



UNIVERSITY OF PIRAEUS

Department of Digital Systems

MSc in “Digital Systems Security”



iOS Forensics

Postgraduate student: Nikolaos Anagnostopoulos MTE 1202

Supervisor: Christos Xenakis

Assistant Professor, University of Piraeus

Academic Year: 2013-1014

THIS PAGE WAS INTENTIONALLY LEFT BLANK

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Table of Contents

Acknowledgements.....	1
Abstract.....	1
Introduction.....	2
Digital Forensics	3
The current state.....	4
Smartphones operating system market share	4
Smartphones Security.....	5
Mobile Device Forensics	6
Evidence of great interest.....	6
Live forensics	7
Data storage areas.....	7
Forensics procedure.....	8
Methods of data acquisition	8
History of Apple Inc	10
Evolution of iOS devices	11
iPhone Operating System (iOS).....	12
iOS versions	12
Kernel.....	16
iOS SDK.....	17
iOS components.....	18
Boot process	18
iOS file system.....	19
Partitions.....	21
Databases.....	23
Property Lists	23
Forensics procedure prerequisites.....	24
Existence of deleted data.....	24
iPhone and iOS restrictions	25
iPhone Jailbreak	25
Logical Acquisition.....	27

Cyberduck over SSH.....	27
Extraction through SQLite database files.....	29
Passcode bypass	35
Physical Acquisition	37
Data extraction with Foremost	39
Data extraction with PhotoRec.....	42
iOS Anti-Forensics.....	47
Conclusions.....	48
References.....	50

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

THIS PAGE WAS INTENTIONALLY LEFT BLANK

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Acknowledgements

From this position I would like to thank everyone who contributed in the production of this postgraduate thesis, which proved to be one of the most important parts of my whole studies by giving me the chance to enrich my knowledge and experience on the special area of iOS Forensics.

I would also like to thank Dr. Christos Xenakis for his input and constructive criticism because it helped me improve my comprehension of the subject and technical writing and, therefore, any reader's comprehension of this thesis.

Last but not least, I would like to thank my family for the psychological and material support throughout my studies.

Abstract

Nowadays, where the needs of people have changed and increased a lot, smartphones and tablets have spread into the corporate environment and show no sign of receding. This means users will be performing work on devices other than the traditional organizational desktops or laptops running windows operating system. Since smartphones and tablets are equipped with more hardware and upgraded software than ever before they are being used to surf the internet, run different and heavy applications, transfer data and communicate with corporate mail servers. A large section of these devices are running Apple's iOS and the ability to perform accurate and clear forensics on these devices will be interesting due to the privacy of a device like iPhone is. This final postgraduate thesis will dive into the special field of iOS forensics and will take an in-depth look at methods that analyze the iPhone in an official legal manner.

Introduction

Generally, computer forensics has been the first and most well-known subject that professional forensic analysts occupied with in recent years. They concentrated primarily on non-volatile data and specifically data and evidence stored on digital storage media devices such as hard drives and optical drives. Computer memory forensics is considered to be in a primitive stage until today. Technology and peoples' knowledge about computers is growing in a feverish pace. However, except for the vast array of available memory acquisition software, there is still much room for improvement.

On the other hand, mobile phones and mainly smartphones have been widely disseminated on the market recently. More and more people of all ages use smartphones today and they seem to become reliant on their devices as part of their everyday lives. This is because of the large variety of applications and facilities these devices provide which have seen it overlap with computers. So, the need for mobile phone forensics is becoming increasingly mandatory. Mobile device forensics analysis includes technical examination of mobile phones and the recovery of data from the device. Data for analysis can be obtained from SIM cards, memory cards or the phone handset itself including its persistent and main memory.

In this postgraduate thesis, our main concern is forensics on mobile devices using Apple's iOS (iPhone Operating System). All the devices that use iOS as their operating system, such as iPhone and iPad are mainly characterized by great privacy. This is arising out from the fact that both hardware and software on most of iOS devices work in harmony in order to achieve high levels of data privacy and security in general. Therefore, little has been written about memory acquisition and analysis of iOS devices, one of the most active areas of research in the field of mobile forensics. Non-volatile memory constitutes the most important part for an investigation as it contains both existing and deleted data that reside in slack space. Volatile memory, also referred as RAM, is a critical piece of evidence for every forensic investigator since it contains a wealth of information that is gone as soon as the device is rebooted or turned off. Moreover, directories and files that reside on a file system partition constitute an element of great interest which can prove to be very useful for a professional forensic analyst.

Digital Forensics

Smartphones are considered more than devices with ever-growing capabilities than devices performing only phone calls. The popularity among users, the vast scope of services offered in conjunction with their increased availability, has consequently increased the interest of criminals to use this technology as well. The exploitation of personal and professional data stored, information regarding financial transactions, online purchases, social networking and location data has led to the increase of frauds, thefts, industrial espionage, harassment etc. At the same time, data stored in smartphones are essential to analysts and criminologists in terms of their investigation.

Because of all the above-mentioned reasons, the professionals who occupy with technology and generally digital systems started to deal with digital forensics. Digital forensics (sometimes known as digital forensic science) is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime. The term digital forensics was originally used as a synonym for computer forensics but has expanded to cover investigation of all devices capable of storing digital data. Digital forensics investigations have a variety of applications. The most common is to support or refute a hypothesis before criminal or civil (as part of the electronic discovery process) courts. Forensics may also feature in the private sector; such as during internal corporate investigations or intrusion investigation.

Digital forensics is divided into several sub-branches, relating to the type of digital devices involved, as they appear below:

- Computer forensics
- Network forensics
- Forensic data analysis
- Mobile device forensics
- Forensic data analysis

In this postgraduate project we will mainly occupy with iOS forensics, namely forensics which have to do with the operating system that Apple Inc uses for its

smartphones and tablets (iPhone and iPad). However, before we do this, we will examine mobile device forensics, as iOS forensics is a sub-branch of them.

The current state

Smartphones operating system market share

Smartphones have gained popularity as far as mobile communication is concerned lately. Devices committed to acting as mobile phones and a personal computer at the same time have become a trend, while raising security issues regarding the type of data communicated and stored. Among others, those security issues are related to the operating system in use.

Gartner Inc. reports regarding 2012, indicates that sales of mobile phones have declined to a percentage of 2.3% compared to the sales of 2011 due to the challenging economic environment and users postponing upgrades in order to take advantage of high-profile device launches and promotions available later in the year.

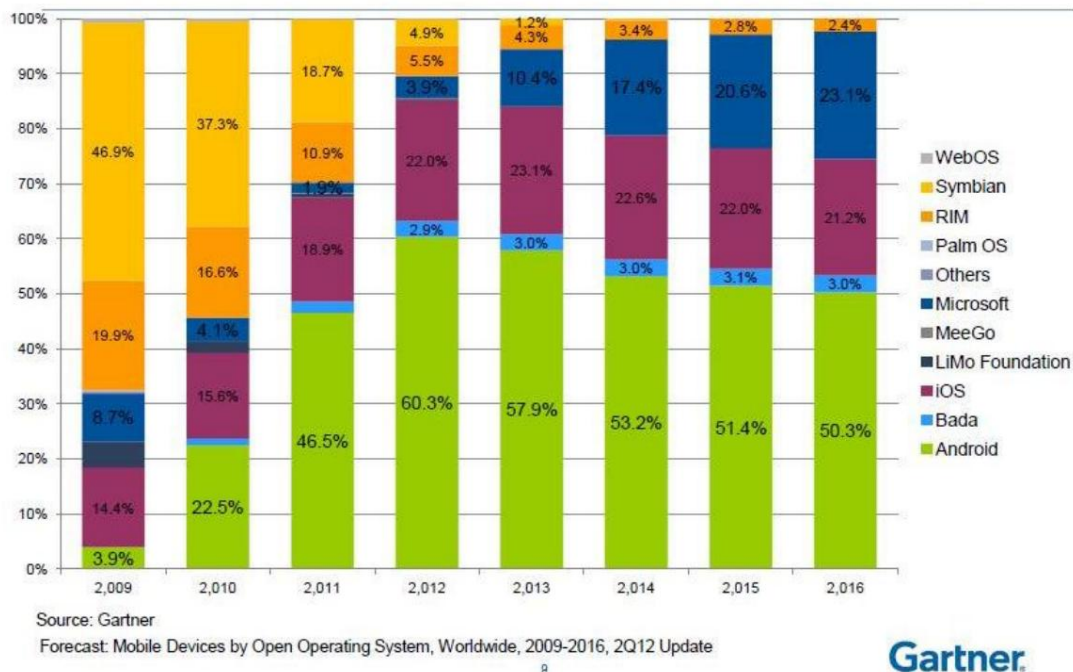


Illustration 1: Smartphone OS market share (via Gartner, Inc)

The picture above illustrates that Android operating system dominates mobile operating systems shipments in 2013 followed by Apple's iOS. However, numerous

operating systems for smartphones appear in the foreground as years pass and customers' preferences change a lot while operating systems' specifications are improving dramatically.

Smartphones Security

Smartphones are nowadays a source of valuable information depending to the user's activity. Data stored in the device or the applications used can reveal information related to payments, work or personal data, communication details as well as interests and hobbies. Thus it is wrong and critical for users to be assured that this type of information is kept secure in their device. In addition, smartphones mobility poses more risks to the confidentiality of information. Finally, user's lack of awareness related to those risks can lead to disclosure incidents in cases that the appropriate precautions have not been taken.

Ponemon Institute has conducted a research regarding security issues raised by smart phones and presented the results. The most salient research highlights, are listed below:

- 84% of consumers use their smartphones for both professional and personal purposes
- 66% of consumers store personal data
- 42% of consumers usage for social networking
- Less than 50% of consumers has taken security measures
- 58% of consumers were targeted by marketers who exploited the users' activity on purchases, Internet browsing and location

The results of the research indicate that most users (84%) are using the same device for both professional and personal practices. Thus, data, such as personal and corporate emails, contact lists, browsing history, location services, purchase and web banking applications are processed and stored. A potential loss of the device or an unauthorized access remotely (i.e. connection to a non-secure Wi-Fi) in conjunction with the lack of essential security measures can lead to disclosure of confidential information, thefts, forgeries, industrial espionage or targeted advertising. Moreover, it appears that less than the half of participants have taken security measures on their device. As a result, the risk of information disclosure is more contingent.

Mobile Device Forensics

Mobile device forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions. It differs from Computer forensics in that a mobile device will have an inbuilt communication system (e.g. GSM) and, usually, proprietary storage mechanisms. The phrase mobile device usually refers to mobile phones; however, it can also relate to any digital device that has both internal memory and communication ability, including smartphones, GPS devices and tablet computers.

The use of phones in crime was widely recognized for some years, but the forensic study of mobile devices is a relatively new field, dating from the early 2000s. A proliferation of phones (particularly smartphones) on the consumer market caused a demand for forensic examination of the devices, which could not be met by existing computer forensics techniques.

Mobile devices can be used to save several types of personal information such as contacts, photos, calendars and notes, SMS and MMS messages. Smartphones may additionally contain video, email, web browsing information, location information, and social networking messages and contacts.

Mobile device forensics is best known for its application to law enforcement investigations. Nevertheless, it is also useful for military intelligence, corporate investigations, private investigations, criminal and civil defense, and electronic discovery.

Evidence of great interest

As mobile device technology evolves, the amount and types of data that can be found on a mobile device is constantly increasing. Evidence that can be potentially recovered from a mobile phone may come from several different sources, including handset memory, SIM card, and attached memory cards such as SD cards.

Traditionally mobile phone forensics has been associated with recovering SMS and MMS messaging, as well as call logs, contact lists and phone IMEI/ESN information. However, newer generations of smartphones also include wider varieties of information. Those are web browsing, Wireless network settings, geolocation information (including geotags contained within

image metadata), e-mail and other forms of rich internet media, including important data, such as social networking service posts, contacts and passwords, now retained on smartphone applications. Other information of great interest could also be the photos, videos and music (sound files) that would be possibly stored in the smartphone's memory.

Live forensics

The data processed and stored in a mobile device can be categorized as communication information (contacts, chat logs, SMS, MMS, emails, etc), personal data (multimedia files, calendars, etc), location data (GPS data) and other data which are used by third-party applications. Some of these information are stored in persistent memory or as live data in the smartphone's RAM (Read-Only Memory).

Live forensics considers the value of the data that may be lost by powering down a system and collect it while the system is still running. The other objective of live forensics is to minimize impacts to the integrity of data while collecting evidence from the suspect system. So, live forensics refer to the information that will be drawn by the smartphone's RAM as this type of memory loses its data when the device is turned off. These information are called live data or else volatile data whereas the data stored in persistent memory are non-volatile. Nevertheless, sometimes the electrical power of the battery stored in the memory cells is capable of keeping data integral inside the device's RAM and makes it easier for a forensics professional to extract them.

Data storage areas

Stored data can be found in different areas inside the smartphone. Those areas can be the following:

- **Internal memory**
Nowadays mostly flash memory consisting of NAND or NOR types are used for mobile devices.
- **External memory**
External memory devices are SIM cards, SD cards (commonly found within GPS devices as well as mobile phones), MMC cards, CF cards, and the Memory Stick.

- Service Provider logs

Although not technically part of mobile device forensics, the call detail records (and occasionally, text messages) from wireless carriers often serve as back up evidence obtained after the mobile phone has been seized. These are useful when the call history and/or text messages have been deleted from the phone, or when location-based services are not turned on. Call detail records can show the phone owner's location, and whether they were stationary or moving. Carrier data and device data together can be used to corroborate information from other sources, for instance, video surveillance footage or eyewitness accounts or to determine the general location where a non-geotagged image or video was taken.

Forensics procedure

The forensics procedure for mobile devices is very similar to other branches of digital forensics. Generally, the procedure can be separated into four main categories which are the following:

- Collection phase: identification, recording and acquisition of data from all possible sources of relevant data, while preserving the integrity of data acquired.
- Examination phase: process of collected data in a forensically sound manner, using a combination of automated and manual methods, and assessing and extracting data of particular interest.
- Analysis phase: analysis of the results of previous phase, using legally justifiable methods and techniques, to derive useful information related to the investigation in progress.
- Reporting phase: reporting of the results of the previous phase, which may include describing the actions used, explaining how tools and procedures were selected, determining what other actions need to be performed and providing recommendations for improvement to policies, procedures, tools, and other aspects of the forensic process.

Methods of data acquisition

Generally speaking the methods of smartphones data acquisition can be categorized as follows:

- Manual acquisition

In this type of acquisition, the examiner utilizes the user interface to investigate the content of the phone's memory. Therefore, the device is used as normal, with the examiner taking pictures of each screen's contents. This method has an advantage in that the operating system makes it unnecessary to use specialized tools or equipment to transform raw data into human interpretable information and all the phone's data is directly available to the professional. In practice this method is applied to cell phones, PDAs and navigation systems. A main disadvantage is that only data that are visible to the operating system can be recovered. Additionally, all data can only be available in pictures and the process itself is time-consuming.

