DIGITAL FORENSIC ANALYSIS OF IPHONE

By

Waqas Ali Khan

Thesis submitted to the Faculty of the Department of Information Security, College of Telecommunications (MCS), National University of Sciences and Technology, Pakistan, in partial fulfillment for the requirements of MS degree in Information Security

February, 2012

ABSTRACT

iPhone is always under the scrutiny of mobile and computer hackers because of its popularity and the diverse set of tasks it is able to perform. However, the purpose of the scrutiny is always to reveal the internal functionality of the iPhone and to circumvent the protection mechanism. No comprehensive work has been done in the forensic investigation of the iPhone.

The usage of smart phones in criminal cases is not uncommon phenomenon and several case studies show that criminals are using the easily available encryption and steganography tools built in these smart phones for criminal purposes and for hiding the evidence. Smart phones are also reportedly used in computer and wireless hacking and cracking incidents. Security researchers regularly demonstrate usage of all the popular hacking tools through iPhone in different security conferences. This indicates the way network and other sensitive infrastructure will be attacked by people in the near future and the reason that the attach methods are available to everyone having an Internet connection makes it even more dangerous. Using small devices that can be hidden inside a pocket to launch sophisticated attacks against organizational infrastructure, it is possible to compromise the security system of an organization sitting inside the same organization without anyone's notice.

The main focus of this research is to analyze iPhone for the purpose of forensic investigation. We will design tools and forensic procedures that can extract and analyze data stored on the iPhone. The procedures outlined in this research can work on any iPhone firmware models, from version 3.1.3 to the latest firmware version 4.3.0 with little or no modification to the underlying code of the investigation toolkit. Further, architecture of the toolkit is such that no component is dependent on another. All components work in isolation; therefore, adding additional functionality is not a problem.

All research work and findings are tested on iPhone firmware version 3.1.3. Validation and verification of the findings clearly demonstrated the effectiveness of the procedures and can be reviewed and analyzed independently through the custom developed open source forensic investigation toolkit that is also part of this research activity.

DEDICATION

All praise and thanks to Almighty Allah, the most gracious and the most merciful, Master of the Day of Judgment. Guide us with courage and right path, path of those to whom You have bestowed your blessings.

Dedicated to my beloved teachers, friends, and family.

ACKNOWLEDGEMENT

I would like to thank my supervisor, Dr. Faheem Arif, Associate Professor at the Department of Computer Software Engineering, Military College of Signals, National University of Sciences and Technology (NUST), for his continuous encouragement and able guidance. Without his support and guidance this thesis could not be completed in time.

I am also thankful for the guidance of my co-supervisor Lec. Ahmad Raza Cheema, and to my committee members Lec. Asad Raza, and Lec. Ayesha Noreen of Department of Information Security for their able guidance and support.

Table of Contents

Introduction 1

1.1 Background 1
1.2 Statement of Problem 2
1.3 Objective 3
1.4 Research Methodologies and Goals 3
1.5 Document Organization 3

Forensic Investigation Methodology 5

2.1 Introduction 5
2.2 General Guideline for Forensic Investigation 5
2.3 Forensic Investigation Methodology for Smart Phones 6
2.3.1 Seizing Evidence 8
2.3.2 Acquiring Investigation Data 9
2.3.3 Logical Extraction 10
2.3.4 Physical Extraction 11
2.3.5 Analyzing Evidence 12
2.3.6 Reporting the Findings 13
2.4 Conclusion 14

iPhone File System Architecture 15

3.1 Introduction 15
3.2 Architecture 15
3.2.1 Unsigned Code Execution 16
3.2.2 Complete Control of the Device 16
3.2.3 Work with Unsupported Networks 17
3.3 Conclusion 18

Implementation of the Toolkit 19

4.1 Introduction 19
4.2 Toolkit Structure 19
4.3 Gaining Root Privilege 20
4.4 Logical Acquisition 21
4.5 Physical Acquisition 22
4.6 Data Carving 23
4.7 Conclusion 24

Analysis of Forensic Evidence 25

5.1 Introduction 25

5.2 Forensic Evidence 25

5.3 Conclusion 30

Evaluation of the Toolkit in Real World Case Studies 31

6.1 Introduction 31

6.2 Traditional Crimes Investigation 31

6.3 Hacking and Cracking Investigation 33

6.4 Conclusion 36

Conclusion and Future Work 37

7.1 Overview 37

7.2 Objectives Achieved 37

7.3 Future Work 38

7.4 Summary 38

Bibliography 40

Appendix A 42

View Assigned IP Address 42

Appendix B 44

Jailbreaking iPhone 44

Appendix C 47

Check the default SSH port 47

Appendix D 48

Check OpenSSH Server through Cydia 48

Appendix E 49

Installing OpenSSH on iPhone 49

Appendix F 50

Resetting OpenSSH Password 50

List of Figures

Figure 2.1 iPhone Sync Data on Windows 7 09

Figure 2.2 Logical Acquisition 11

Figure 2.3 Physical Acquisition 12

Figure 7.1 iPhone Used as a Hacking Device 42

Figure A.1 View IP address in iPhone 26

Figure B.1 iPhone Jailbreaking Process 28

List of Tables

Table 4.1 Logical file recovery 22

Table 4.2 Recommended file sizes 23

Table 6.1 Evidence files on iPhone 34

Chapter 1

Introduction

1.1 Background

Digital forensics is a domain of forensic sciences in which digital equipments are analyzed for recovering forensically important data which can help in conducting investigation of a crime. As large number of devices is introduced in the market, the need to develop hardware and software capable of analyzing and recovering data from these devices is becoming ever more important.

Today, mobile technology is used everywhere. Laptops, mobile phones, motor cars, wrist watches, and even many household appliances such as microwave ovens and refrigerators are using wireless technology for their operation. The latest to these long list of items is a smart phone which apart from performing the traditional jobs of a mobile phone such as phone call and Short Messaging Service (SMS) is capable of storing and processing large amount of data, audio/video recording, word processing, performing complex mathematical operations in spread sheet, reading PDF documents, surfing the Internet, sending and receiving emails, and many other tasks that were until recently associated with a computer system.

iPhone is one of the most widely used smart phone having a firmware operating system based on Mac OS X which is a variant of FreeBSD and is commonly categorized as Unix-like operating system. Mac OS X itself has been developed by Apple for its Mac family of computers. The difference between Mac OS X and a common Unix or Linux operating system is the proprietary file system and kernel enhancements as well as the graphical user interface that Apple Inc. has developed. However, since iPhone operating system is developed on top of a Unix kernel, in spite of the fact that Apple Inc. has locked down the OS, still majority of the functions that can be performed by a Unix or Linux OS can also be performed by the iPhone OS as well.

Logically, iPhone has two partitions: one for storing the OS specific files such as kernel images and default programs. The other partition is used for the storage of user specific settings, applications, and data.

From a forensic point of view, the second partition is more important because it contain majority of the functions the user performs on the iPhone and the data for those functions. For example, call history, Short Messaging Service (SMS) messages, contact list, emails, audio and video, pictures taken through the built-in camera, and configuration and setting files of all the applications installed by the user from App Store or Cydia are all located on the second partition.

The main hurdle in the forensic investigation of these devices is the proprietary nature of hardware and software used by these devices.

iPhone is always under the scrutiny of mobile and computer hackers because of its popularity and the diverse set of tasks it is able to perform. However, the purpose of the scrutiny is always to reveal the internal functionality of the iPhone and to circumvent the protection mechanism and to investigate classic criminal cases [9] [10] [11]. No comprehensive work is done in the forensic investigation of the iPhone. Companies have developed complex, expensive, and proprietary investigation tools and frameworks [12]. However, because of the closed-source nature of those tools, no independent peer review can be done on those to determine whether they are using forensically secure methodologies for investigation or not.

