the Cellibrate forensic tool.

Conclusions

With the help of several forensic tools utilized in the research we successfully recovered the data from both the devices. As the operating systems and security architecture design is unique and completely different in both the devices, the methods we utilized for the debugging the android device and collecting the recovered data has different approach when compared to iPhone forensic methods. All the techniques that we performed do have several challenges and flaws that made little variations in the data recovery results when compared with both the devices. we are partially successful in bypassing the iPhone as we could not crack the passcode or pattern by using different forensic tools but recovered most of the deleted data when the iPhone is set to recovery mode and then to DFU mode and back up the data using the iTunes and with the help of all the forensic tools like Gihosoft, iMyFone, Cellibrate tool kit. From all the research and experimental analysis performed in this paper, we conclude that the in-depth analysis using the logical recovery methods in android can be achieved when the device is set to debugging mode with the specific os version whereas for the iPhone we are partially successful in debugging the phone by setting the phone to recovery mode and DFU mode and taking a backup using iTunes and In-depth analysis of recovered data with logical acquisition techniques can be achieved if the device unlocked or forensically open by the forensic team with legal warrant or proceedings.

Future Work

Constantly upgrading the OS versions and enhancing the security architecture designs by adding the security patches and features including fingerprint sensors and face id to protect the device from compromising the data made a more interesting and challenging environment for the

forensic team to acquire data. With further research and analysis, we may be able to bypass the updated security features by using various forensic methods and tools.

References

ADC. (2000). *HFS plus volume format.* Retrieved from Technical Note TN1150: https://www.fenestrated.net/mirrors/Apple%20Technotes%20(As%20of%202002)/tn/images/tn1150_001.gif

Adorama. (2018). *EDEC digital forensics black hole faraday bag standard size, non-window.* Retrieved from ADORAMA: https://www.adorama.com/images/Large/edbhfb2.jpg

Alison. (2019). *Architecture overview*. Retrieved from Alison.com: Architecture Overview - Alison. https://alison.com/topic/learn/34794/architecture-overview

Andriller. (n.d.). *Andriller - android forensic tools*. Retrieved from Andriller: https://www.andriller.com

Axon, S. (2017). *The curious case of the time-traveling phone.* Retrieved from Ars Technica: https://arstechnica.com/gadgets/2017/09/iphone-8-and-8-plus-review-the-curious-case-of-the-time-traveling-phone/6/

Benny. (2014). *Benny's hub about cyber security.* Retrieved from bennysecurity: http://bennysecurity.blogspot.com/2014/02/infosecinstitute-android-architecture.html

Carrier, B. (2019). *Analysis features.* Retrieved from sleuthkit.org: https://www.sleuthkit.org/autopsy/features.php

Cellebrite. (2018). *UFED ultimate*. Retrieved from Cellebrite: https://www.cellebrite.com/en/products/ufed-ultimate/

Chan, R. R. (2019). *How to put your iPhone or iPad into DFU mode*. Retrieved from imore.com: https://www.imore.com/how-to-iphone-ipad-dfu-mode#ipad

Dhinakaran Pandiyan, S. P. (n.d.). *Android architecture and binder*. Retrieved from /project_final/CSE_598_Android_Architecture_Binder: http://rts.lab.asu.edu/web_438/project_final/CSE_598_Android_Architecture_Binder.pdf

Dimitar. (2018). *The mobile forensics process: Steps & types*. Retrieved from infosec institute: http://2we26u4fam7n16rz3a44uhbe1bq2.wpengine.netdna-cdn.com/wp-content/uploads/11-13.png

Elenkov, N. (2014). *Android's security M*. Retrieved from nostarch.com: https://nostarch.com/download/Android_Security_Internals_ch1.pdf

Firelord. (2011). *Andriod enthusiasts* . Retrieved from android,stackexchange.com: https://android.stackexchange.com/questions/112040/how-to-enable-usb-debugging-in-android-if-forgotten-pattern-for-screen-unlock

Flylib. (2017). *Section 12.3. The structure of an HFS Volume*. Retrieved from Flylib.com: https://flylib.com/books/3/126/1/html/2/images/singhfig12-6.jpg

Gondi, T. (2015). *iOS – architecture*. Retrieved from By a Beginner: https://tilakgondi.wordpress.com/2015/01/14/ios-architecture/