- Logical acquisition

Logical acquisition creates a bit-by-bit copy of logical storage objects such as directories and files that reside on a logical store which could be a file system partition. Logical acquisition has the advantage that system data structures are easier for a tool to extract and organize. Logical extraction acquires information from the device by using a program which synchronizes the phone's contents with a personal computer. A logical extraction is generally easier for a professional and not only to work with.

- File system acquisition

Sometimes logical extraction does not recover any deleted information, because it can be normally removed from the phone's file system. However, in some cases, particularly with platforms built on SQLite databases, such as iOS and Android, the smartphone may keep a database file of information which does not overwrite the information but simply marks it as deleted and available for later overwriting. In such cases, if the device allows file system access through its synchronization interface, it is possible to recover deleted information. File system extraction is useful for everyone to understand the file structure, web browsing history, or app usage of a smartphone. It can also provide the examiner with the ability to conduct an analysis with traditional computer forensic tools.

- Physical acquisition

Physical acquisition creates a bit-for-bit copy of the entire physical storage of a smartphone's memory. Hence, it is the method that is more close to the examination of a personal computer. Physical acquisition is the only method of acquisitions which allows the examiner to detect deleted files inside the smartphone's memory. Physical extraction acquires information from the device by directly accessing the flash memory. Generally this method is harder for a professional to perform because the manufacturers want to secure their devices against arbitrary reading of memory. Consequently, a device may be locked to a certain operator and in order to overcome this security, mobile forensics tool developers (and not only) often develop their own bootloaders, giving the opportunity to the professional examiners to access the smartphone's memory.

Smartphones have deluged the market so more and more people use them in their everyday lives for every need and purpose. On the other hand, forensic examiners are obliged to do their job correctly so new methods of acquisition are discovered continuously, as the already existing tools become more expensive, analysis takes longer and every mobile operating system may need different approach and confrontation.

History of Apple Inc

Apple Inc., formerly Apple Computer Inc., is an American multinational corporation headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software and personal computers. Its best-known hardware products are the Mac line of computers, the iPod music player, the iPhone smartphone, and the iPad tablet computer. Its consumer software includes the OS X and iOS operating systems (for its computers and smartphones respectively), the iTunes media browser, the Safari web browser, and the iLife and iWork creativity and productivity suites.

Apple had a history of trials and failures until the release of the iPhone, which is the phone that actually changed the mobile phone game and Apple's profits. For instance, in 1988, Apple started the development of the Newton which was an early version of a PDA tablet. The first Newton project was the Message Pad 100, released in August

1993, and the last was MessagePad 2100, released in November 1997. The Newton line of products was subsequently killed upon the return of Steve Jobs to Apple in 1997. There were six models of the Newton, and all of them had an ARM processor, with a clock speed of 20MHz to 162MHz. The Message Pad also had its own operating system called NewtonOS. The platform had a touchscreen, handwriting recognition, and applications that were able to share information in “soups.” Soups were not unlike what we see in the iPhone’s databases, where one application can refer to data in another application. Therefore, the Newton had everything that a usual PDA had to offer to consumers. However, this device didn’t have the expected acceptance by them, who preferred to buy other PDAs with similar characteristics. After that, Apple turned its focus on developing a device that would be widely accepted by people. This was the first iPod which proved to be the springboard for the oncoming iPhone and iPad.

Evolution of iOS devices

On January 9, 2007 Steve Jobs who was the co-founder, chairman and CEO of Apple Inc, announced the first iPhone at the Macworld convention, receiving substantial media attention, and that it would be released in this year. On June 29, 2007 the first iPhone, which is well-known as iPhone 2G, was released in the market and it got great acceptance by numerous people around the world. Because iPhone met so great success, Apple continued to promote new devices every year. Those mobile devices which belong to the family of smartphones are the following:

- iPhone 3G (*11/7/2008 – 7/6/2010*)
- iPhone 3GS (*19/6/2009 – 12/9/2012*)
- iPhone 4 (*24/6/2010 (black model) and 10/2/2011 (white model) – 10/9/2013*)
- iPhone 4s (*availability by country, 14/10/2011 – 10/9/2013*)
- iPhone 5 (*availability by country, 21/9/2012 – 10/9/2013*)
- iPhone 5C and 5S (*availability by country, 20/9/2013*)

Simultaneously with the forwarding of those devices, Apple proceeded to the production of tablet computers that proved to be very useful for people from their

everyday lives to their jobs. Those tablets are referred to as iPads and they became very likeable to consumers until today, with their latest versions.

iPhone Operating System (iOS)

iOS is a mobile operating system developed and distributed by Apple Inc. The term iOS is derived from OS X which is the operating system that Apple uses in its computers. It was originally unveiled in June 2008 for the first generation iPhone and it revolutionized the way cell phones would be created in the future. After a period, it was extended and configured in order to support other and newer Apple devices such as iPad tablets, iPod Touch and Apple TV. iOS is programmed in ARM assembly, C, C++ and Objective C and it is Unix-like (BSD). It is a scaled-down version of OS X and iOS devices use a variant of the Mac OS X kernel. Moreover, it is a closed source operating system which means that Apple does not licence iOS for installation on other non-Apple devices and hardware, as Microsoft and Google does with Windows Phone and Android respectively. It consists of four abstraction layers: the Core OS layer, the Core Services layer, the Media layer and the Cocoa Touch layer. The OS totally drains approximately 350MB (in newer versions it reaches between 1-1.5GB) of the device's memory storage, which means that users don't get their full storage space.

The iOS platform works like a computer system, but on mobile devices and it is designed to be smaller, faster and use less power. The user interface is very friendly, simple and functional and it is based on the concept of direct manipulation, using multi-touch gestures. So, interaction with this type of OS includes different types of gestures which vary from iDevice to iDevice (such as from iPhone to iPad). All of these gestures have specific definitions within the context of the iOS and its multi-touch interface.

iOS versions

The first version of what became iOS was released on June 29, 2007, concurrently with the first iPhone and was called iPhone OS 1.x (Operating System). The "x" character that is added after the number of each version will indicate the release of some sub-versions which fix different bugs and problems of the operating system or offer new functions. It was the first iteration of Apple's touch-centric mobile operating system which gave users a new experience. The iPhone OS 1.x gave iPhone

applications such as SMS, calendar, photos, camera, Youtube, stocks, maps, weather, notes, clock, calculator, settings, iTunes, phone, mail, safari and iPod. The user interface (UI) that iPhone OS offered to the iPhone 2G, has a top portion that displays network strength, network type, time, Bluetooth icons and battery strength. Below the portion of the UI are the home screens. Each screen is able to hold 16 applications, which at first were web applications that could be downloaded from Apple or bookmarks from Safari that could be added to a home screen. The dock consists of four icons. At first this could not be changed but new versions of iPhone OS allowed every application to be placed in the dock.



Home Screen



Dock

Illustration 2: iOS 1.x as it appears on iPhone 2G

On July 11, 2008 iPhone OS 2.x made its debut and because it contained the App Store, it supported third-party applications. It included numerous user enhancements, including screen capture and tapping the top of the screen to scroll to the top. It offered a GPS application to the iPhone as well, which allowed many applications, such as Google Maps and Camera, to use GPS API for a multitude of purposes. When GPS first became available to the iPhone, it was not very accurate. However, with subsequent firmware updates and the next generation iPhone, the accuracy improved.

The iPhone OS 3.x was released on June 17, 2009 and became available with the iPhone 3GS. Many features were added in it, including cut, copy and paste that were glaringly missing in previous versions. It also contained a global search, multiplayer gaming, turn-by-turn navigation using a third-party application, purchasing upgrades within the application, push notifications (applications to be notified even if they are not running on the foreground) and MMS multimedia messaging. iPhone OS 3.x was available for the iPhone 3G and the iPod Touch.

The iOS 4.x coincides with the iPhone 4 and was made available to the public for the iPhone and iPod Touch on June 21, 2010. Apple licenced the iOS brand name from Cisco and iPhone OS was renamed to iOS. Along with multitasking, Apple touted more than 100 features. A very important feature, was that folders were added, allowing a user to create one by dragging one application icon onto another in a wiggle mode.

The iOS 5.x was released for iPhone 3GS, iPhone 4, iPhone 4S, iPod Touch, iPad and iPad 2 on October 12, 2011. After that, iOS could be updated over the air instead of only via iTunes connected to a computer. iCloud could synchronize data among all Apple devices wireless as well. There were a lot of camera enhancements that were added, such as cropping and red-eye reduction. Additionally, similar to the Android, a brand new notification centre allows users to review messages and alerts in a central location. Another innovation was iMessage which let users send text messages to other iOS devices provided that both of them have an internet connection.

The sixth version of iOS which was iOS 6.x appeared on September 2012 coinciding with the iPhone 5. A lot of functions were added to the renewed mobile operating system of Apple. So, Facetime was also available over cellular instead of only Wi-Fi, mail was improved and new web browsing and Siri functionalities were incorporated in the operating system. Apple replaced Google's map data with its own called Apple Maps, as well.

Finally, the most recent version of Apple's operating system for mobile devices is iOS 7.x, which was fully released on September 18, 2013 and it is a complete change in the user interface design since the iPhone's inception. It uses a native 64-bit kernel, libraries and drivers, all the built-in applications are re-engineered but it also runs both 32-bit and 64-bit applications. Bold colors, different appearance of the

application icons and dynamic wallpapers are only a few changes that happened with the creation of iOS 7. Nevertheless, the most important changes are considered to be the extremely fast and easy multitasking, the notification center that gives information to the user about every new notification and the control panel which consists of some necessary to all users functions that can be directly manipulated by there.



Illustration 3: The iOS 7 homescreen (completely different from the previous versions till iOS 6)

In the diagram below we can clearly detect which versions of iOS are preferred by iPhone users at this current state.

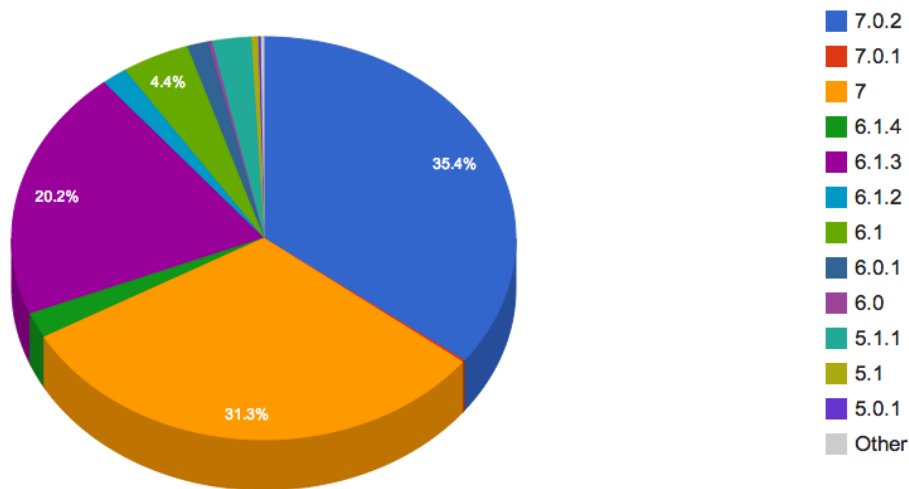


Illustration 4: iOS version preference from 5.0.1 to 7.0.2

Kernel

The core of iOS (and Mac OS X) is the XNU Kernel and it is based on Darwin OS. XNU has a layered architecture consisting of three major components. The inner ring of the kernel is referred to as the Mach layer, derived from the Mach 3.0 kernel. Mach was developed as a microkernel, as well as IPC (inter-process communication), which is a core concept of the Mach kernel. Because of the layered architecture, there are minimal differences between the iOS and Mac OS versions of XNU.

While the Mach layer in XNU has the same responsibilities as in the original project, other operating system services, such as file systems and networking, run in the same memory space as Mach. Apple cites performance as the key reason for doing this, as switching between address spaces in an expensive operation.

Because the Mach layer is still an isolated component to some degree, many refer to XNU as a hybrid kernel, as opposed to a microkernel or a monolithic kernel, where all OS services run in the same context.

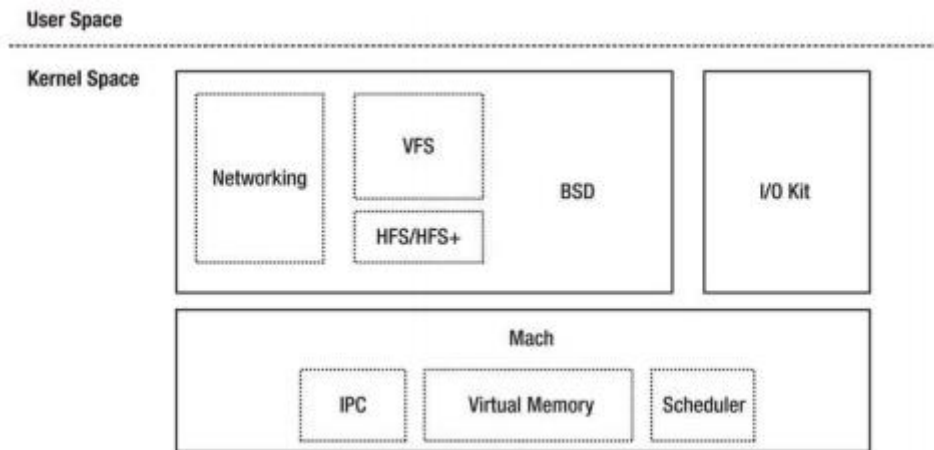


Illustration 5: XNU kernel architecture

The second major component of XNU is the BSD layer, which can be thought of as an outer ring around the Mach layer. BSD provides again a programming interface to end-user applications. Responsibilities include process management, file systems and networking.

The third and last major component is the I/O Kit, which provides an object-oriented framework for device drivers.

While it would be nice if each layer had clear responsibilities, reality is somewhat more complicated and the lines between each layer are blurred, as many OS services and tasks span the borders of multiple components.

iOS SDK

Beginning with iOS 2, Apple allowed the development of applications for its App store. The iOS SDK (Software Development Kit) is a software development kit developed by Apple Inc in order to develop native applications for iOS devices. It provides a complete and integrated process for developing, debugging and distributing free, commercial or even in-house applications. However, because Apple has a strict and sometimes time-consuming approval process, loading an application onto the devices is only possible after paying an iOS Developer Program fee, which is \$99.00 per year. The SDK includes the Xcode which is the development environment where all the procedure of writing an application is done. As aforementioned, because iOS uses a variant of the same XNU kernel that is found in OS X, the tool chain used for developing on iOS is also based on Xcode.

iOS components

At this point of the postgraduate thesis, we will analyze the internal components of the iOS operating system such as the boot process, the file system, the partitions and the databases. This will help not only a professional forensics analyst but a simple user as well, to understand the forensic procedure that can be held inside an iOS device.