1.2 Statement of Problem

Retrieving forensic data from storage media is the most important and primary objective of forensic investigation. So far, the forensic research carried is solely dependent on investigating traditional crimes where smart phones are used as a medium which contain the evidence and the focus is to retrieve phone book record and SMS etc. Since the upcoming smart phones such as iPhone and Android is based on Linux or Unix, these devices can also be used to launch sophisticated attacks such as network hacking and exploitation. The proposed research work will develop forensic investigation methodology for recovering evidence from embedded devices. The methodology can be used on all form of embedded devices with little or no modification. Further, the research will use the forensic investigation methodology for the secure digital forensic investigation of the market leading smart phone called iPhone.

1.3 Objective

Objectives of the thesis are:

1. To analyze the architecture of iPhone in order to examine the file system, and to determine where important user files such as call history, phone book record, messages sent and received, calendar schedules etc.
2. Develop a forensic investigation methodology which can be applied to the investigation of all smart phones, particularly iPhone.
3. Develop a mechanism to retrieve the forensically important data in a complete and accurate manner to maintain its integrity.
4. Design and develop a forensic analysis toolkit for recovering and analyzing evidence from iPhone.

1.4 Research Methodologies and Goals

The research was divided into four phases. In phase one, the architecture of iPhone was thoroughly analyzed in order to understand the file structure and to determine where important data is saved on the device. In phase two, a forensic investigation methodology was developed for the secure forensic investigation of embedded devices. The methodology is generic; therefore, it can be used for all embedded devices. In phase three, focus of the research was on the design and development of a forensic investigation toolkit which can recover all the data from the iPhone. The toolkit is able to not only recover data stored on the iPhone file system but also recover the data deleted by the user. Final phase of the research was evaluation of the toolkit in investigation of real world scenarios and case studies.

1.5 Document Organization

The thesis report is organized in seven chapters. In the second chapter, the forensic investigation methodology is discussed in detail. Chapter three focuses on iPhone architecture and the security mechanisms that are implemented in the iPhone. Chapter four discusses the forensic investigation toolkit that is developed to extract the forensic evidence data from the iPhone. Chapter five is a chapter which contains step by step procedures for performing different helping task that the forensic investigator will require to perform on the iPhone for extraction of evidence. Chapter six is analysis of the forensic evidence and the purpose and importance of each file needed

for forensic investigation. The last chapter is the evaluation of the forensic investigation methodology and toolkit for the investigation of real world scenarios and crime cases.

Chapter 2

Forensic Investigation Methodology

2.1 Introduction

This chapter provides a detailed overview of custom developed forensic methodology specifically developed for smart phones and other embedded devices. Since smart phones are considerably different devices from traditional desktop computers, the nature for forensic investigation for these devices is also different. Applying tools and techniques that are normally used in the computer forensic investigation field on smart phone forensics will result in loss of critical evidence and loss of data. Therefore, a comprehensive forensic methodology has been developed that can provide a reference guide for any forensic investigator performing investigation on smart phones and other embedded devices.

2.2 General Guideline for Forensic Investigation

The Association of Chief Police Officers (ACPO) Good Practice Guide for computer based electronic evidence suggests four principles while dealing with digital evidence:

1. No action performed by the forensics investigator or the tool used for the forensics investigation should change the evidence or temper the integrity of the evidence.
2. Individuals assessing the data must be competent
3. Audit trail of process applied for later replication by a third party must be preserved.
4. Person in charge of the investigation is responsible to ensure the above guidelines are followed.

Although the above guidelines were developed for evidence found on computers, it can also be applied for digital investigation of smart phones. The first point is related to preserving the integrity of the evidence. The primary purpose of any digital forensic investigation is to gather evidence from the seized device or media. Number of times because of lake of knowledge or exposure of the forensic investigator, unintentional damage is done to the digital evidence. This is especially true in case of

dealing with proprietary devices such as iPhone and BlackBerry. Therefore, every effort must be made to ensure that the evidence is not tempered either by the investigator or by the forensic investigation tool. The second point relates directly to the technical and forensic competency of the investigator. Since most of the smart phone devices are proprietary (both hardware and software), some of the key components necessary for forensic investigation such as file systems and network protocols are not published and remain closed source. Therefore, the person responsible for retrieving and analyzing evidence from such devices should be competent in areas such as reverse engineering and protocol analysis. In case of smart phones and other closed source devices, a general idea and technical competency can be acquired even if the device components are not formally published by the vendor. For example, in case of iPhone, it is known that it is using a Unix kernel on top of proprietary hardware. Therefore, the forensic investigator should be trained on the techniques and tools used for Unix kernel investigations so that a general idea for the working of the device can be acquired. Third point is necessary to prove that the process used by the forensic investigation team can be repeated by an impartial third party. This is a necessary scientific principle used in every branch of science to prove the maturity of the process. If a certain technique cannot be repeated again, its scientific value in the community and its legal authority cannot be established. The last point in the guideline is to establish a proper authority so that the above guidelines can be followed.

2.3 Forensic Investigation Methodology for Smart Phones

The first thing the forensic investigator needs is a desktop or laptop machine. The advantage of laptop over a desktop is that it is easy to carry and can be used in the field as well. Sometimes evidence found on a crime scene is stored in such a manner that it cannot be brought back to the investigation lab. For example, a server located in a remote location outside the country jurisdiction and boundary might not be possible to be physically seized and brought to the investigation lab. In case of smart phones or other embedded devices, the investigator can face a situation where the warrant may not allow disturbance of the crime scene and it may not be possible to take the smart phone back to the lab. In such a case, laptop can be used to acquire evidence on site. The other advantage of laptop over a desktop computer is the constant supply of power. Sometimes, if a sudden power failure occurs and no

alternate source of power is available, the desktop system will shut down and the data stored on it can become damage or corrupted. Therefore, the forensic investigator should have a laptop for investigation of digital data. Some of the requirements for the forensic investigation kit are:

1. A desktop or laptop machine
2. USB 2.0 or 3.0 port for fast data transfer rate on the Universal Serial Bus
3. Common operating systems such as Windows and Linux. This is necessary requirement because it is a proven fact that if a smart phone is based on Linux, investigating it using a Linux OS on the host is easier than a Windows operating system. This is because common file systems found on a Linux operating system such as EXT2, EXT3, RAISUREFS, etc can be mounted directly on another Linux machine. In case of Windows, we need to use third party tools, which, if proper care is not taken can damage the evidence. This requirement can be easily fulfilled using separate virtual machines for each operating system
4. Connection cables for each of the popular smart phone devices used at that time.
5. Software drivers needed to communicate to the smart phone
6. A Digital Video Recorder (DVR). This is because of the proprietary nature of the smart phones. In case of a non jailbreak iPhone, if the forensic investigator wants to examine the phone without jailbreaking it first, the only option the forensic investigator has is to manually navigate the file system and use a DVR to capture the screen shots and other evidence. Therefore, a video recorder is very necessary.
7. A signal disruption bag or more commonly known as a Faraday's bag. This is used to disrupt the flow of signal to the device
8. A software or hardware write blocker
9. Forensics tools in case available for the target phone. If a forensic toolkit is not available, use a management tool commonly available through the vendor's site
10. All firmware versions released by vendor of the smart phone.
11. Prior to working in the field, the forensic investigator needs to study the software and hardware architecture of those phones which he is expected to be working on in the field. It is necessary to gain hands on experience in a lab before applying those techniques on actual devices.