Hammond, G. (2017). *Top 6 forensic iPhone data recovery services.* Retrieved from imyfone.com: https://www.imyfone.com/ios-data-recovery/top-6-forensic-iphone-data-recovery/

ifixit. (2015). *iphone 6s tear down.* Retrieved from ifixit: https://d3nevzfk7ii3be.cloudfront.net/igi/alGVuoVYfFpuqK5o.huge

Inc, A. (2018a). *How to erase your iPhone, iPad, or iPod touch*. Retrieved from Apple.com: https://support.apple.com/en-us/HT201274

Inc, A. (2018b). iOS-security. *iOS- security guide*, 78. Retrieved from ios: https://www.apple.com/business/docs/iOS_Security_Guide.pdf

Infosec. (2019). *Computer forenscis: Chain of custody form.* Retrieved from Infosec: https://resources.infosecinstitute.com/category/computerforensics/introduction/areas-of-study/legal-and-ethical-principles/chain-of-custody-in-computer-forensics/#gref

ireparo. (n.d.). *The best Android data recovery app ireparo for android*. Retrieved from android recovery: https://www.androidrecovery.com/

iris. (2016). *Levels of mobile device forensic acquisition.* Retrieved from Iris Investigations: http://www.irisinvestigations.com/wordpress/wp-content/uploads/2016/12/Toolbox/14-WORKSHEET-FLOW%20CHARTS/Levels%20of%20Mobile%20Device%20Forensic%20Acquisition.pdf

John, G. (n.d.). *Analysis And research of system security based on ANDROI.* Retrieved from a4academics: http://a4academics.com/images/ProjSeminarImages/Android-Platform-Architecture.png

Kostadino, D. (2018). *The mobile forensics process: Steps & types.* Retrieved from Infosec Institute: http://resources.infosecinstitute.com/category/computerforensics/introduction/mobile-forensics/the-mobile-forensics-process-steps-types/#gref

Laboratory, P. (2015). *Introduction to mobile forensics*. Retrieved from eForensics Magazine: https://eforensicsmag.com/introduction-to-mobile-forensics/

Mahadewa, K. (n.d.). *Android security model*. Retrieved from Tech Star: https://techstarspace.wordpress.com/category/android-security-model/

Mahalik, H. (2016). *Practical mobile forensics.* Mumbai: Packt publishing Ltd. Retrieved from http://file.allitebooks.com/20170426/Practical%20Mobile%20Forensics,%20Second%20Edition.pdf

Mona Bader. (2010). iphone 3gs forensics. *Logical analysis using Apple itunes backup,* 15. Retrieved from research gate: https://www.researchgate.net/profile/Neelakant_Varma2/post/Can_someone_provide_research_papers_related_to_this_topic_iPhone_Forensics_using_the_iTunes/attachment/59d61de079197b807797b927/AS%3A273790993403906%401442288267248/download/10.1.1.185.4439.pdf

Morgana. (2009). *iphone evolution chart.* Retrieved from Fansshare: http://2.bp.blogspot.com/-oz1BFn0C6Vc/UyXzwZj_K9I/AAAAAAAACVs/zwBhzlkF16k/s1600/iphone_evolution_chart_v4.jpg

Proffitt, T. (2012). Forensic analysis on iOS devices. *iOS devices,* 23. Retrieved from https://www.researchgate.net/profile/Neelakant_Varma2/post/Can_someone_provide_research_papers_related_to_this_topic_iPhone_Forensics_using_the_iTunes/attachment/59d61de079197b807797b926/AS%3A273790989209601%401442288266845/download/forensic-analysis-ios-

Sachowski, J. (2019). *Implementing digital forensic readyness*. Retrieved from Safari: https://www.oreilly.com/library/view/implementing-digital-forensic/9780128045015/XHTML/B9780128044544150142/B9780128044544150142.xhtml

Sharma, N. (n.d.). *The beginner's guide to Android: Android architecture*. Retrieved from edureka!: https://www.edureka.co/blog/beginners-guide-android-architecture/

Sutton, M. (2011). *Faraday bags help secure seized mobile devices.* Retrieved from ITP.Net: http://www.itp.net/images/content/585942/article/11105-disklabs_article.jpg

Velu, V. K. (2018). *Apple's iOS security model.* Retrieved from Safari: https://www.safaribooksonline.com/library/view/mobile-application-penetration/9781785883378/graphics/B05055_02_28.jpg

Wikipedia. (2018). *iOS*. Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/IOS

Zentek. (2004). *JTAG/Chip off forencis*. Retrieved from Zentek forensic limited: https://www.zentekforensics.co.uk/jtagchip-off-forensics