Boot process

When an iPhone boots up, it walks through a chain of trust, which is a series of RSA signature checks among the software components in a specific order as shown below.

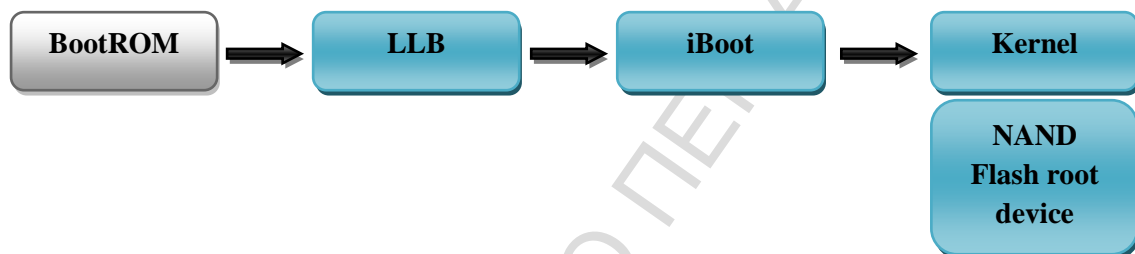


Illustration 6: Normal mode boot sequence

The BootRom is Read-Only Memory and it is the first stage of booting an iOS device. BootRom contains all the root certificates to signature check the next stage and has keys for subsequent phases that are used to verify the integrity of the later boot stages.

An iPhone can operate in three modes:

- Normal mode
- Recovery mode
- DFU mode

In normal mode, which appears in illustration 6, BootROM start off some initialization stuff and loads the low level boot loader (LLB) by verifying its signature. LLB signature checks and loads the stage 2 boot loader (iBoot). iBoot signature checks the kernel and device tree, while the kernel signature checks all the user applications.

In DFU mode, iPhone follows the boot sequence with a series of signature checks. BootROM signature checks the second level boot loaders (iBSS and iBEC). Boot

loader signature checks the kernel and afterwards the kernel signature checks the Ramdisk. This mode is designed to perform firmware update for iPhone. All this procedure is shown in the following illustration.

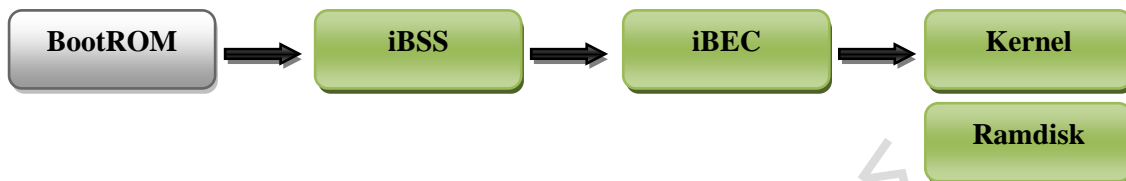


Illustration 7: DFU mode boot sequence

iOS file system

The local storage on an iOS mobile device has several differences from the traditional Microsoft Windows or UNIX operating systems. Understanding those differences can help a mobile investigator understand which tools to utilize and which actions to take in order to reach the desirable result.

As physical disk space was increasing with an extremely fast speed, a file system had to be developed by Apple to support the growing need for storage. So, Apple developed the Hierarchical File System (HFS) which was able to accommodate storing of large data sets. At the physical level, the disks formatted with HFS are in 512-byte blocks and this is similar to Windows-based sectors. There are two types of blocks in the HFS system:

- Logical blocks
- Allocation blocks

The logical blocks are numbered from the first block to the last block available on the volume and will remain static. On the other hand, the allocation blocks can be tied together as groups to be utilized more efficiently by HFS.

Data within the HFS file system utilizes a catalog file system or B*tree to organize files. This balanced tree uses a catalog file and extents overflows in its organization scheme. B*trees are comprised of nodes. These nodes are grouped together in linear fashion and that makes data access faster. When data is added or deleted, the extents

are constantly balanced in order to keep its efficiency. Each file that is created on an HFS file system is given a unique number that is called a catalog ID number. The HFS volume header tracks the numbering of the catalog ID and will increment by one when a new file is added. These numbers can be reused, but this is tracked by the HFS volume header.

There are nine structures that make up a typical HFS+ volume:

1. Sectors 0 and 1 of the volume constitute the first 1024 bytes and are reserved for *boot blocks*.
2. The *Volume Header* which reserves the next 1024 bytes and stores a wide variety of data about the volume itself. These data can be the size of allocation blocks, a timestamp that indicates when the volume was created or the location of other volume structures such as the Catalog File or Extent Overflow File. There is also a backup volume header at the last 1024 bytes of the HFS volume and some volume header signatures. The Volume Header is always located in the same place.
3. The *Allocation File* that simply tracks which allocation blocks are free and which of them are in use.
4. The *Extents Overflow File* which is a B-tree that tracks all the allocation blocks belonging to a file's data fork. It contains a list of all extents used by a file and the associated blocks in the appropriate order.
5. The *Catalog File* which is a catalog file system used by the HFS+ file system in order to maintain all the information related with files and folders within a volume.
6. The *Attributes File* which is a new B-tree in HFS+ file system that stores three different types of records:
 - *Inline Data Attribute records* that store small attributes that can fit within the record itself.
 - *Fork Data Attribute records* which contain references to a maximum of eight extents that can hold larger attributes.
 - *Extension Attribute records* which are used in order to extent a Fork Data Attribute record when its eight extent records are already used.

7. The **Startup File** which was a file designed to assist in booting a system that did not have built-in ROM support.
8. The **Alternate Volume Header** that constitutes a backup volume header and is mainly used for disk repair.
9. The last sector in the volume consists of 512 bytes reserved for use by Apple.

All the above-mentioned structures are figured in the illustration below.

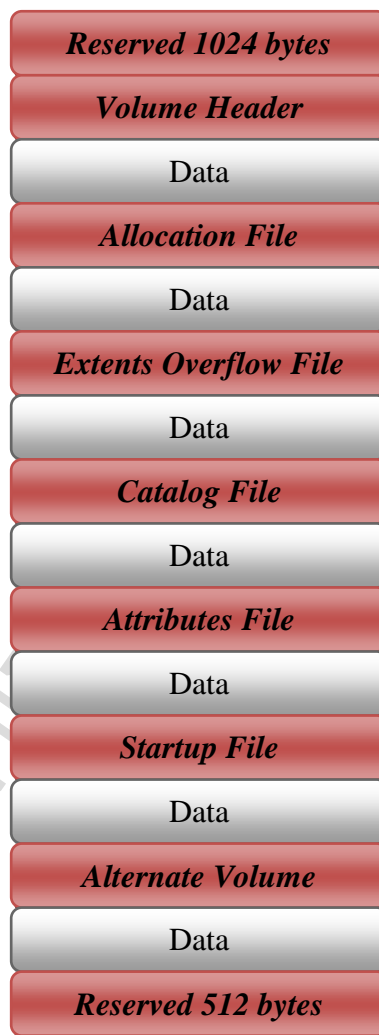


Illustration 8: The structure of the HFS+ file system

Partitions

All iOS devices have two partitions. The first partition is the system or root partition and is reserved for iOS file systems. It is the smaller of the two partitions and it contains the operating system and the default applications that are delivered with a

factory fresh iOS device. This partition is read-only unless a firmware update is being performed. So, when an upgrade is performed, this partition is overwritten by iTunes or over the air (by Apple servers) with the new partition.

The second partition is the resuming space of the hard disk and is partitioned as the user-space partition. It is the larger of the two partitions and varies depending on the flash memory available which with its turn is dependent on the device's total internal storage. It contains all the third-party applications, music, photos, videos, contacts, SMS and other things that a user would like to store in his device. Consequently, this specific partition is the focus of most forensic investigations.

The two partitions of a jailbroken iOS device are shown in the illustration below.

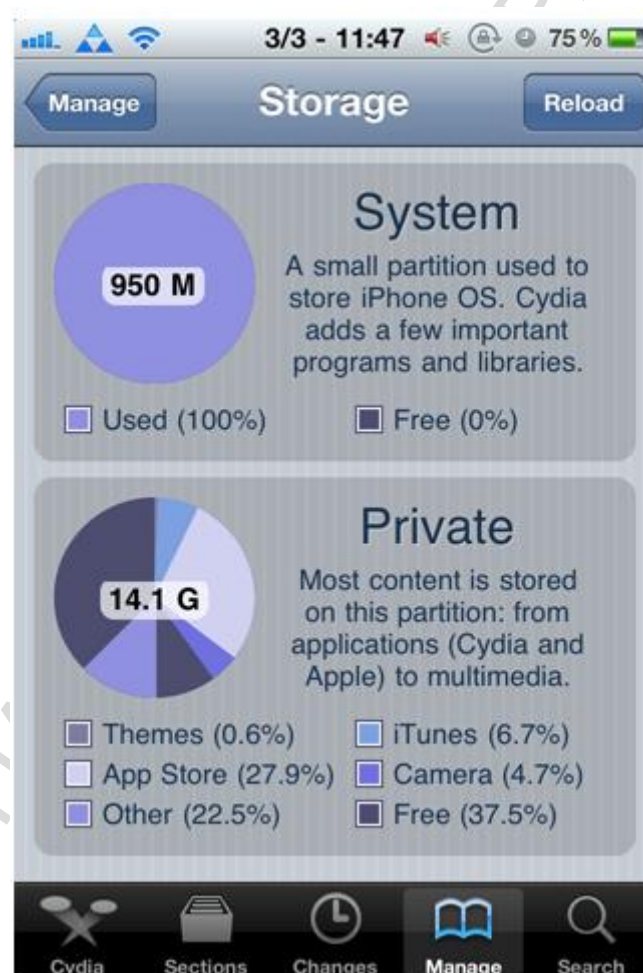


Illustration 9: The two partitions of an iOS device

Databases

The iOS operating system uses the SQLite database format in order to store information on the device. The SQLite is an embedded SQL database engine and its format is generally very popular for mobile devices and open source applications. This database is relational and can be completely contained in a small C programming library. The iOS shows to have numerous SQLite databases for the operation of the phone and by developers of applications. Many of the native iOS applications that an iPhone has factory installed, as Address Book, SMS, Call History, Calendar, Notes, Photos, Videos and Music is, use SQLite database to store and organize their data. The iPhone also uses these databases to cross-reference information from one database to another, which gets displayed on the user interface. Thus, these databases interact with each other so as to give the user an excellent informative experience.

In order for a forensics analyst to manage those databases and view all the important clues and evidence inside them, there are a few databases viewers for this specific reason such as the SQLite Browser of Sourceforge.net and the free SQLite Manager plugin of Mozilla Firefox Internet Browser.

Property Lists

The property list (or colloquially referred to as “plist”) is a structured data representation used by Cocoa and Core Foundation as a convenient way to store, organize and access standard types of data. Property lists are XML files which are used extensively by applications and other software on OS X and iOS. So, applications that maintain configuration data such as browsing history, favorites, configuration data, etc can place their data in a plist. Moreover, applications on iOS use plists in their settings bundle to define the list of options displayed to users.

If a forensics analyst encounters a plist file and he is not able to open it with a standard text editor, then a viewer such as Plutil will be needed. Plutil is a command line tool which can convert the binary plist files into human readable form.

Forensics procedure prerequisites

For the needs of this postgraduate thesis, a second hand iPhone 3G was used. This device has 8GB internal storage capacity and it runs the latest software which is iOS 4.2.1 firmware. Additionally, the device seems to have been reset to its factory settings or the previous user deleted manually the most of its contents, as it does not contain any photos, music, videos or messages. The only things that do exist are 5 contacts, a few incoming calls, 2 applications and 1 game.

The forensics workstation, in which all the data acquisition process took place, is an Apple's MacBook Pro, running OS X Mavericks 10.9.1 operating system.

Some information about the iPhone 3G, which derive from the connection of the device with iTunes through the MacBook Pro, are shown in the following picture.



Illustration 10: iPhone 3G connected with iTunes on MacBook Pro computer

Existence of deleted data

Practically, every hard drive contains deleted or lost data that can be recovered. Any time that a file is deleted from a hard drive, it is not erased. What is actually erased is the bit of information that points to the location of the file on the hard drive. Every operating system, and so does the iOS, uses these pointers to build the directory tree structure or the allocation table, which consists of the pointers for every other file on the hard drive. When the pointer is erased, the file essentially becomes invisible to the

operating system. Nevertheless, the file still exists but the operating system's file explorer just does not have a way in order to find it.

From all the above mentioned, it becomes clear why the iPhone 3G in illustration 10 has only 3,74GB free from the 6,9GB of its total memory, although the device has nearly nothing stored in it. The yellow line represents the percentage of stored data in the device. However, the device can't assort these data into different categories because it can no longer revoke them.

iPhone and iOS restrictions

Unlike computer forensics, where the professional analyst is capable of exporting the hard drive out of a computer and then extracts all of its internal data, smartphones, including iPhones, are supplied with integrated hardware that can't be exported out of the device or else it will become useless. So, the only way for a forensics investigator to gain access to the internal memory of a smartphone is to connect it with a computer through a USB cable. This is mainly met on Android smartphones, where the computer identifies the device as an external memory storage after the connection is completed and the examiner is able to communicate with it using ADB (Android Debugging Bridge) and then extract all the necessary information he wants.

On the other hand, all iOS devices seem not to respond with the same way as Android smartphones do, when connected to a computer. Although an iOS device can be accessed via iTunes, it does not mount on the desktop as a hard drive. Hence, it does not appear as a drive that can be scanned with data recovery software. Apple does not allow this since iOS is a closed source operating system and iPhones are considered to be very secure devices, so it would take away a lot of their power as a sole provider. Consequently, the only way for a professional to make a computer identify the iPhone as a storage drive, is to jailbreak the device and then make an SSH (Secure Shell) connection between this and the computer.

iPhone Jailbreak

iOS jailbreak is defined as the process of removing the limitations on Apple's devices running the iOS operating system through the use of software and hardware exploits. Jailbreaking is a form of privilege escalation and permits root access to the iOS operating system. It allows the user to do things that could not traditionally do with

the factory installed firmware, such as to download additional applications, extensions and themes which are neither available nor free through the App Store.

Jailbreaking the iPhone, and generally any other device that runs Apple's iOS, is a very easy and simple process as long as the appropriate software does exist. By far the most popular and well-known tool for jailbreaking is redSn0w. For this specific case, redSn0w 0.9.6b4 edition was used. This software automates the procedure in a few steps through a simple wizard which will pass the iOS device through the process of replacing the firmware. First of all, the program asks the user to define the software version of the iPhone. Then, the device must be turned off and plugged in a personal computer via a USB cable. The final step asks from the user to put his device in DFU mode and after that, all the remaining process takes place on the phone itself. The iPhone reboots and when it turns on, the Cydia application is already installed in it which shows that jailbreak was successful.