The forensic investigation methodology for smart phones and embedded devices are classified into the following phases:

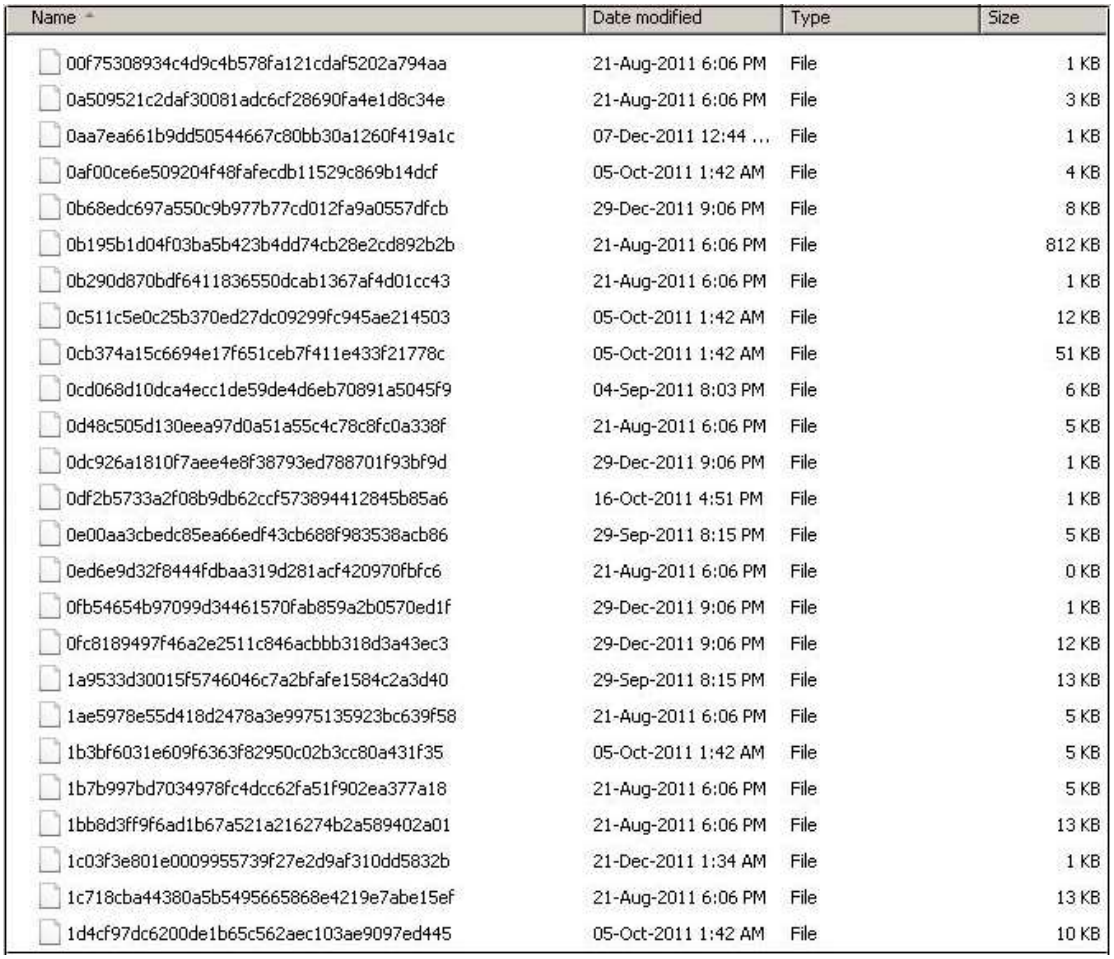
2.3.1 Seizing Evidence

The first step in the forensic investigation of smart phone is to seize the evidence. In this case, the person(s) responsible for the seizure of evidence needs to have proper training in this field. Some general guidelines necessary for seizing a smart phone device are:

1. Identify the device. This will help selecting proper cables and other accessories for data acquisition later. To identify a particular phone, visit http://www.phonescoop.com/phones/finder.php.
2. Do not turn off the device. Restarting the phone may activate some authentication mechanism such as a pass code that cannot be bypassed easily or may damage the evidence in the process of bypassing.
3. Every effort should be made to maintain the integrity of the evidence.
4. Prepare for the worst case such as hardware and software damage to the evidence.
5. All possible avenues which can inflict harm on the phone and its data should be considered and proper mitigation plan should be devised for each one of them. Some of the sources that can harm a smart phone and damage its evidential data are the wireless communication, power supply, and improper training of the person seizing the evidence.
6. Seize all the accessories and supporting documents and manuals found on the crime scene which are related to the smart phone. Often times connection cables, software drivers etc are lying on the crime scene. These can be very helpful for the investigation of the phone. If these are not collected and the forensic investigator does not have a duplicate available in the lab, it can delay the investigation.

Sometimes evidence is found on the computer system with which the smart phone was synced and not on the phone itself. Therefore if a computer system is found on the crime scene, seize it as well. Useful locations for the evidence files on the computer systems for the iPhone are:

1. In case of Microsoft Windows operating systems before Vista and Windows 7 such as Windows XP, the synced data can be found at Documents and Settings\<user account>\Application Data\Apple Computer\Mobile Sync\Backup
2. In case of Microsoft Windows Vista and Windows 7, the location is C:\Users\<username>\AppData\Roaming\Apple Computer\MobileSync\Backup. The sync data is stored in directories with random generated names.



Name	Date modified	Type	Size
00f75308934c4d9c4b578fa121cdaf5202a794aa	21-Aug-2011 6:06 PM	File	1 KB
0a509521c2daf30081adc6cf28690fa4e1d8c34e	21-Aug-2011 6:06 PM	File	3 KB
0aa7ea661b9dd50544667c80bb30a1260f419a1c	07-Dec-2011 12:44 ...	File	1 KB
0af00ce6e509204f48fafecdb11529c869b14dcf	05-Oct-2011 1:42 AM	File	4 KB
0b68edc697a550c9b977b77cd012fa9a0557dfcb	29-Dec-2011 9:06 PM	File	8 KB
0b195b1d04f03ba5b423b4dd74cb28e2cd892b2b	21-Aug-2011 6:06 PM	File	812 KB
0b290d870bdf6411836550dcab1367af4d01cc43	21-Aug-2011 6:06 PM	File	1 KB
0c511c5e0c25b370ed27dc09299fc945ae214503	05-Oct-2011 1:42 AM	File	12 KB
0cb374a15c6694e17f651ceb7f411e433f21778c	05-Oct-2011 1:42 AM	File	51 KB
0cd068d10dca4ecc1de59de4d6eb70891a5045f9	04-Sep-2011 8:03 PM	File	6 KB
0d48c505d130eea97d0a51a55c4c78c8fc0a338f	21-Aug-2011 6:06 PM	File	5 KB
0dc926a1810f7aee4e8f38793ed788701f93bf9d	29-Dec-2011 9:06 PM	File	1 KB
0df2b5733a2f08b9db62ccf573894412845b85a6	16-Oct-2011 4:51 PM	File	1 KB
0e00aa3cbedc85ea66edf43cb688f983538acb86	29-Sep-2011 8:15 PM	File	5 KB
0ed6e9d32f8444fdbaa319d281acf420970fbfc6	21-Aug-2011 6:06 PM	File	0 KB
0fb54654b97099d34461570fab859a2b0570ed1f	29-Dec-2011 9:06 PM	File	1 KB
0fc8189497f46a2e2511c846acbbb318d3a43ec3	29-Dec-2011 9:06 PM	File	12 KB
1a9533d30015f5746046c7a2bfafe1584c2a3d40	29-Sep-2011 8:15 PM	File	13 KB
1ae5978e55d418d2478a3e9975135923bc639f58	21-Aug-2011 6:06 PM	File	5 KB
1b3bf6031e609f6363f82950c02b3cc80a431f35	05-Oct-2011 1:42 AM	File	5 KB
1b7b997bd7034978fc4dcc62fa51f902ea377a18	21-Aug-2011 6:06 PM	File	5 KB
1bb8d3ff9f6ad1b67a521a216274b2a589402a01	21-Aug-2011 6:06 PM	File	13 KB
1c03f3e801e0009955739f27e2d9af310dd5832b	21-Dec-2011 1:34 AM	File	1 KB
1c718cba44380a5b5495665868e4219e7abe15ef	21-Aug-2011 6:06 PM	File	13 KB
1d4cf97dc6200de1b65c562aec103ae9097ed445	05-Oct-2011 1:42 AM	File	10 KB

Figure 2-1 iPhone Sync Data on Windows 7

3. In case of Mac, the location is ~/Library/Application Support/MobileSync/Backup where the ~ sign is used for the user's home directory which is normally /home/<username>.