Therefore, after a successful jailbreak on an iOS device, the forensics analyst will have a lot of tools not normally available, such as SSH and Terminal. These tools will help him to obtain an image file of the device's partition that can be manipulated by specific software. SSH function can be activated on iPhone by downloading the openSSH application from Cydia and the controlling of SSH state can be done very easily by another application called SBSettings. Furthermore, after jailbreak the analyst will be able to access the root folder of the iPhone as an administrator and browse a lot of interesting files that he could not be able to see as a simple user.

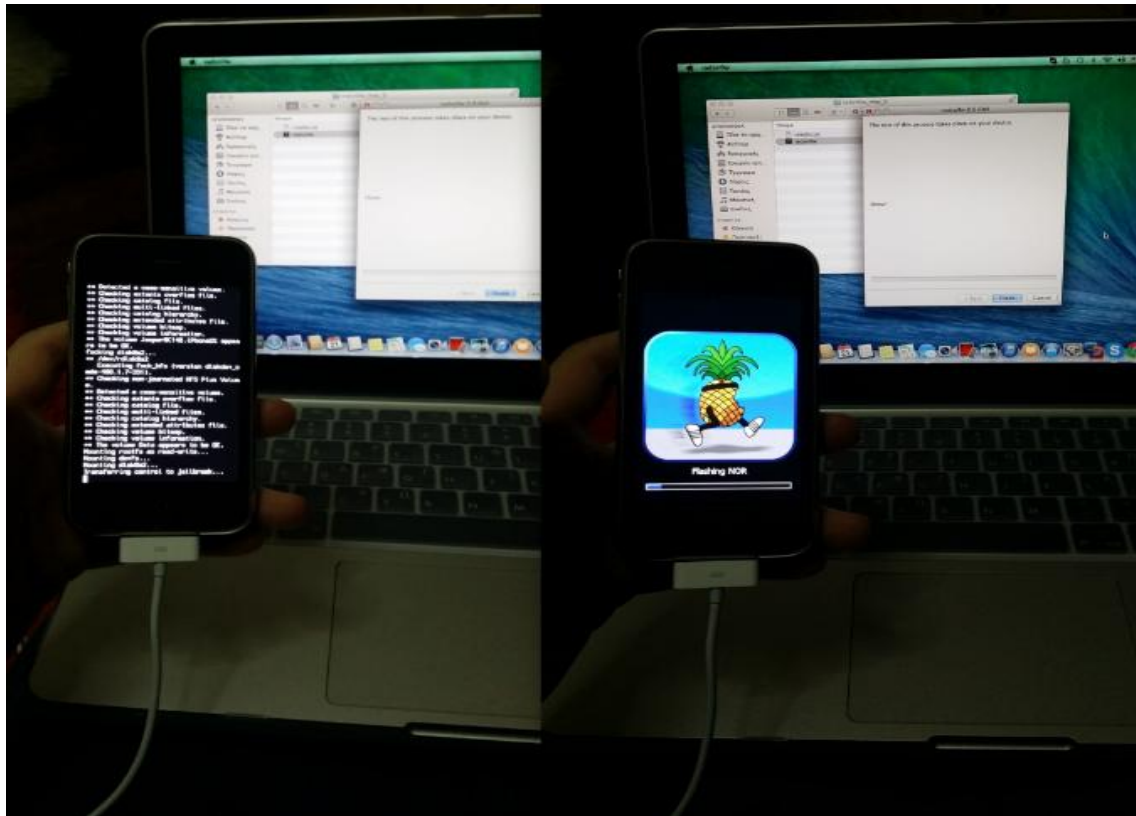


Illustration 11: Two different phases of the iPhone jailbreak

Logical Acquisition

One of the most popular approaches today for smartphones data extraction, and not only, is the logical acquisition method. This method does not often recover any deleted information. Nevertheless, by using this approach, all the allocated active files that reside on the iOS device, which is SQLite based, are recovered and analyzed using a synchronization method built into the iOS operating system. This will allow the analyst to gather important evidence on call logs, contacts, SMS, photos, web history, bookmarks, email accounts and passwords.

Logical acquisition has a growing market of tools sets that a professional forensics analyst can use for his investigation. However, one of the most complete and effective tools which recovers plenty of both existing and deleted data from the SQLite databases that iPhone devices use, is Cyberduck.

Cyberduck over SSH

Cyberduck is an open source FTP and SFTP client for Mac OS X computers which is written in the Java language and uses the Cocoa user interface framework. It features

an easy to use interface with quickly accessible bookmarks. The outline of the browser allows easy, fast and efficient browsing of large folder structures, the files of which can be previewed with Quick Look. It also includes a bookmark manager and supports the Mac OS X Keychain. In this case, the most important files for a forensics examiner, that contain evidence, are the SQLite database files residing in the iPhone's file system.

As mentioned above, SSH connection between the forensics workstation and the iPhone device is necessary in order for the computer to recognize the smartphone as an external memory disk and reveal all its contents. SSH (Secure Shell) is a cryptographic network protocol that allows secure data communication, remote command-line login, remote command execution and other secure network services between two devices that are connected to the same network. In a jailbroken iPhone, an SSH connection over a wireless network is the best way for browsing the device's contents and carrying out simple file transfer back and forth to it.

In this project, the iPhone device will be connected via SSH to the forensics workstation which runs Cyberduck. The two devices need to be connected to the same wireless network. First of all, the SSH function on iPhone has to be turned on, using the SBSettings application mentioned before. Another important information for making an SSH connection is the iPhone's IP address which can be found either on SBSettings application or on wireless networks settings. When launching Cyberduck on the forensics workstation, a choice of "Open Connection" exists. After chosen, some fields have to be filled in order for the program to know the exact device in which the connection must be made.

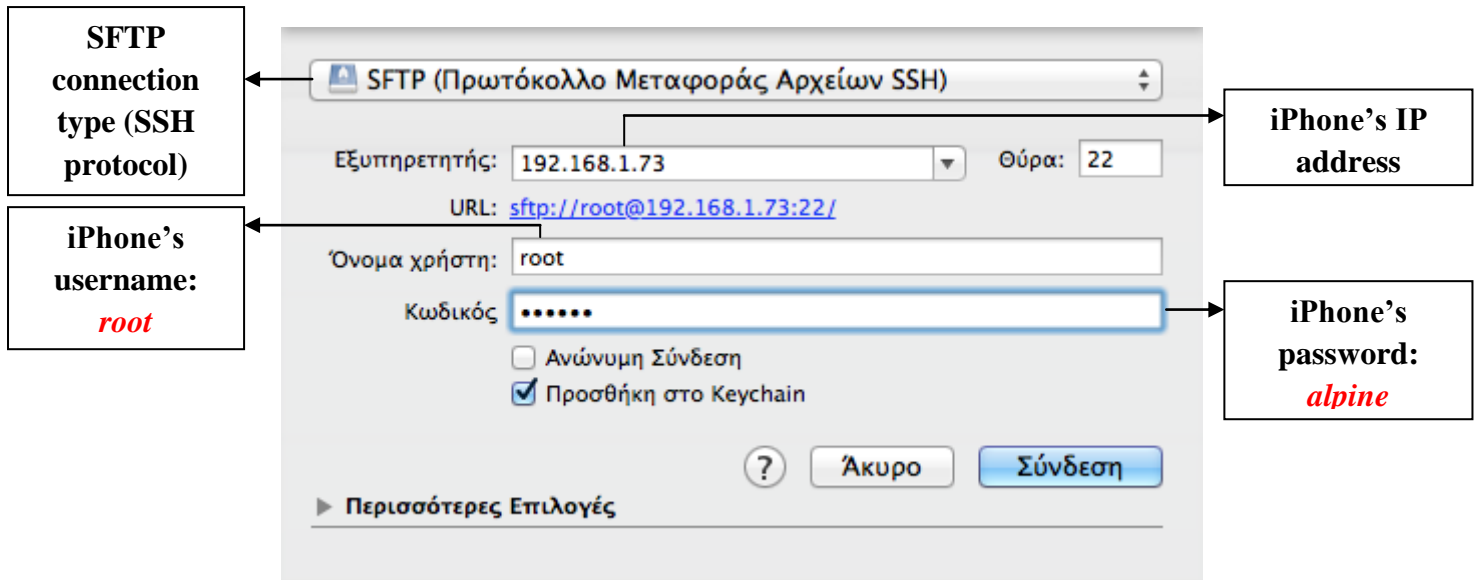


Illustration 12: Fields on Cyberduck for specifying the connection type and the device's information

After all the fields are correctly filled, the SSH connection between the two devices is finally established and the examiner is now able to dive into the iPhone's file system, as it is shown in the following picture.

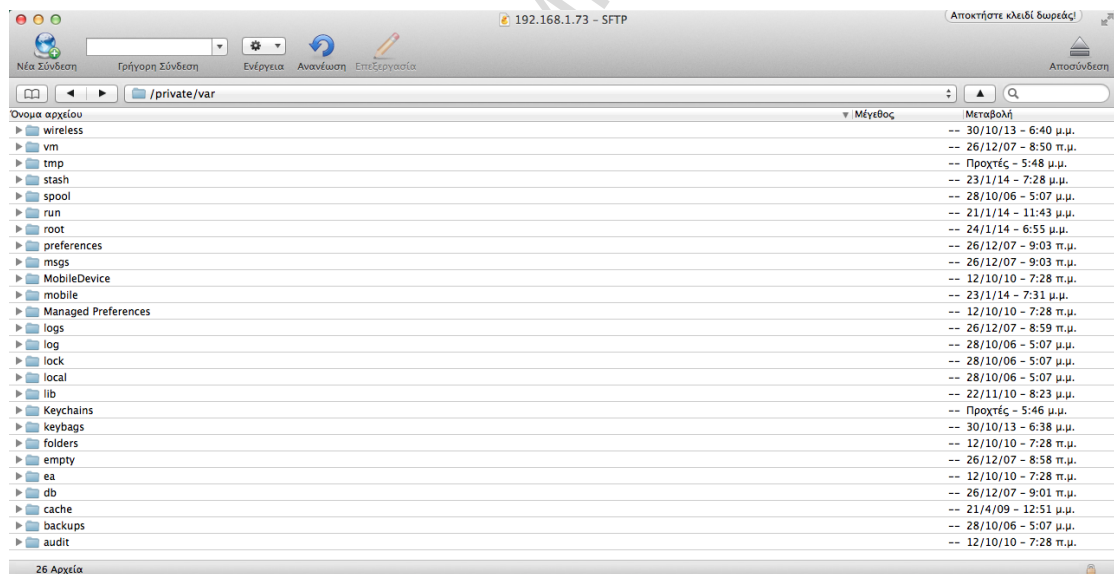


Illustration 13: Some of the numerous folders existing on the iPhone's file system

Extraction through SQLite database files

The iOS directory structure is common across all iOS devices. When accessing the internal file system of the device, the `/private/var` directory contains the paths to different files through SQLite databases, containing the most valuable information that an analyst can find about a user. In particular, the `/private/var/mobile/Library`

directory contains all the information about the Address book, SMS, Mail, Calendar, Notes and other pre-installed applications. On the other hand, all the downloaded applications by the user exist in the */private/var/mobile/Applications* directory.

The Address Book information are stored in the */private/var/mobile/Library/AddressBook* directory where the *AddressBook.sqlitedb* file does exist. This file can be easily opened with the SQLite Database Browser and it can reveal all the contacts stored in the iPhone alongside with their numbers.

ROWID	First	Last	Middle	FirstPhonetic	MiddlePhonetic	LastPh
1	13	B...				
2	14	M...				
3	15	C				
4	16		1			
5	17		2			

Illustration 14: Contact names in the “ABPerson” table of the AddressBook.sqlitedb file

UID	record_id	property	identifier	label	value
1	1	13	3	0	1 00 880 1813095921
2	2	14	3	0	1 00 880 1850596544
3	3	15	3	0	1 00 880 1824439179
4	4	16	3	0	1 694 9292299
5	5	17	3	0	1 694 0141940

Illustration 15: Contact numbers in the “ABMultiValue” table of the AddressBook.sqlitedb file

From the pictures above, it is deduced that the contact names and the contact numbers are stored in different tables inside the same database file, but they correspond with each other.

The SMS information are stored in the */private/var/mobile/Library/SMS* directory where the forensics examiner can find the *sms.db* file. In this occasion, the file gives the examiner more information, such as messages that don't exist in the device alongside with their service provider, modification date and country.

ROWID	address	date	text	flags	replace	svc_e
1	65	1388653908		129	0	
2	66	1388737693		129	0	
3	67 Vodafone	1389608761	ΜΙΛΗΣΤΕ ΔΩΡΕ	2	0	

Illustration 16: message table in the sms.db file

ROWID	group_id	address	country
1	33	QCARD	gr
2	34	CUoffer	gr
3	35	Vodafone	gr

Illustration 17: group_member table in the sms.db file

Other written information can be found inside the *private/var/mobile/Library/Notes* directory, where the notes.sqlite file is located. This file has all the notes that the iPhone user has made using the pre installed application in the device.

ZCREATIONDATE	ZMODIFICATIONDATE	ZTITLE
1 412958368.943742	412958368.943742	iExplorer
2 412958389.72143	412958389.72143	Mplamplampla

Illustration 18: “ZNOTE” table in the notes.sqlite file

The “ZNOTE” table in the notes.sqlite file comprises what the user has written in his iPhone “Notes” application. It also includes the exact date that the notes were made and the date that they were last modified.

As far as internet browsing is concerned, all iOS devices use Safari as their web browser. So, Safari cookies are a very important piece of evidence when identifying web browsing from the device. The persistent cookies of the browser are stored in the Cookies.binarycookies file which is located in the */private/var/mobile/Library/Cookies* directory. This file is in a binary format in contrary to other browsers such as Internet Explorer and Google Chrome where the cookies are saved in plain text, so a HEX editor is needed in order for the examiner to open it and read its components. The existing cookies can reveal the web sites that the user of the iOS device was usually visiting.

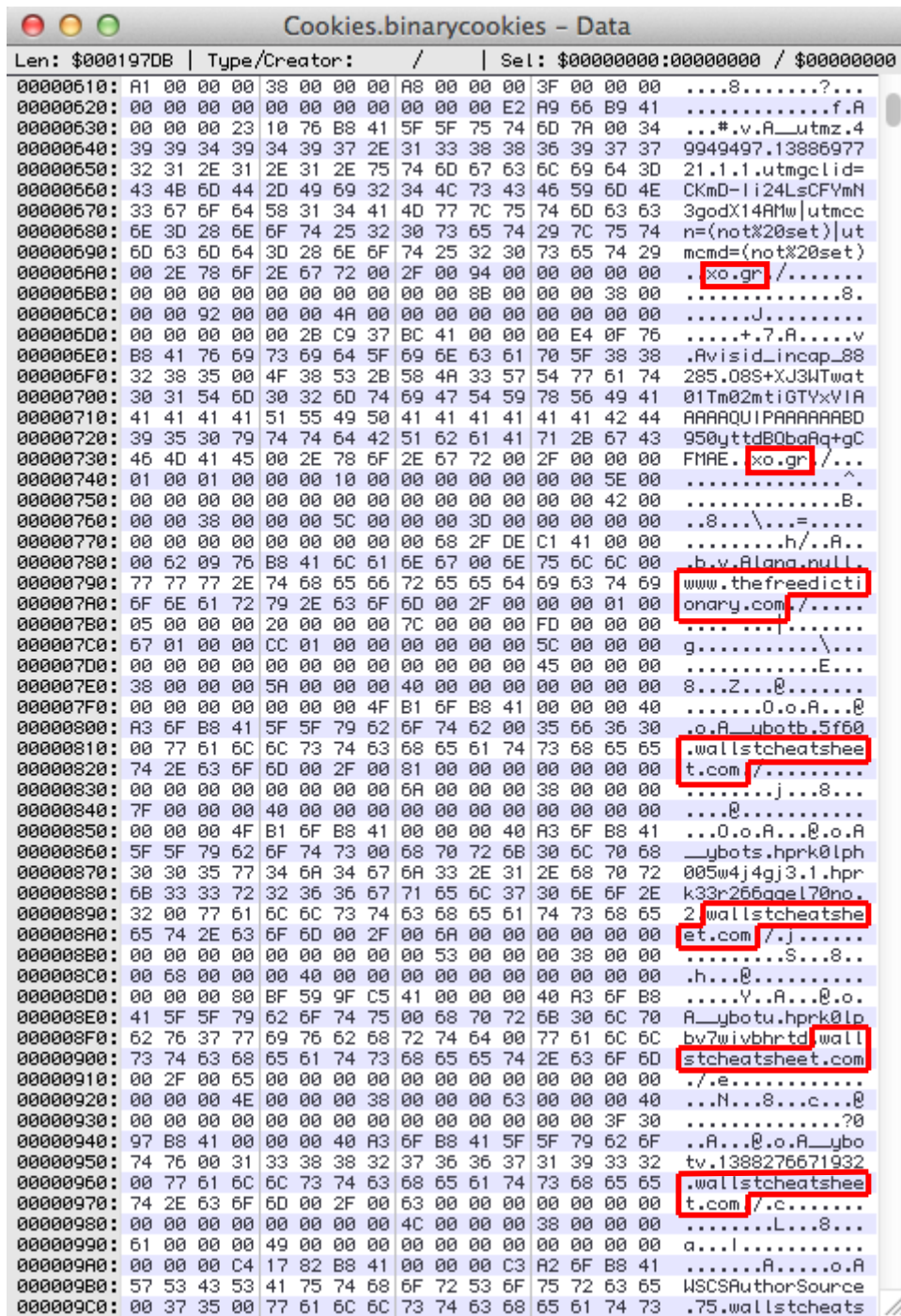


Illustration 19: Some of the contents in the Cookies.binarycookies file opened with a HEX editor

Along with the browser cookies, iOS devices keep logs of the searches that the user makes when surfing in the Safari browser. These logs do exist in the RecentSearches.plist file which is located in the /private/var/mobile/Library/Caches/Safari directory and contains information in XML

format. The following picture shows the stored browser searches inside the RecentSearches.plist file.



```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>RecentSearches</key>
  <array>
    <string>facebook</string>
    <string>google</string>
    <string></string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>India songvideo</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
    <string>facebook</string>
    <string>google</string>
  </array>
</dict>
</plist>
```

Illustration 20: RecentSearches.plist file contents

The call history of an iOS device is in the */private/var/Library/CallHistory* directory, where the call_history.db file exists. This file can provide a lot of information to the analyst about the calls that the user had made from his iPhone. When searching to the “call” table of the file, he will be able to find a lot of information such as telephone numbers, the date that the calls were made, their duration and the country code in which they belong.

ROWID	address	date	duration	flags	id	name	country_code
1	390 306940141940	1388644271		0	4	-1	202
2	391 306940141940	1388644296		0	4	-1	202
3	392 306940141940	1388644330		0	4	-1	202
4	393 306940141940	1388644589		0	4	-1	202
5	394 306940141940	1388646766		0	4	-1	202
6	395 306940141940	1388646787		0	4	-1	202
7	396 018850596544	1388653533		0	5	-1	000
8	397 801850596544	1388653810		0	5	-1	000
9	398 801850596544	1388654018	3300		5	-1	202
10	399 801850596544	1388657339		0	5	-1	202
11	400 801850596544	1388657368		0	5	-1	202
12	401 801850596544	1388657393	537		5	-1	202
13	402 6944718799	1388686174		6	5	-1	202
14	403 69598982795	1388686238		14	5	-1	202
15	404 694461002	1388686426		0	1966085	-1	202
16	405 6943511027	1388686504		0	5	-1	202
17	406 306940141940	1388735486		0	4	-1	202
18	407 801850596544	1388735745		5	5	-1	202
19	408 801013095921	1388735905	33		5	-1	202
20	409 801813095921	1388736120	21		5	-1	202
21	410 801813095921	1388736173		0	5	-1	202
22	411 306940141940	1388736306		0	5	-1	202
23	412 306940141940	1388736317		0	5	-1	202
24	413 306940141940	1388736325		0	5	-1	202

Illustration 21: Call logs information inside the call_history.db file

At last, the most of the iOS applications use Apple’s keychain in order to ensure their password management. In the `/private/var/Keychains` directory, the `keychain-2.db` file resides and comprises accounts and passwords of users in the iOS device. When referring to accounts, this means that there are also email addresses in this file, a thing that proves to be very interesting for a forensics analyst, as emails can be used by someone is numerous occasions. Additionally, SIM card existence can be appeared in this file in conjunction with all the access points from which the device has wirelessly connected to the internet. The analysis of the `keychain-2.db` file is shown in the following pictures.

acct	svce	gena	data	agrp	pdmn
1 Koutsoumpos-wireless	AirPort			apple	dk
2	com.facebook.datr			T84QZ565DQ.platformFamily	ak
3	push.apple.com			com.apple.apsd	dku
4 Wind WiFi 290D60	AirPort			apple	dk
5 Wind WiFi Y3qQDk	AirPort			apple	dk
6 OTE34f56c	AirPort			apple	dk
7 ote2C16F4	AirPort			apple	dk
8 TNCAp418B9F	AirPort			apple	dk
9 NetFasteR IAD 2 (PSTN)	AirPort			apple	dk
10 SpeedTouch216AAC	AirPort			apple	dk
11 Private	com.apple.managedconfiguration			apple	aku
12 SIM_PIN	CommCenter	89300100121206117518		apple	dk

Illustration 22: The contents in the “genp” table of the keychain-2.db file

Table: inet										New Record	Delete Record
	desc	icmt	label	acct	svr	ptcl	atyp	port			
1	Web form password	default	www.androidfreeware.net (bilal qasim1989)	bilal qasim1989	www.androidfreeware.net	http	form		0		
2	Web form password	default	accounts.google.com (rajabashir22@gmail.com)	rajabashir22@gmail.com	accounts.google.com	https	form		0		
3	Web form password	default	m.facebook.com (m.facebook.com)	m.facebook.com	m.facebook.com	https	form		0		
4	Web form password	default	www.iphonellahellas.gr (www.iphonellahellas.gr)	www.iphonellahellas.gr	www.iphonellahellas.gr	http	form		0		

Illustration 23: The contents in the “inet” table of the keychain-2.db file

In iOS keychain, all the items are stored in 4 tables which are genp, inet, cert and keys. Genp table contains generic password keychain items, inet table contains internet password keychain items and cert and keys tables contain certificates, keys and digital identity keychain items.

A lot of information can be extracted from the “inet” table of the keychain-2.db file such as the web sites in which the user has logged in, the username or the email address he uses to log in and the type of protocol the site uses for connection between the client and the server. However, no passwords seem to exist yet inside this file, so physical acquisition is necessary for this type of data extraction.

Passcode bypass

All the above procedure took place in an unlocked iPhone device, without any passcode that would prevent a forensics analyst from gaining access to the file system. However, many users lock their iOS device with a four-digit passcode in order to protect it from unauthorized access. In this case, the only way to gain access is to create and load a custom forensic recovery RAM disk to the device that is booted as if it was a firmware restore. So, rather than actually restoring the iPhone, this procedure installs a recovery payload onto the iPhone’s read-only system partition granting the examiner SSH access to the device and bypassing any passcode security that might exist.

The above mentioned process is automated using a runnable JAR archive. When the iOS device is deactivated and connected to a computer in DFU mode, this Java tool automatically downloads all the required files, patches and sends them to the device. After that, the RAM disk is created and loaded to the iOS device.


```

Extracted resource to /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/
Patched to /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/T/ssh_rd/i
Kernel prepared at /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/T/s
Downloading 038-0029-002.dmg
Downloaded to /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/T/ssh_
Decrypted to /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/T/ssh_rc
Extracted resource to /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/
Added ssh.tar to the ramdisk
Ramdisk prepared at /var/folders/f9/c7c45crs65ngfmzb8_c00g700000gn/T
Preparing to load the ramdisk..
Ramdisk load started!
MobileDevice event: DfuDisconnect, 1227, 12223100
MobileDevice event: RecoveryConnect, 1281, 12803100
MobileDevice event: RecoveryDisconnect, 1281, 12803100
Almost there..
MobileDevice event: MuxConnect, 0, 0

Success!
Connect to localhost on port 2022 with your favorite SSH client!

login: root
password: alpine

```

Illustration 24: Automated RAM disk creation and loading into the iPhone

Opening a terminal window, the forensics analyst connects to localhost on port 2022 as it is prompted by the Java tool, writing the `ssh -p 2022 root@localhost` command. Then, the `ls /` command checks the components of the iPhone's file system and with `mount.sh` all its contents are mounted on `/mnt2` directory.

```

nikosanagnostopoulos — ssh — 80x24
Last login: Sat Feb  8 16:35:21 on ttys000
MacBook-Pro-toutes-Nkos:~ nikosanagnostopoulos$ ssh -p 2022 root@localhost
root@localhost's password:
Use mount.sh script to mount the partitions
Use reboot_bak to reboot
Use 'device_infos' to dump EMF keys (when imaging user volume)
-sh-4.0# ls
-sh-4.0# ls /
System bin dev etc mktar.sh mnt1 mnt2 private sbin usr var
-sh-4.0# mount
/dev/md0 on / (hfs, local, noatime)
devfs on /dev (devfs, local, nobrowse)
-sh-4.0# mount.sh
Checking /dev/disk0s1 ..
** /dev/rdisk0s1
   Executing fsck_hfs (version diskdev_cmds-488.1.7~391).
** Checking non-journaled HFS Plus Volume.
** Detected a case-sensitive volume.
** Checking extents overflow file.
** Checking catalog file.
** Checking multi-linked files.
** Checking catalog hierarchy.

```

Illustration 25: Connection and mounting of the iPhone

Finally, the analyst is able to browse the iPhone's file system with Cyberduck, by connecting to localhost on port 2022.

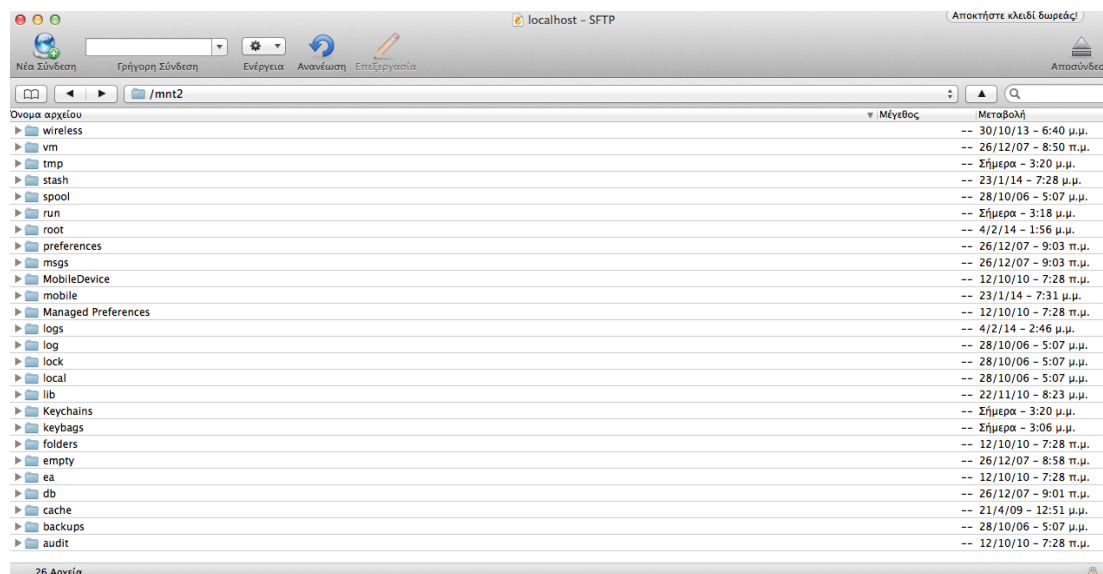


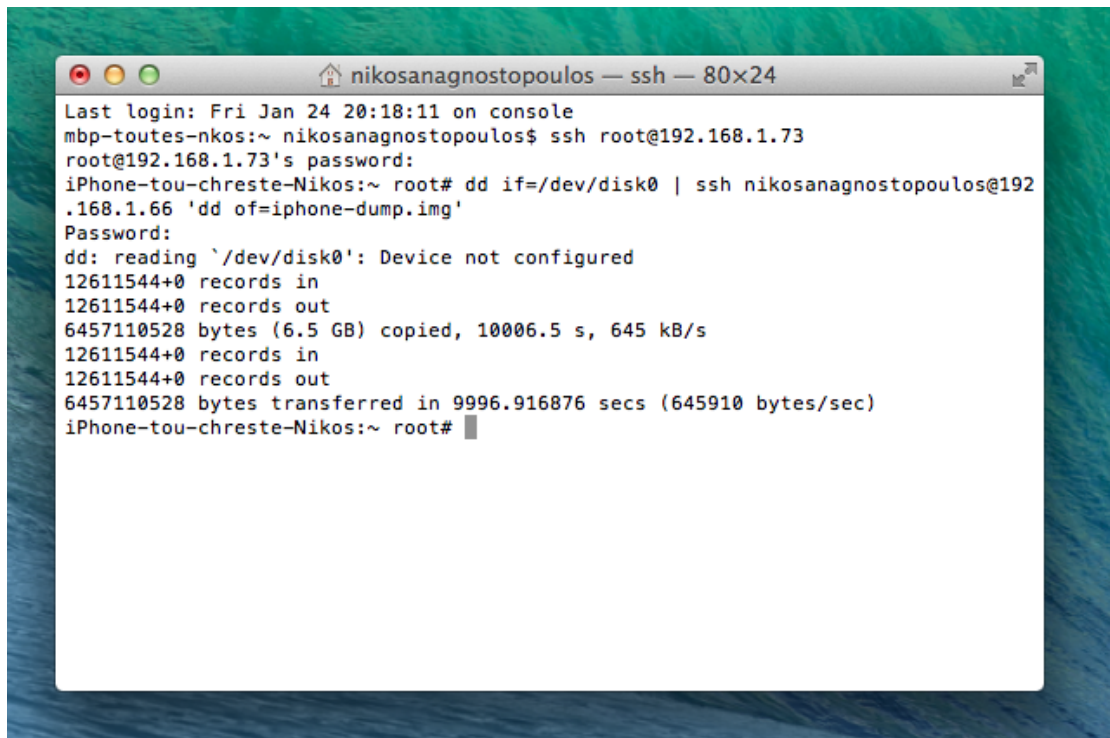
Illustration 26: iPhone's contents mounted on /mnt2 directory

So, it was proved that even a locked with a passcode iPhone can be accessed and all its contents can be browsed as if the device was not protected.