In some cases, the hidden files and folders and Operating Systems files and folders need to be turned visible to access the locations.

2.3.2 Acquiring Investigation Data

The next step in the forensic investigation of smart phone is to acquire the evidence from the device. This is the most difficult part since it involves direct interaction with the underlying hardware and software of the phone and if proper care is not taken, the evidence can be permanently damaged. Before performing the actual acquisition from the phone, some steps are needed to be performed on certain phones to make them ready for data acquisition. Two generic steps are needed before acquiring data from the device: First look for any pass code or other authentication mechanism used to

login to the phone. Secondly, perform any necessary steps used to make the device ready for data acquisition such as opening a port or assigning an IP address.

In case of obstructed devices (those requiring pass code), the following steps might be helpful:

1. Do not experiment with the device
2. Contact the manufacturer for possible backdoors and/or vulnerabilities
3. Review manufacturer's documentation
4. Review the seized material. Passwords might be written down somewhere nearby on a paper.
5. Exploit possible default settings e.g. default passwords.
6. If SIM is locked and require PUK code, ask the network provider
7. Ask the suspect for the password. This method is however not very helpful since in some cases, entering a particular password may actually open the device but format the whole hard drive. Therefore, used this as a last option
8. If the JTAG debugging interface is available on the circuit board of the phone, the authentication mechanism can be bypassed through this debugging port and data can be accessed directly from the phone at the hardware level.

2.3.3 Logical Extraction

Logical extraction of data is done for two reasons: when the device is closed source and no method for physical acquisition is available; and when there is a need of rapid investigation and analysis of data and the forensic investigator has limited time. This is often done through forensic tools and in some cases through the management and sync tools normally found with each smart phone. Logical extraction can acquire:

1. Live SIM (Subscriber Identification Module) data
2. Live phone data
3. Deleted SIM data using a card reader

The advantage of logical acquisition is that the data recovered from the device is in proper format and understanding of the recovered data is very easy. However the disadvantage is that deleted data cannot be recovered using the logical acquisition since this method only recovers data accessible through the file system Application Programming Interface (APIs).

The process to recover data using the logical acquisition is as follows:

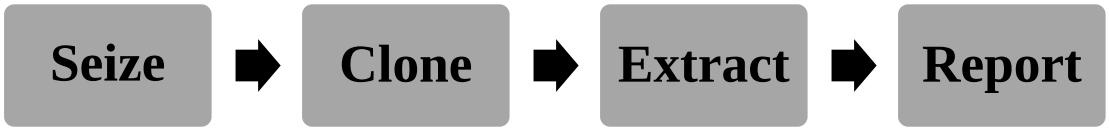


Figure 2-2 Logical Acquisition

2.3.4 Physical Extraction

Physical extraction is the preferred method of acquiring the data from the device. However, this is only possible if the phone file system is open source or if sufficient specification details are available. Physical data dumping involves data retrieval at the media level where the bits and bytes are recovered along with the meta data. Some of the advantages of physical data dumping are:

1. Complete data image
2. Extract even deleted data
3. Extract all system information normally not accessible through the file system such as network provider information, previous IMSI (International Mobile Subscription Identification), etc
4. Retrieve data even if SIM is not available
5. Memory card analysis
6. View data in raw (HEX) format

Because of the ability to recover deleted data as well through this method, the forensic investigators always prefer physical data acquisition to the logical one. However, the disadvantage of physical data acquisition is that it requires time and expert knowledge to interpret the recovered data since it is in raw format. Some of the other disadvantages are:

1. Harder to interpret
2. More data to interpret
3. Harder to retrieve data
4. Fewer devices are supported

The process of physical acquisition is:

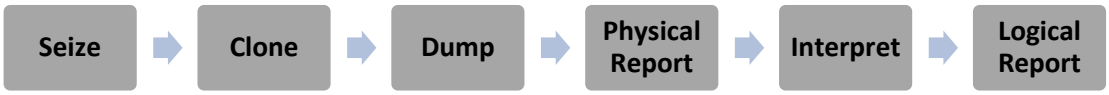


Figure 2-1 Physical Acquisition

If both logical and physical acquisition is possible, it is better to first acquire a physical acquisition so that all the data is cloned and then a logical acquisition is performed to interpret the data more easily and fast.

Note: There is no software or method available to retrieve data from a non jailbreak iPhone. The only method of non jailbreak iPhone is to manually navigate the file system and take pictures, videos, and notes along the way as evidence. However this method is not 100% safe and may change the file system by a small proportion e.g. the time stamps are updated when the file or folder is accessed through the file system. Therefore, the best way to gather evidence from a non jailbreak iPhone is to first jailbreak it using a safe method (discussed later in chapter 5) and then used a standard acquisition procedure.

2.3.5 Analyzing Evidence

This is the part where the forensic investigator analyzes the data acquired from the smart phone. This is a manual process where the investigator links the data and evidence found on the smart phone with the crime and either proves or disproves the connection of the person and the device with the crime. Following guidelines should be followed here:

1. Know your expectations. Too small data might not be sufficient for forensic investigations. Too much data might be difficult to analyze
2. Know where the data is located
3. Deals with proprietary data
4. Focus on the deleted data
5. Examination should describe source as well as significance of the data
6. Identify individuals who created, accessed, or modified a file

7. Determine when events occurred by analyzing the time stamps. Network service providers may also store certain information e.g. call time. Compare the network service provider's time with the time on the phone to uncover any discrepancies.
8. Analyze headers of the file to determine the file types. User might have deliberately changed the file extension to confuse the forensics investigator and the data acquisition tool. Therefore, don't rely on file extensions. Instead, analyze file headers.

2.3.6 Reporting the Findings

The final step is to report the findings. Here, the forensic investigator needs to take proper care since the upper management in case of civil investigation and the judge and jury in case of criminal investigation only study the final report and don't give too much concern how the data is acquired and analyzed. Therefore, a formal report should be devised where the findings should be properly documented and explained. It is very important that the report should contain a clear statement stating whether the evidence of the crime was found on the device or not.

A formal report should contain the following items:

1. Identity of the reporting agency
2. Case identifier or submission number
3. Case investigator
4. Identity of the submitter
5. Date of the receipt – Date of the report
6. Descriptive list of items submitted for exam e.g. serial number, make, model etc
7. Identification and signature of the examiner
8. Equipment and setup used for examination e.g. hardware and software used for data acquisition and analysis
9. Brief description of steps taken during the examination
10. Report should be supported by proper pictures, audio/video files wherever appropriate and possible
11. Details of findings
    i. Specific files
    ii. Other files such as deleted ones
    iii. String searches
    iv. Internet related evidence e.g. websites, chat logs, cache files etc

v. Graphic images
vi. Indication of ownership e.g. program registration
vii. Data analysis
viii. Report conclusion

2.4 Conclusion

In this chapter, the custom developed forensic methodology is explained. Each phase of the methodology and the tasks for each phase are also adequately explained so that the forensic investigation knows exactly how the investigation will proceed.

Chapter 3

iPhone File System Architecture

3.1 Introduction

This chapter provides an overview of the iPhone file system and the underlying security restrictions and constraints implemented by Apple Inc that can hinder the forensic investigation. iPhone has a history of always been the target of hackers to perform functions that were not intended and hence, Apple Inc, with each subsequent release of a new iOS version (formerly known as iPhone OS) implement more rigorous security restrictions. However, the forensic investigator can use tools and techniques that can bypass majority of the security constraints and perform a smooth forensic investigation of the iPhone. However, the forensic investigator also needs to understand the security restrictions, how to bypass each one of them and what effect it has on the underlying architecture and file system. This is important to make sure no unintended action is executed on the iPhone and no important evidence is lost in the process.

3.2 Architecture

iPhone file system is based on the HFS+ file system implemented in the Mac OS. Before firmware 4.0, the iPhone operating system was known as iPhone OS but after the introduction of iPhone 4G with firmware 4.0, the firmware OS is now called iOS. Functionally, iPhone consist of two distinct components: the iOS and the baseband. Function of iOS is that of a computer operating system. It performs all the low level functions such as process and memory management, file storage, etc. Each of the core functionality such as process and memory management is performed by one or more daemon services implemented by the iOS. On top of the core operating system functionality, a framework is implemented which communicates with the underlying iOS on behalf of the user. This framework is called Cocoa framework. An interesting thing to be noted here is that iOS has two distinct interfaces through which users are able to interact with the underlying hardware: one is the official Apple Cocoa framework which acts as a proxy between user applications written in objective C language and the kernel of the iOS. The Cocoa framework translates and pass on the

communication between applications and iOS. The second method is jailbreaking the device and communicating directly with the hardware completely bypassing the Cocoa framework. This is possible because the iOS is based on FreeBSD Unix distribution and much of the core of iOS is composed of Unix daemons and kernel. When the user directly communicates with the underlying hardware, much of the security restrictions implemented by Apple is completely circumvented. A short summary of the security restrictions bypassed through jailbreaking is provide in the below sections.

3.2.1 Unsigned Code Execution

By default, iOS 4.0 and above firmware allow only the code signed by Apple Inc to be executed. When a particular application is developed for iPhone and other iOS based devices such as iPad and iTV, the application has to be submitted to the Apple AppStore for evaluation. The application is evaluated by the Apple iPhone developers to ensure that proper guidelines are followed and no backdoor is implemented within the application for stealing sensitive user data. Once the evaluation is completed, Apple Inc signs this application code with its private key. The Cocoa framework security component then uses the Apple public key to make sure the code is signed before execution of the application on the iPhone. This protection measure is used to ensure that only safe code is able to be executed on the device and hence no unintended function could be performed by the iPhone. However, this protection mechanism is not hack-proof and firmware 4.0 and above is already jailbreaked. The way unsigned code is executed on the device is achieved through a two step process: first vulnerability is exploited which give the attacker the ability to execute her shell code. However, since the unsigned code cannot be executed directly, the shell code is attached through the boot strap process. When the iOS is restarted, the unsigned code is executed before the signed code protection comes into play. Hence the user is able to execute arbitrary shell code on the device and gain full control of the device.

3.2.2 Complete Control of the Device

By default, Apple has restricted the iPhone user to access only a handful of functionality and resources through the iOS interface. There are two user accounts in iOS: one is the all powerful root account and the other is a normal Unix mobile account. By default, iOS runs through the mobile user account which is restricted to

the directory /private/var/mobile and the sub directories. Linux utilities are normally saved in /bin/, /sbin/, /usr/bin/, and /usr/sbin/ directories and hence the mobile user cannot access these directories. This restriction is called “jail” or “jailing” in Linux or Unix world. Here, the root directory of the file system represented through “/” in Linux/Unix is changed for a particular user. In case of mobile user in iOS, the “/” of the file system is changed to /private/var/mobile and hence everything outside of the /private/var/mobile directory cannot be accessed by the mobile user account. When the iOS device is jailbreaked (here we have used iOS device instead of iPhone because the same technique can be used to jailbreak all iOS devices such as iPad, iTV, and iPhone), the root account is used to access the whole file system. Therefore the user is no longer restricted to the /private/var/mobile jail directory. The process is known as jailbreaking the device.

3.2.3 Work with Unsupported Networks

As discussed previously, iPhone is composed of two components: iOS and baseband. The baseband is the component which is responsible for communication with the mobile network. iPhone can communicate with GSM and CDMA, By default majority of the iPhone devices are carrier locked. This means that a particular iPhone can only be operated with a particular mobile network such as Verizon and AT&T. However, this restriction can be bypassed with both hardware and software depending on the baseband version currently in use. The way Apple Inc has restricted the device is through a unique encryption key known as the SHSH blobs. This key is unique to each device and is used by the iTunes software to authenticate the device to the particular network during the activation process. Since SHSH blobs contain the device ID, Apple Inc activation server checks the iPhone and the inserted SIM (Subscriber Identification Module) to verify whether the particular SIM can be used by the device or not. However, this restriction can be bypassed. Prior to the baseband version 05.16.02, the baseband has a vulnerability through which a software exploit code in “Ultrasn0w” could be used to unlock the device. It is also a two stage process. In the first stage, the iPhone iOS is jailbreaked and then a community unlocked server, hosted by a person named Jay Freeman also known as Saurik, is used to unlock the device with any SIM. However, with the latest baseband, the only way to unlock the device is through the hardware chip known as the magic SIM. The way magic SIM works is that a normal SIM is mounted over the hardware SIM and the magic SIM

acts as a proxy between the Apple Inc activation server and the original user SIM card. When the Apple Inc servers check the SHSH blobs of the device, it checks the IMSI of the SIM. However, the magic SIM changes the original IMSI and make it compatible with what the Apple Inc activation server is expecting. This makes the activation server to believe that a supported SIM card is inserted into the device. Hence, an unsupported SIM card starts communicating through the baseband.

3.3 Conclusion

In this chapter the iPhone architecture is described to provide an overview of the iPhone file system, how files are organized within the file system and the internal hardware and software components that are responsible for each function. Further, the chapter also discusses the security restriction implemented in iPhone, how those restrictions are bypassed and the implications when the restrictions are circumvented.

Chapter 4

Implementation of the Toolkit

4.1 Introduction

Since iPhone is a close source hardware and software platform, therefore, forensic investigation is a challenge. Most of the forensic investigators do not know the platform completely; therefore, they rely on incomplete and closed source forensic tools that have little credibility. As a part of this research initiative, an open source forensic toolkit has been designed for forensic investigation of iPhone that can extract logical file system contents as well as can recover deleted data from the iPhone. This chapter provides an overview of the architecture and operations of the toolkit, which files are covered by the toolkit, as well as some best practices and guidelines on how to effectively perform the recovery procedures through the forensic investigation toolkit.

4.2 Toolkit Structure

The forensic investigation toolkit is developed in C#.NET to provide cross platform compatibility. The same code can be run on Unix-like operating systems using the open source Mono project [15]. Mono is the open source fork of the .NET framework which is a Microsoft proprietary technology. The official Mono project page describes Mono as:

"Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin [a Boston, Massachusetts based technology firm], Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime."

Therefore, it is easy to take the source of the toolkit and compile it on a Linux platform using the Mono project's Common Language Runtime (CLR). CLR is the .NET framework compiler which uses the Just-in-time compilation technique to convert the .NET high language source code into machine code.

4.3 Gaining Root Privilege

iPhone comes by default with a restricted environment where every code that is run by the firmware needs to be digitally signed by Apple Inc. This restriction is bypassed by securely jailbreaking the device so that root privilege can be obtained. Root privilege on the iPhone can be gained using both the manual and automatic method. Manual method is more preferred since the examiner has more control over the jailbreaking process. However, the examiner can also follow the automated method developed by Nicholas Allegra [16] because changes in the firmware during the jailbreak process occur on the first partition where only the OS files are saved. User address book, SMSs, emails, and other important information are stored on the second partition and therefore, they remain intact no matter which method is followed for jailbreaking the device.