Physical Acquisition

The best way for a forensics analyst to examine an iOS device is to obtain a bit by bit copy of the original media. This method will allow access to data which have previously been deleted and reside in slack space. As mentioned above, since iOS devices can't be mounted to a computer through a USB cable, a professional examiner needs to connect it to the forensics workstation via an SSH connection.

First of all, both the iPhone and the forensics workstation need to be connected to the same wireless network. Then, the IP addresses of the two devices are necessary for the connection between them and the iPhone's memory image acquisition. iPhone's IP address can be found either from the SBSettings application or by its wireless network settings. Furthermore, SSH has to be turned on both devices. On iPhone, this can be easily controlled from the SBSettings application again. On a MacBook Pro computer, SSH can be turned on by selecting Remote Login on System Preferences. Finally, the SSH communication between the two devices is ready to start using the computers' terminal as it is illustrated below.

A screenshot of a terminal window titled "nikosanagnostopoulos — ssh — 80x24". The terminal shows a sequence of commands and their outputs. It starts with a login from a workstation to an iPhone's root folder. Then, a second SSH connection is established from the iPhone to the workstation. Finally, a 'dd' command is used to copy the iPhone's disk image to the workstation. The output shows that 6.5 GB of data was copied at a rate of 645 kB/s over approximately 10 minutes.

```
Last login: Fri Jan 24 20:18:11 on console
mbp-toutes-nkos:~ nikosanagnostopoulos$ ssh root@192.168.1.73
root@192.168.1.73's password:
iPhone-tou-chreste-Nikos:~ root# dd if=/dev/disk0 | ssh nikosanagnostopoulos@192
.168.1.66 'dd of=iphone-dump.img'
Password:
dd: reading `/dev/disk0': Device not configured
12611544+0 records in
12611544+0 records out
6457110528 bytes (6.5 GB) copied, 10006.5 s, 645 kB/s
12611544+0 records in
12611544+0 records out
6457110528 bytes transferred in 9996.916876 secs (645910 bytes/sec)
iPhone-tou-chreste-Nikos:~ root#
```

Illustration 27: SSH connection between iPhone and forensics workstation alongside with memory image extraction

The *ssh root@192.168.1.73* command gives the forensics examiner access to the iPhone's root folder. The iPhone's password which is asked by the system is *alpine*. After that, an SSH connection has been established between the iPhone and the workstation. At last, the *dd if=/dev/disk0 | ssh nikosanagnostopoulos@192.168.1.66 'dd of=iphone-dump.img'* command accesses the iPhone's root folder and instructs it to "dump" its disk image into the forensics workstation, where 192.168.1.73 is the iPhone's IP address and 192.168.1.66 is the computers IP address. This carves out an image of the iPhone's internal storage in all of its dimensions and deposits it into the workstation. The "iphone-dump.img" file is created inside the "downloads" folder and its size is continuously increasing as time passes, as it is shown in the following picture.

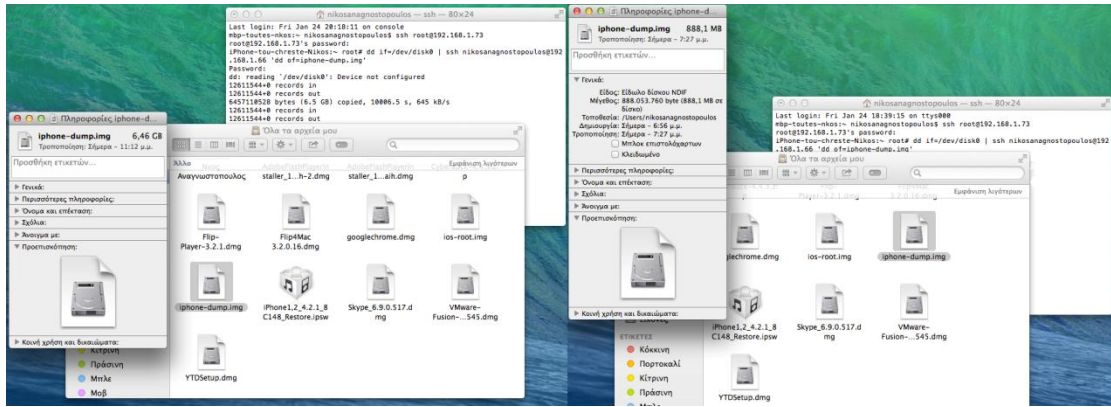


Illustration 28: Left: Completion of the “iphone-dump.img” Right: A different phase of the “iphone-dump.img” downloading to the forensics workstation

The final size of the image file is 6,5GB out of an 8GB iPhone and it allows the examiner to view additional items such as deleted files in unallocated space, using his choice of tools.

Data extraction with Foremost

Foremost is a forensic data recovery program for Linux operating system which is used to recover deleted files using their headers, footers and data structures through a process known as file carving. It is primarily used by government authorities and professional forensics analysts but it is freely available and can be used as a general data recovery tool. Moreover, it is used from the command-line interface and it has no available option for a graphical user interface. It is able to recover specific types of data from image files and so it can be used via a forensics workstation to recover data from iOS devices.

Kali Linux is a Debian-derived Linux distribution designed for digital forensics and penetration testing. It is a newer and updated version of BackTrack and it comes pre-installed with numerous penetration testings programs and forensics tools. Foremost does exist among those tools and it is also pre-installed in the operating system and ready for use.

Once the image file has been transferred in Kali Linux, the examiner can start the data carving process by opening Foremost and typing the following command:

foremost -T -i /root/Desktop/iphone-dump.img

The **-T** flag gives the opportunity to foremost not to care about the type of the image file given by the examiner and carve every possible files from it. The **-i** flag specifies the input file. Then, the **/root/Desktop/iphone-dump.img** constitutes the full path in which the **iphone-dump.img** is located.

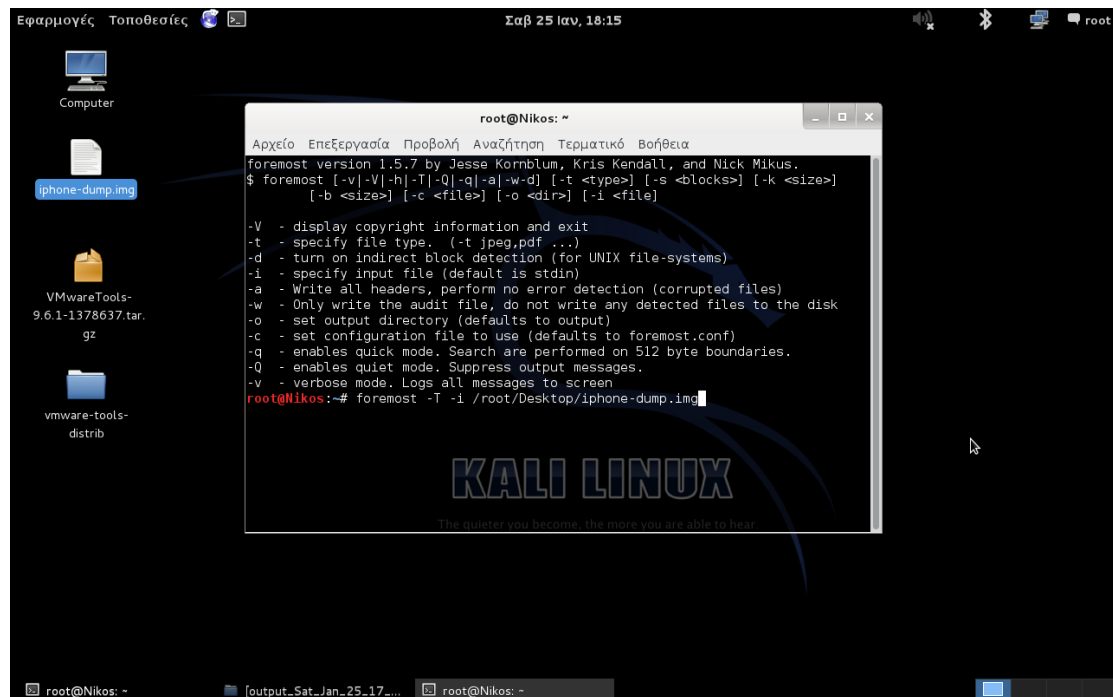


Illustration 29: execution of file carving command through foremost

After the command is executed, foremost starts looking for every possible existing and deleted files inside the iPhones image file.

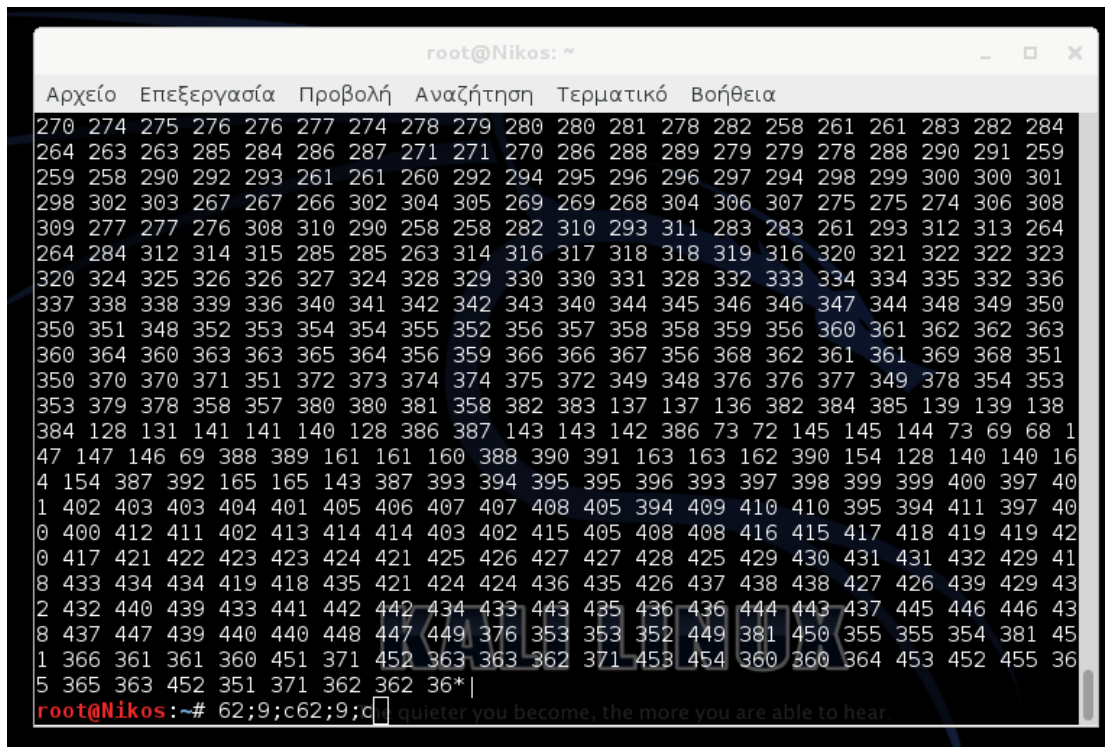


Illustration 30: foremost’s searching process

When the procedure is finished, a new folder is created in the operating system’s home folder, which contains all the recovered data that were found, as it is pictured below.

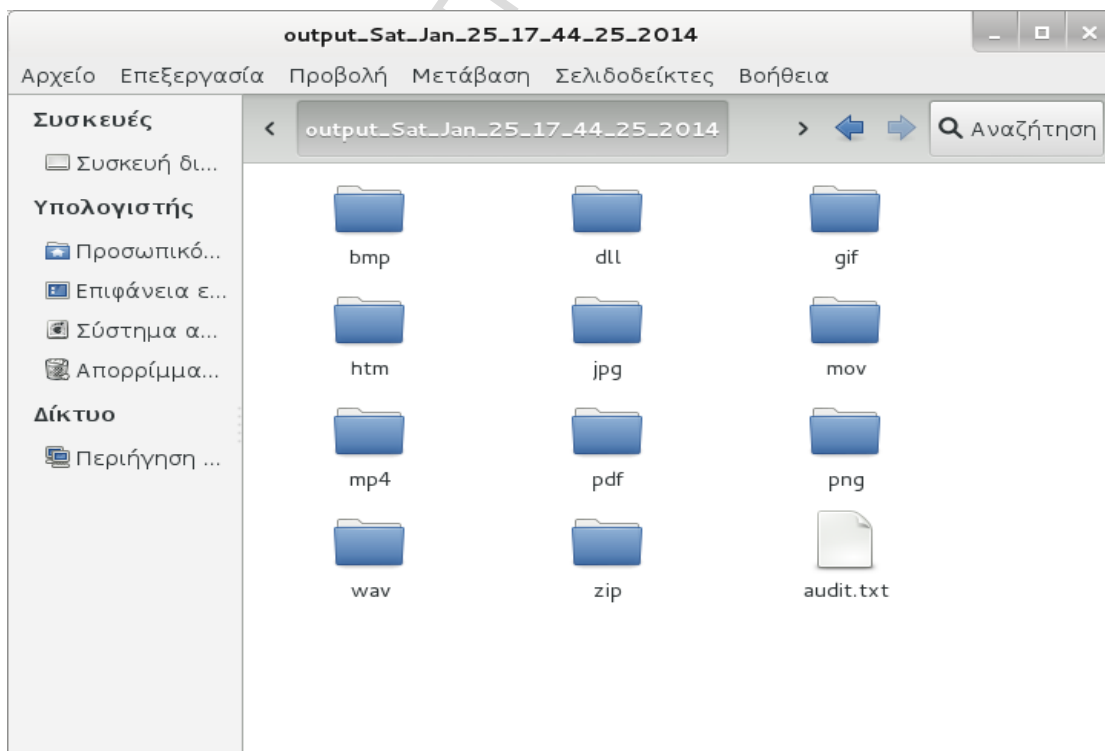


Illustration 31: All the recovered data categorized into different types of folders

As it is shown, 11 types of files were carved from the iPhone's image file with the most important among them being the .jpg files for a forensics analyst. The .jpg folder contains numerous pictures that the previous user had captured with the device's camera or even screenshots. Nevertheless, the drawback is the potential for many false positives such as screenshots of application contents like contacts, messages, notes, etc. At last, this proves to be very important evidence for a forensics examiner in case where no other data can be extracted from the device.