Once the device is jailbreaked and root access is obtained, an OpenSSH server is installed on the device. A secure shell client implemented within the forensic investigation toolkit is then used to connect to the OpenSSH server and pull data from the iPhone device to the forensic investigation machine. This process is both transparent and secure since all the data is transferred through the encrypted SSH tunnel where both confidentiality and integrity of the data is provided by the protocol. The forensic investigation toolkit menu is organized as follow:

1. Logical acquisition: Important files stored on the iPhone file system are transferred from the phone to the forensic investigation machine. No deleted data is recovered and only those files that are stored on iPhone file system are recovered. This is helpful in fast searching and retrieving evidence.
2. Physical acquisition: Bit by bit copy of the whole iPhone file system is obtained. This provides the forensic investigator a complete copy of all the data stored on the iPhone.
3. Data carving: Once bit by bit copy of the iPhone file system is complete, Scalpel, which is an open source forensic data carving tool, is used to recover data from the device. Scalpel is an industry standard tool for securely recovering deleted data and is integrated within the forensic investigation toolkit.

Before performing the Logical acquisition, the forensic examiner needs to check the following:

1. An IP address is assigned to the iPhone
2. OpenSSH server is installed on the device
3. The forensic examiner can authenticate to the OpenSSH server

If the examiner does not know the IP address of the iPhone, procedure described in chapter 5 section 5.2 on how to view the assigned IP address in iPhone can be used as reference.

If the examiner suspects that no OpenSSH server is installed on the device, procedure described in chapter 5 section 5.3.5 explains how to install OpenSSH server in iPhone.

If the examiner does not know the OpenSSH credentials, the first step is to use the default username “root” and default password string “alpine” as majority of the users do not change the default credentials. However, if the authentication fails, please refer to chapter 5 section 5.3.6 on how to reset the OpenSSH server password securely.

4.4 Logical Acquisition

Logical acquisition is the step where the examiner recovers all the necessary data files that are stored on the iOS file system. No deleted data can be recovered through this step as the forensic investigator will be able to recover only those files that are part of the normal operating system file management structure. There are two advantages of performing this step. One is that the forensic investigator will be able to work with a copy of the original data and the original data will be preserved. This is to ensure that the guideline described in the forensic investigation methodology for smart phones in chapter 2 is properly followed. The second advantage of the step is to perform quick forensic investigation without going into the tedious steps of data carving and recovery. If the forensic investigator needs to perform an investigation where time is of outmost importance such as in kidnapping case, this step will ensure that data is provided to the law enforcement agencies as soon as possible and hence logical acquisition of the files will help the forensic investigator in such cases. The forensic investigator will be able to recover the following files through this step:

Table 4-1 Logical file recovery

/private/var/mobile/Library/AddressBook/AddressBook.sqlitedb
/private/var/mobile/Library/Calendar/Calendar.sqlite
/private/var/mobile/Library/CallHistroy/Call_history.db
/private/var/mobile/Library/Cookies/Cookies.plist
/private/var/mobile/Library/Draft/PENDING.draft/DraftMessage.plist
/private/var/mobile/Library/Logs/AppleSupport/General.log
/private/var/mobile/Library/Safari/SafariHistory.plist
/private/var/mobile/Library/SafeBrowsing/SafeBrowsing.db
/private/var/mobile/Library/SMS/sms.db
/private/var/mobile/Library/voicemail/Voicemail.db

These files are very important in the forensic investigation of the suspect as these files contain all the data necessary for the investigation such as the SMS messages, call history, phone book, browsing history etc. These files are downloaded from the device to the working directory from which the forensic investigation toolkit is invoked. All these files are stored inside a new directory by name of “Evidence” in the current working directory of the forensic investigator. Majority of the files are in SQLite database format, therefore, the forensic investigator can use the command line or graphical based SQLite browser. Command line is more preferred as the forensic investigator can easily execute one SQL query against multiple files to ensure proper data is returned. For example, the investigator can run a nested query against the sms.db and AddressBook.sqlite files to return the SMS messages and sender/receiver number from the sms.db file and then match those sender receiver numbers in the AddressBook.sqlite file to return the stored contact names for those numbers.

4.5 Physical Acquisition

Physical acquisition is the bit by bit copy of iPhone hard drive. This is a compulsory step in any forensic acquisition as through this step all the data on the disk is recovered i.e. both the data available in the file system and the data deleted by the user from the file system but still not overwritten by other data. This step can potentially recover more important forensic evidence because most of the time the

suspect might delete the data from the iPhone which she thinks might get her in trouble in the court of law and used as evidence. However, the data is still stored on the iPhone hard drive until it is overwritten. However, this step is bound to be more time consuming as the complete hard drive copy is taken.

4.6 Data Carving

Because of the file formats used within the iPhone, data carving and recovery of the hard drive produce false positive rate as well. Majority of the important forensic investigation files are SQLite database files which have a defined header format but no trailer or footer of the file is defined. Hence, searching byte pattern on the hard drive is bound to produce false positives. However, false positive is accepted in forensics because too much data is better even if it takes more time to analyze. If incomplete data is recovered, the forensic investigator might miss important evidential data. Also, the false positive rate can be reduced through intelligently selecting the maximum size of the files we are recovering. The file sizes specified in the table below can be used as a reference for specifying how big the files size to specify when recovering files.

Table 4.2 – Recommended File Sizes

File Name	Type	Size
Sms.db	Default SMS file	1.2 MB ~ 2 MB
AddressBook.sqlitedb	Default address book	220 KB
Notes.db	Notes file	50 KB
Calendar.sqlitedb	Calendar entries	204 KB
Call_history.db	User call history	30 KB
History.plist	Safari browser history	1 MB
Voicemail.db	Voice mail information	100 KB

File sizes specified above are the best estimated file sizes observed during the research where the iPhone was heavily used for call and SMS for a continuous period of 9 months. The forensic investigator needs to determine for how long the iPhone was used by the suspect or the victim before specifying the file size. More

information on determining how long the iPhone is used is discussed in the forensic investigation scenarios.

4.7 Conclusion

This chapter explains the architecture and design of the forensic investigation toolkit that is within the scope of this research activity. Each of the tasks performed by the toolkit is explained with details of the internal working of the toolkit. Additionally, some best practices are also discussed through which the capabilities of the toolkit can be used more effectively. At the end of this chapter, a table summary of the base line file sizes are also presented through which the deleted data can be recovered more effectively with less false positive rate.

Chapter 5

Analysis of Forensic Evidence

5.1 Introduction

This chapter is organized in two parts: the part outlines the helping procedures and steps that are necessary for successful extraction of data. The second part deals with the analysis of data once the evidence is successfully extracted from the iPhone device. The most important task the forensic investigator needs to perform is to decide which files to analyze and recover during the forensic investigation process. Knowing where important files and data that can be helpful in the investigation are saved on the file system is a primary task. The second part of the chapter describes all files and their location on the iOS file system structure, as well as provides details about why the files are important. Detailed description of the purpose of each file is given so that the forensic investigator knows where to find a particular piece of data on the iOS file system.

5.2 Forensic Evidence

The iPhone file system is based on HFS+ file system which is based on FreeBSD Unix. Therefore, most of the file structure on the iPhone hard drive closely resembles Unix file system structure.

By investigating the file system, we have identified list of files that are needed in the iPhone forensic investigation. Table below identifies important files needed for forensic investigation.

Table 6-1 Evidence files on the iPhone

File Name	Location on iPhone file system	Description of the file
General.log	/Library/logs/AppleSupport/	Contain information about iPhone firmware
*.deb	/private/var/mobile/Library/Backup	Contain Apps downloaded through AppStore
Status	/private/var/lib/dpkg/	Contain installation status of the Apps downloaded on the iPhone
localtime@ which is a soft link to /usr/share/zoneinfo/<configured timezone name>	/private/var/db/timezone/	Contain local time zone configured on the iPhone
All directories	/private/var/stash/Applications/	Contain all the applications on the default SpringBoard desktop and installed from Cydia
AddressBook.sqlitedb	/private/var/mobile/Library/AddressBook/	Database file of default address book containing all the phone numbers
Calendar.sqlite	/private/var/mobile/Library/Calendar/	Database file contain important dates and events configured on iPhone
Call_history.db	/private/var/mobile/Library/CallHistroy/	Database file containing information about calls dialed and received as well as missed.
Cookies.plist	/private/var/mobile/Library/Cookies	Cookies saved for websites in Safari browser
DraftMessage.plist	/private/var/mobile/Library/Draft/PENDING.draft/	Messages written but not yet send through the Messages application
General.log	/private/var/mobile/Library/Logs/AppleSupport/	Contain diagnostic logs and iOS version information
SafariHistory.plist	/private/var/mobile/Library/Safari/	Contain website history information of the Safari web browser
SafeBrowsing.db	/private/var/mobile/Library/SafeBrowsing/	Database file contain websites visited using the Safe Browsing feature of Safari.
sms.db	/private/var/mobile/Library/SMS/	Default SMS database file contain all the messages sent and received
Voicemail.db	/private/var/mobile/Library/voicemail/	Database file contain voicemail information of the sender

A brief description of all the files specified in the table is given below.

general.log: This is the first file that should be analyzed in order to find out what version of the iOS is used on the device. This is important because this information will specify what protection measures are in place and which vulnerabilities can be

exploit for the forensic investigation. Each version of iOS firmware has new security measure and restrictions that needs to be bypassed for successful forensic investigation. For example, iOS firmware 4.0+ has Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR) which needs to be bypassed in order to gain root privilege on the device. Therefore, the firmware version specified in the general.log file will give the forensic investigator the idea of which tools he/she needs for the forensic investigation of the device and for bypassing the security and compatibility restrictions.

***deb files:** These are traditional package files used in debian based Linux OS. Since iOS is based on FreeBSD which has the underlying debian file system, therefore, all package files are in deb format. Examining these files can give the examiner the idea that which packages were downloaded by the user and used on the iPhone. Even if the installed package is subsequently removed from the device, the downloaded package still remains there until the maximum archive size specified in apt.conf file is reached. There is a high probability that the package will remain in the folder and hence can be deducted that particular hacking or cracking applications, such as Nmap, Metasploit, or Netcat which are popular hacking applications and are successfully ported to the iOS environment, were used by the user. Finding deb files on the iPhone is also a proof that the phone is jailbreaked because Apple does not support package installations on the iPhone through the debian package manager, therefore, it means the user has either manually installed the applications on the iPhone through the apt-get command line package manager or has used Cydia package repository. Particular care should be taken while dealing with .deb files because these files are not thoroughly checked for backdoors and other malicious code and executing an unfamiliar deb file on the iPhone can destroy potentially sensitive forensic evidential data.

Status text file: Together with the .deb archive files, this file can be very useful in establishing that the user not only downloaded a particular application that was used in the hacking/cracking incident but also installed the application. This file contains the status of each package ever installed on the iPhone such as applications downloaded and installed from Cydia repositories. Packages can be downloaded in a "download only" mode as well where a particular package and its dependencies are downloaded but not installed. Therefore, just finding a deb package in

/private/var/stash/Applications/ or apt-get temporary archive directory does not establish the user has actually installed the package. Therefore, the “status” text file must be used to manually verify the installation status of each package.

Local time settings: Local time settings can be useful in deducting the actual location of the accused or the victim possessing the iPhone. For example, if a dead body is found in a location and iPhone is recovered from the crime site, it can be established from the local time settings where the person belonged to, if he/she is the owner of the phone. This file is also helpful in correcting the time difference between different time zones.

/private/var/stash/Applications: This directory has sub directories representing each of the application available on the default SpringBoard desktop of the iPhone and applications installed through the Cydia application repository. This can be useful when a user deliberately hides applications from the normal display. For example, applications are available from both Cydia and AppStore application repositories which hides any application from the SpringBoard until a particular key combination such as a password is entered. This protection can be circumvented by examining the contents of this directory.

The most important files required for investigating forensic crimes are located in /private/var/mobile/Library/ folder. This folder contains sub folders and files containing SMSs, call history, address book, emails, drafts, etc. Some of the most important files examined during the research are:

Calendar.sqlitedb: This file has the user calendar entries such as any important events that were set on particular days. Calendar entries are very important in the forensic investigation because it provides the schedule of the accused or the victim (from whoever the iPhone was recovered from).

call history.db: As the name suggests, this files contains the call history. Calls dialed, received, and missed can be located here.

sms.db: This file contains SMSs sent and received through the mobile. A lot of third party SMS applications from AppStore and Cydia repository also use this file for the storage of SMS messages. However, the forensic investigator should note that sometimes the user might use a third party SMS application which has a difference

file and storage format. In such situations, the forensic investigator needs to find the particular file used by the application.

PENDING.draft: This file contains SMSs that are not yet send. By default, iPhone do not have a draft saving feature. However, messages written in the Messages applications not yet send are saved in this file. As soon as a message is send through the Messages application, the content are transferred from this file to the sms.db file.

History.plist: This file contains the browser history such as the website visited through the iPhone. iPhone uses Safari web browser as the browser application which stores all history information such as websites visited from the browser in this plist file. Because of the lack of support for flash in the iPhone, no other mainstream web browsers such as Google Chrome and Mozilla Firefox has been ported to the iPhone platform, therefore, Safari is usually the only browser the forensic investigator needs to investigate.

voicemail.db: Voice mails send and received are located in this file.

iPhone file system architecture is a combination of both open source and proprietary technology. The basic kernel of the OS is based on FreeBSD Linux kernel but on top of the kernel, Apple Inc has implemented many proprietary technologies for performance and security. From a forensic perspective, it is important to understand both the open source and the proprietary technologies. Luckily, the proprietary technologies are either simple or are adequately explained by the Apple's technical support document. For example, the ".plist" files are proprietary files developed by Apple Inc for iPhone and Mac OS but the structure of the file is well defined where configuration information are stored in a hash table data structure in the form of variable-value pairs. The most important file format on the iPhone file system is the SQLite database file format. It is an open source database format specifically developed for low processing and low memory devices. The complete database engine is implemented in a single file and hence no pre-requisites are needed. Most of the user files such as address book, SMS, call history, emails, and voice mails are all stored in SQLite database format. Therefore, understanding and recovering these files are very important. The only drawback of SQLite file format is that it has a header but no footer. Therefore, from a forensic perspective, recovering these files is very difficult. When file formats does not have well defined footers and we need to

recover the files through data carving, it produces a lot of false positive. The only feasible method for recovering of such files is to combine the header of the file with an optimum file size. This technique is also used in the forensic framework toolkit that is a part of this research activity.

SQLite specifications have a 16 byte special hexadecimal pattern *\x53\x51\x4c\x69\x74\x65\x20\x66\x6f\x72\x6d\x61\x74\x20\x33\x00*. This header information can be used in combination with the maximum file size to intelligently restore deleted files in this format. During our experiment, we examined an iPhone used heavily for both calls and SMSs for a continuous period of 9 months. Table 4.1 can be consulted for the observed file sizes. Also, Table 4.1 can be used as a reference base line and hence the forensic investigator can determine intelligent file size entries for the case in hand. The amount of time for which the iPhone was used can be obtained through the general.log file. The file contains the date of the installation of firmware and hence the number of days can be calculated from that date to the present. Once it is established for how long the iPhone was used, intelligent file sizes can be specified using the above table as a reference.