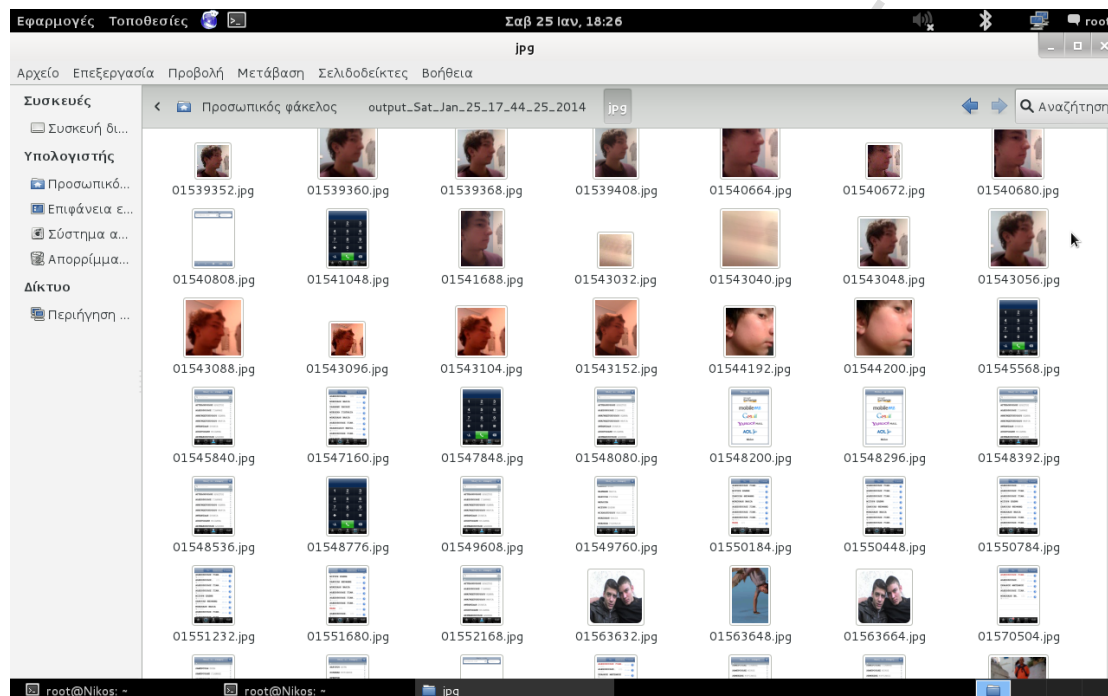


Illustration 32: The .jpg folder containing both photos and device's screenshots

Data extraction with PhotoRec

As explained previously, Foremost is mostly capable of recovering specific types of data including pictures, videos, music, documents, compressed files and internet files. A forensics analyst though, is interested in discovering other files as well such as database files including deleted contacts, messages, call logs, etc. PhotoRec is one of the most advanced data carving tools available in both the open source and the commercial market which is able to recover almost every deleted type of files from a raw disk image, including SQLite database files. It is a command-line program like foremost, so the forensics analyst has to execute some commands in order to specify his preferences.

At first, a terminal window has to be opened and the examiner must be directed to the exact folder in which the “iphone-dump.img” file is located, by typing the exact path to it. After that, the `ls -la` command shows all the contents of the chosen directory with the “iphone-dump.img” file included inside. Then, the `/applications/Recovery/photorec iphone-dump.img` command finds and executes PhotoRec software following the right path, with the specific raw disk image as input.

```

nikosanagnostopoulos — bash — 87x31
Last login: Thu Feb  6 14:28:47 on ttys000
mbp-toutes-nkos:~ nikosanagnostopoulos$ /Users/nikosanagnostopoulos
-bash: /Users/nikosanagnostopoulos: is a directory
mbp-toutes-nkos:~ nikosanagnostopoulos$ ls -la
total 13061784
drwxr-xr-x+ 22 nikosanagnostopoulos  staff          748  6 εβ 14:22 .
drwxr-xr-x   6 root                    admin          204 30 κτ 22:07 ..
-rw-----   1 nikosanagnostopoulos  staff           3 26 εν 19:47 .CFUserTextEncoding
-rw-r--r--@   1 nikosanagnostopoulos  staff        12292  4 εβ 17:38 .DS_Store
drwx-----   9 nikosanagnostopoulos  staff          306  4 εβ 15:11 .Trash
-rw-----   1 nikosanagnostopoulos  staff          869  6 εβ 14:28 .bash_history
drwx-----  20 nikosanagnostopoulos  staff          680  6 εβ 14:08 .dropbox
drwx-----   4 nikosanagnostopoulos  staff          136 27 Ιαυ 13:44 .dropbox-master
-rw-r--r--   1 nikosanagnostopoulos  staff        3265 27 Ιαυ 23:43 .photorec.cfg
drwx-----   3 nikosanagnostopoulos  staff          102 23 Ιαυ 18:15 .ssh
drwx-----+  9 nikosanagnostopoulos  staff          306  6 εβ 14:08 Desktop
drwx-----+  5 nikosanagnostopoulos  staff          170 18 κτ 21:25 Documents
drwx-----+ 61 nikosanagnostopoulos  staff        2074  4 εβ 15:11 Downloads
drwx-----@ 21 nikosanagnostopoulos  staff          714  6 εβ 14:08 Dropbox
drwx-----@ 48 nikosanagnostopoulos  staff        1632 11 Ιαυ 12:12 Library
drwx-----+  6 nikosanagnostopoulos  staff          204  4 εβ 15:20 Movies
drwx-----+  5 nikosanagnostopoulos  staff          170 19 Δεκ 00:39 Music
drwx-----+  6 nikosanagnostopoulos  staff          204 27 Ιαυ 14:55 Pictures
drwxr-xr-x+  4 nikosanagnostopoulos  staff          136 26 εν 19:47 Public
-rw-r--r--   1 nikosanagnostopoulos  staff    230457344 23 Ιαυ 18:27 ios-root.img
-rw-r--r--   1 nikosanagnostopoulos  staff    6457110528 24 Ιαυ 23:12 iphone-dump.img
-rw-r--r--   1 nikosanagnostopoulos  staff     40960  6 εβ 14:22 photorec.ses
mbp-toutes-nkos:~ nikosanagnostopoulos$ /applications/Recovery/photorec iphone-dump.img

```

Illustration 33: Execution of PhotoRec software with iphone’s raw disk image as input

After the execution of the last command, PhotoRec is ready for use and the examiner is prompted to walk through a series of configurations. There is a capability of choosing the file types to be recovered and the place where they will be saved. The program also gives the ability of recovering and keeping the corrupted files. In this specific case, the SQLite database files are of great interest for an examiner, so selecting the carving of such files gives 182 results saved in the previously specified folder.

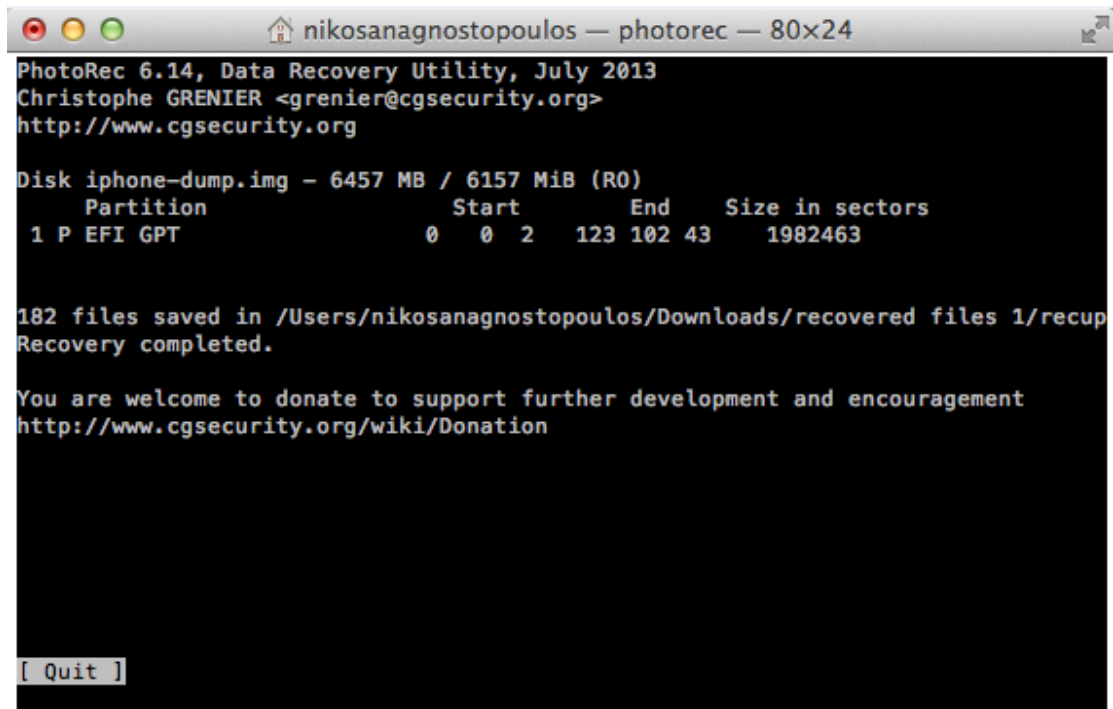


Illustration 34: The results of file carving with PhotoRec

The results for this specific case can be found in the */Users/nikosanagnostopoulos/Downloads/recovered files 1* directory, inside the *recup_dir.1* folder, as it is illustrated below.

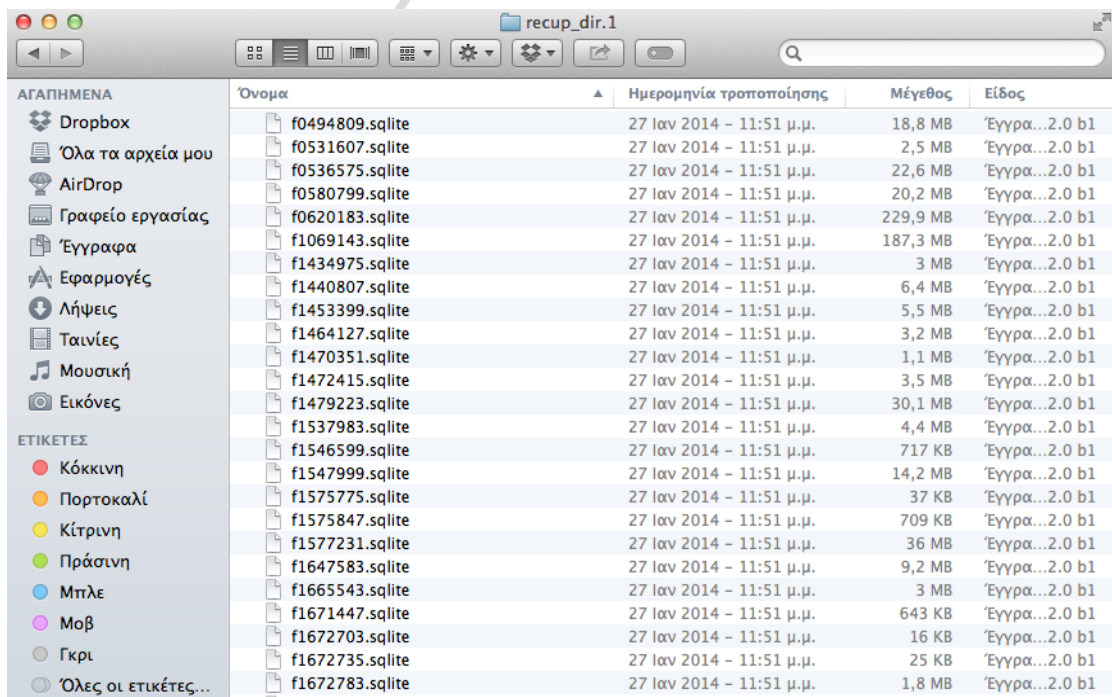


Illustration 35: All the recovered SQLite database files

Searching through the carved files, a forensics examiner can find very interesting evidence such as SMS that had previously been deleted from the iOS device. It seems that physical extraction to the memory storage of the iPhone is able to find more files than logical extraction does.

ROWID	address	date	text	flags	replace	svc_cr	group_id	asso	height	UIFlags	version	subject	country	headers	recipients	read
1	2228	306937170038	1291878779	ΜΝ ΤΟΛΜΗΣΕΙΣ Ν Μ ΕΑΥ		2	0	361	0	0	4	0	gr			1
2	2293	306985757026	1292268305	ΓΙΑΝΝΑΡΕ ΑΥΡΙΟ ΦΕΡΕ Μ		2	0	395	0	0	4	0	gr			1
3	2395	306987700686	1292433333	ΑΝΤΡΑ Μ ΧΙΛΙΑ ΣΟΡΥ...ΕΙ		2	0	422	0	0	4	0	gr			1
4	2396	306987700686	1292433364	ΑΝΤΡΑ Μ ΧΙΛΙΑ ΣΟΡΥ...ΕΙ		2	0	422	0	0	4	0	gr			1
5	2397	306987700686	1292433536	Α οκ... Γυναίκα μ. Απ		3	0	422	0	0	4	0	gr			1
6	2403	306987700686	1292440958	ΕΛΑ ΑΝΤΡΑ Μ...ΤΕΛΕΙΩΣΑ		2	0	422	0	0	4	0	gr			1
7	2404	306987700686	1292441036	Μ... Πως εισαι;		3	0	422	0	0	4	0	gr			1
8	2405	306987700686	1292441323	Ε ΚΑΛΑ...		2	0	422	0	0	4	0	gr			1
9	2406	306987700686	1292441443	Μπράβο. Τερ πες κανα		3	0	422	0	0	4	0	gr			1
10	2407	306987700686	1292442354	Μπράβο. Τερ πες κανα		3	0	422	0	0	4	0	gr			1
11	2408	306987700686	1292443882	ΤΠΤ...ΕΣΥ?		2	0	422	0	0	4	0	gr			1
12	2409	306987700686	1292443967	ΤΠΤ σπλι κ διαβασμα...		3	0	422	0	0	4	0	gr			1
13	2410	306987700686	1292444125	ΤΠΤΤ ΣΑΒΒΑΤΟ ΔΝ ΒΓΗ		2	0	422	0	0	4	0	gr			1
14	2411	306987700686	1292445043	Μ... Σε κτλβενω... (μιλα		3	0	422	0	0	4	0	gr			1
15	2412	306987700686	1292445115	ΝΑΙ ΑΛΛΑ ΛΙΓΟ...ΟΥΤΕ ΠΙ		2	0	422	0	0	4	0	gr			1
16	2413	306987700686	1292445176	Κτλβα... Αν κ αυτο μ ακ		3	0	422	0	0	4	0	gr			1
17	2414	306987700686	1292445235	ΔΔΔ?		2	0	422	0	0	4	0	gr			1
18	2415	306987700686	1292445834	ΤΠΤ αστο δν πειραζει...		3	0	422	0	0	4	0	gr			1
19	2416	306987700686	1292445968	ΠΕΣ Τ ΕΝΝΟΥΣΕΣ...		2	0	422	0	0	4	0	gr			1
20	2417	306987700686	1292446310	ΠΕΣ Τ ΕΝΝΟΥΣΕΣ...		2	0	422	0	0	4	0	gr			1
21	2418	306987700686	1292446828	ΠΕΣ Τ ΕΝΝΟΥΣΕΣ...		2	0	422	0	0	4	0	gr			1
22	2419	306987700686	1292447480	ΤΠΤ : : : : : ηταν...		3	0	422	0	0	4	0	gr			1
23	2420	306987700686	1292447521		2	0	422	0	0	4	0	gr			1
24	2421	306987700686	1292447631 Απλα μ ακουστηκε κ		3	0	422	0	0	4	0	gr			1

Illustration 36: Some of the 332 carved SMS of the iPhone

In conjunction with the text messages, the SQLite file gives the examiner extra information such as the numbers from which the SMS were sent, the data they were sent and if they were read or not by the receiver.

Another important evidence is the notes.sqlite file which now comprises all the deleted notes of the previous user.

ROWID	creation_date	title	summary	contains_cjk	modification_date
1	27	312473091	Καναρης παναγιωτης		312731993
2	28	313018497	Θελω κ παθενω		314639904
3	29	314192662	Αντιο επανασταση	Ζαρια με φοντο κοκκινο	314192679
4	31	314750374	Star	Αγγελιες	314750521

Illustration 37: Contents of the notes.sqlite file

This file includes all the contents of the notes alongside with the date they were created and the date they were last modified.

Moreover, all the Safari bookmarks are saved in a SQLite database file. This proves to be a necessary clue for an investigator because it lets him learn a lot of things about the web sites the user used to visit while having the iPhone device. All the URL

addresses of the saved bookmarks are contained into this file as it is illustrated in the following picture.

	id	special_id	parent	type	title	url	num_children	editable	deletable
1	0	0		1	Root		4	1	1
2	2	0	0	0	Ματρώνα Τεχνολογία	http://www.xmpro.com./vid	0	1	1
3	3	0	0	0	Επιδοχές Τεχνολογία	http://www.xmpro.com./vid	0	1	1
4	4	0	0	0	Untitled	http://www.google.com.gr	0	1	1
5	5	0	0	0	Επιδοχές Τεχνολογία	http://www.xmpro.com./vid	0	1	1

Illustration 38: Some of the contents into the Safari bookmarks database file

The most important database file that can be found inside the raw disk image is the keychain-2.db file. This was found previously performing a logical acquisition to an iPhone. However, physical acquisition helps an examiner to find more information about deleted accounts of the user’s iPhone.

	acct	svce	gena	data	agrp
1	ONTelecoms	AirPort		r	apple
2	kostasWiFi	AirPort		V+	apple
3	FEATHER	AirPort		dF; ^ RC	apple
4	Marias Place	AirPort		v A H	apple
5	ultimate_gr@hotmail.com	com.apple.itunesstored.keychain		z \	apple
6	SIM_PIN	CommCenter	725660062	C	apple

Illustration 39: Contents of the “genp” table inside the keychain-2.db file

	acct	sdmn	srvr	ptcl	atyp	port	data	agrp
1	spyrou.george@gmail.com		imap.gmail.com	imap		143	\$ < sE	apple
2	spyrou.george@gmail.com		smtp.gmail.com	smtp		25	~ 0 6Q	apple
3	El greco		ikariam.com	http	form	0	e & w	apple
4	giorgos1993		www.cosmote.gr	https	form	0	> \$	apple
5	ultimate_gr@hotmail.com		www.facebook.com	http	form	0	7 #	apple
6	spyrou.george@gmail.com		www.google.com	https	form	0	m d }	apple
7	giorgos1993		www.cosmote.gr	https	form	443	bl O	apple

Illustration 40: Contents of the “inet” table inside the keychain-2.db file

The “genp” table inside the keychain-2.db file contains all the access points from which the iPhone had previously connected to the internet alongside with the device’s SIM existence. The “inet” table contains all the accounts and email addresses that the user had in order to log in to different web sites. In both tables, passwords do exist in “data” field but they are all encrypted by Apple’s keychain and can’t be parsed unless

the appropriate commercial software is available, such as iPhone Password Breaker from Elcomsoft.

In conclusion, the contacts and the call history information appear to be exactly the same with the ones extracted from the iPhone performing the logical acquisition method.

iOS Anti-Forensics

Anti-Forensics are actions which goal is to prevent proper forensic investigation process or make it much harder and therefore, more expensive. These actions are aimed at preventing access in a device and at reducing quantity and quality of digital evidence. Anti-Forensic techniques can be used to increase the security of a device in order to protect its data from unauthorized access. However, a lot of perspectives do exist about anti-forensics including both malicious intents and creative reasons for new forensic methods and tools. As regards iOS devices, there are some anti-forensic and security methods which will be introduced below.

First of all, the main way for securing an iOS device is activating the “Auto-Lock” and then the “Passcode Lock” as well. For better security levels, long passwords with letters, numbers and special characters have to be activated, so as for a forensic analyst not to be able to perform an easy brute force attack to discover it. Additionally, location services have to be turned off to all third-party applications for preventing unauthorized access to the device’s location. Wireless networks and 3G connection to the Internet has to be turned off as well, for blocking a possible connection of the device with an unknown network. Furthermore, iTunes Backups must be encrypted using long passwords with letters, numbers and special characters, as they may contain a completed image of an iOS device’s saved data. In case that the iOS device is jailbroken, SSH network protocol has to be generally disabled and its default password and port must be changed, as SSH connection is the most standard way for connecting an iOS device to a workstation.

Apart from those primary ways of protecting the data of an iOS device, there are also other special methods and applications for achieving high levels of security. Data encryption and obfuscation are very good methods for preventing an analyst from retrieving and reading data. Some applications such as Picture Safe and Picture Vault

offer the ability to encrypt images on the device and lock these files with strong passwords. Data hiding is another anti-forensics technique which great advantage is the availability of the data when there is need. Incognito Web Browser and Invisible Browser are two similar applications that allow a user not to cache any web browser history. However, the history may not be left behind, but the cookies.plist file does exist yet containing domain information. Moreover, data deletion is one of the most important anti-forensic techniques because it makes it impossible for an examiner to recover deleted data. iErase is a utility that overwrites the free space on an iOS device and tigertext is an application which completely deletes all the communication between the sender and the receiver after its completion. Finally, data forgery is also a practice aimed at avoiding the identification of incriminating material by changing file extensions or by significantly falsifying the true nature of information.

Conclusions

To sum up, iOS Forensics is a fairly new field of study which continuously improves as iOS devices evolve. There are a lot of methods for conducting a forensics investigation which depend on the iOS device itself and its software version. A technique for acquiring an iPhone 2 image does not necessarily suffice for acquiring an iPhone 5 or 5s image as iOS version 3 can have different security methods than iOS version 7 which is the latest. Furthermore, newer iPhone models such as iPhone 3Gs and later, are equipped with encryption hardware (AES coprocessor). So, changing security models on the iOS device can keep an examiner from extracting a forensically sound image until a new technique or tool is developed to grant privileged access.

So far, the most popular methods for retrieving and analyzing both existing and deleted data from an iOS device are logical and physical acquisition. Logical acquisition does not usually recover many deleted data but this method helps a forensics examiner to organize all the discovered data from iPhone's file system and databases. On the other edge, physical acquisition appears to have a great advantage as it is able to recover deleted data that reside in memory's slack space, which is the main difference from logical acquisition. However, it seems that neither of them is able of finding all the necessary evidence for a forensics examiner without each other.

The right combination of those two methods is the best way of recovering the desirable amount of information from an iPhone.

Alongside with those extraction methods, there are also some other techniques such as acquisition from iTunes backups and iCloud. With those methods, an investigator is able of retrieving evidence in case that the iOS device is not available but a backup of it does exist on the user's workstation. Nevertheless, RAM data acquisition methods on iOS devices do not exist yet and constitute a strong motivation for future work, as RAM may contain important evidence such as users' passwords in plaintext without being encrypted.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

References

1. Sean Morrissey, iOS Forensic Analysis for iPhone, iPad and iPod touch, SANS Institute, 2010.
2. Jonathan Zdziarski, iOS Forensic Investigative Methods, Technical Draft, 5/13/12.
3. iPhone Analyzer User Guide, Forensically examining an iPhone or iOS device, Cruptic Bit.
4. iPhone 3GS Forensics: Logical analysis using Apple iTunes Backup Utility, Mona Bader, Ibrahim Baggili, PhD, Advanced Cyber Forensics Research Laboratory, Zayed University, September 2010.
5. Mac OS X, iPod, and iPhone Forensic Analysis DVD Toolkit, Jesse Varsalone, Ryan R. Kubasiak, Sean Morrissey, Walter Barr, James “Kelly” Brown, Max Caceres, Mike Chasman, James Cornell, SYNGRESS, 2009.
6. iPhone and iOS Forensics, Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices, Andrew Hoog, Katie Strzempka, SYNGRESS, 2011.
7. Forensic Analysis on iOS Devices, Tim Proffitt, SANS Institute, InfoSec Reading Room, November 5th 2012.
8. Handling iOS encryption in a forensic investigation, Jochem van Kerkwijk, Universiteit Van Amsterdam, System and Network Engineering, July 19, 2011, Final version, rev. 2.
9. OS X and iOS Kernel Programming, Master kernel programming for efficiency and performance, Ole Henry Halvorsen, Douglas Clarke, APRESS.
10. IOS, forensicswiki,
<http://www.forensicswiki.org/wiki/IOS>
11. iPhone Forensics, Infosec Institute,
<http://resources.infosecinstitute.com/iphone-forensics/>
12. Digital forensics, Wikipedia,
http://en.wikipedia.org/wiki/Digital_forensics
13. Mobile device forensics, Wikipedia,
http://en.wikipedia.org/wiki/Mobile_device_forensics
14. What is Live Forensics? MacForensicsLab,

- http://www.macforensicslab.com/ProductsAndServices/index.php?main_page=document_general_info&cPath=5_24&products_id=212
15. History of the iPhone, Wikipedia,
http://en.wikipedia.org/wiki/History_of_the_iPhone
 16. iOS version history, Wikipedia,
http://en.wikipedia.org/wiki/iOS_version_history#iPhone_OS_2.x
 17. iOS, theiphonewiki,
<http://theiphonewiki.com/wiki/iOS>
 18. iOS versions, Definition from PC Magazine Encyclopedia,
<http://www.pcmag.com/encyclopedia/term/63292/ios-versions>
 19. iOS SDK, Wikipedia,
http://en.wikipedia.org/wiki/iOS_SDK
 20. HFS Plus, Wikipedia,
http://en.wikipedia.org/wiki/HFS_Plus
 21. iPhone Forensics, Rahil Parikh's Blog,
<http://bits.rahilparikh.me/2012/09/12/iphone-forensics/>
 22. Jailbreak, StackExchange, Where is the dividing line (mount point) of the two iOS partitions,
<http://apple.stackexchange.com/questions/45123/where-is-the-dividing-line-mount-point-of-the-two-ios-partitions>
 23. Mobile Device Forensics, Cell Phone Forensic Tips, Tricks and Tutorials, September 17, 2008,
<http://mobileforensics.wordpress.com/2008/09/17/iphone-forensics-a-series-2/>
 24. About SQLite, SQLite,
<http://www.sqlite.org/about.html>
 25. About Property Lists, Mac Developer Library, Apple Inc,
<https://developer.apple.com/library/mac/documentation/cocoa/conceptual/PropertyLists/AboutPropertyLists/AboutPropertyLists.html>
 26. iOS jailbreaking, Wikipedia,
http://en.wikipedia.org/wiki/iOS_jailbreaking
 27. iPhone Jailbreak Tips-How To Connect to Your iPhone via SSH, PatrickJ, posted on April 28th 2011, iSource,
<http://isource.com/2011/04/28/iphone-jailbreak-tips-how-to-connect-to-your-iphone-via-ssh/>

28. Recovering Lost/Deleted Data From an iPhone That Has Not Been Backed Up, modmyi.com,
<http://modmyi.com/forums/file-mods/662961-recovering-lost-deleted-data-iphone-has-not-been-backed-up.html>
29. Guide: Recover Lost or Deleted Photos/Giles on iPhone 3G/S (Jailbroken) on a Mac, modmyi.com,
<http://modmyi.com/forums/file-mods/696196-guide-recover-lost-deleted-photos-files-iphone-3g-s-jailbroken-mac.html>
30. Are Deleted Files Completely Erased? Webopedia, Posted June 24 2010,
http://www.webopedia.com/DidYouKnow/Hardware_Software/Erasing_Deleted_Files.asp
31. Secure Shell, Wikipedia,
http://en.wikipedia.org/wiki/Secure_Shell
32. iPhone Forensics White Paper, ViaForensics advancing mobile security, Zdziarski,
<https://viaforensics.com/resources/white-papers/iphone-forensics/zdziarski/>
33. Automatic SSH ramdisk creation and loading, Mostly iPhone hacking, Wednesday January 11 2012, <http://msftguy.blogspot.gr/2012/01/automatic-ssh-ramdisk-creation-and.html>
34. iDevice Anti-Forensics, Technical Support Canada,
<http://www.technicalsupport.ca/2012/01/idevice-anti-forensics-2/>
35. Anti-Forensics Overview, Computer Forensics and Anti-Forensics Research,
<http://www.forensics-research.com/index.php/anti-forensics/#anti-forensics-overview>
36. Anti-Forensics – Part 1, INFOSEC INSTITUTE,
<http://resources.infosecinstitute.com/anti-forensics-part-1/>