5.3 Conclusion

This chapter provides all files and their corresponding storage locations on the iPhone which are needed for data extraction and analysis. Each file is explained separately so that it is clear where a particular piece of data is stored on the device. This chapter provides help in two situations: before the data extraction phase and after the data extraction phase. Before the data is acquired from the device, the forensic investigator can consult this chapter to determine the likely hood of each of the evidence in a particular file or folder. When location of each file is known, non relevant files can be omitted which can greatly reduce the time of investigation. The chapter is also helpful when data extraction phase is complete since the forensic investigator can use it as a baseline to compare and to make sure all important files are recovered successfully.

Chapter 6

Evaluation of the Toolkit in Real World Case Studies

6.1 Introduction

This chapter provides case studies to show the effectiveness of the forensic investigation toolkit in the investigation of real world crime cases where iPhone is used. Knowing that iPhone is a multipurpose device that can not only perform functions of a mobile phone but can also perform almost all the tasks that are commonly done through a computer makes the scope of investigation very broad. The case studies describe here can give the forensic investigator of the iPhone valuable ideas about how to conduct a real world investigation.

6.2 Traditional Crimes Investigation

Because of the capabilities and the popularity of iPhone among general public, the phone can sometimes become the most important piece of evidence in traditional crimes such as murder and theft. Recovering data from the phone can be difficult if proper tools and techniques are not followed. During the research, we recovered data from a modern iPhone 3G with 8 GB of internal storage. The phone was using firmware 3.2.

The first task is to safely jailbreak the device so that root access can be obtained on the device and security restrictions can be bypassed. This should be done using offline jailbreak tools such as Snowbreeze. iPhone firmware version up to 4.3 can be easily jailbreak through this software. Forensically, it is a secure step because all the changes are done on the first partition which is not important for investigating criminal cases. Crime cases normally involve investigating and recovering the iPhone's contact list, SMS, emails, audio/video and call history etc. Jailbreaking causes all the changes to the first partition therefore; no data is altered or lost on the second partition where the entire user's phone data is stored. Once the phone is jailbreaked, use the Cydia repository or manual deb files to install OpenSSH server on the phone and establish a SSH connection. Then download all the important files in /private/var/mobile/Library folder and offline analyze it to recover and analyze the data. This step can be performed with the custom developed toolkit that is developed

within the scope of this research. The most important files are call history, contact list, SMS, and email. All these files are stored in SQLite DB files and therefore, can be analyzed with any SQLite browser such as the open source SQLite Browser or Hex dump software. A step by step procedure the forensic investigator needs to perform for data acquisition and analysis is as follows:

1. Follow the procedures and guidelines specified in chapter 2 for secure acquisition of the iPhone from the accused or the victim.
2. Follow the forensic investigation evidence paper trail procedure.
3. Turn on the iPhone if it is off.
4. If the iPhone is password protected, use the procedure specified by Jurgen Schmidt and Dino Dai Zovi in [13] and [14].
5. If the iPhone is jailbreaked already, ignore this step. If the iPhone is not jailbreaked, follow the procedure specified in Chaper 5, section 5.3 on how to jailbreak iPhone securely.
6. If OpenSSH is already installed on the iPhone, ignore this step. Otherwise follow the steps specified in chapter 5, section 5.3.5 on how to install OpenSSH server on the iPhone.
7. Connect the iPhone to a secure wireless Access Point (AP) and connect the forensic. The forensic investigator should follow the guidelines specified in chapter 5, section 5.2 on how the wireless network should be setup for secure extraction of data.
8. Use the default OpenSSH server credentials with the username “root” and password “alpine” for connection to the OpenSSH server on the iPhone.
9. If the authentication request fails, follow the steps specified in chapter 5, section 5.3.6 on how to reset the OpenSSH password.
10. Once the connection is established, follow the logical acquisition method to extract data from the iPhone.
11. Check the data extracted through the logical acquisition step. The data will be stored in the “Evidence” directory.
12. Majority of the recovered files such as phone book, call history, and SMS messages are in SQLite database files. Use any SQLite browser to open and analyze data stored in these files. The command line SQLite browser is recommended because the forensic investigator can run nested SQL queries on multiple files to gather more intelligent data. For example, selecting message and

sender's phone number from sms.db file and then searching the phone number in the PhoneBook.sqlitedb file to return the name of the contact person can be done using one nested SQL query.

13. Before performing the physical acquisition of the iPhone hard drive, make sure enough space is available on the forensic investigator's computer or laptop. The image size of iPhone hard disk will be equal to the hard disk size in the iPhone. For example, if the forensic investigator needs to dump the hard drive of the iPhone having a 32 GB hard drive, make sure there is at least 32 GB of space available in a single partition on the computer or laptop for the storage of the image file.
14. Perform physical acquisition. The physical acquisition step will dump the complete bit by bit copy of the iPhone hard drive in a dd (data dump) format that can be read by all forensic investigation tools.
15. Follow the data carving procedure to recover deleted data from the hard drive image. If files from a different format needs to be recovered, just insert header (and optionally footer) of the file in the Scalpel file. The new format will also be recovered.
16. If the data carving step produces a high false positive rate, make sure the maximum file size is intelligently selected. Table 4.1 can be used as a reference for selecting intelligent maximum file size values for the files.
17. Once the data acquisition step is complete, analyze the recovered data and produce a report detailing the findings of the investigation. Follow chapter 2, section 2.3.6 for how to write a professional forensic investigation report.

6.3 Hacking and Cracking Investigation

Investigating hacking and cracking incident committed through iPhone can be investigated as well. This involves analyzing the particular application files used for hacking as well as some configuration and log parameters. Note that if we jailbreak the iPhone for hacking incident investigations, it will overwrite all necessary log and application files because those files are also stored on the first logical partition that is changed by the jailbreak process. However, fact of the matter is, out of the box, iPhone cannot be used as a hacking device and if a phone has indeed been used for hacking, it means it is already jailbreak. Nmap, Metasploit, Netcat, aircrack-ng etc., commonly used in iPhone for hacking, can only be installed if the device is already

jailbreak. Therefore, for investigating such crimes, there is no need to jailbreak the device and hence the forensic integrity of the device is never disturbed. The most important files for hacking investigation are the installed application directory at /private/var/stash/Applications and the .bash_history file in the root directory. The former will show all the installed applications on the iPhone while the later shows all the commands run through the terminal. Hence, using these two files, it can be easily determined if the iPhone is used in any digital attack or not.

Figure 7-1 iPhone Used as a Hacking Device

As can be seen in the diagram above, iPhone can be an effective tool for hacking because of the small size and diverse functionality. Anyone can conceal it in the pocket and hence no one can detect it easily. If an attacker wants to attack a network infrastructure, Nmap in the iPhone can be used to port scan all the connected devices to the wireless network. When the port scanning is complete, the attacker will know which services and operating systems are used by the wireless clients. Then the Metasploit exploitation framework can be used to attack those services and compromise security of the whole infrastructure. iPhone has the Netcat networking tool as well which is normally used as a backdoor for future access to the systems. Once attacker gain access to the systems, she can installed Netcat on those devices

providing a future gateway into the infrastructure as well. All this can be done from the small screen of the iPhone. Furthermore, since iPhone provides a complete shell interface to the user using SSH protocol, all the tasks described above can be automated using common scripting languages such as Python [18]. This way, the whole hacking and cracking cycle can be automated without any user interaction. For example, the user while sitting on the reception of the target organization can connect to an open wireless network, port scan and attack the systems connected to the network, installing backdoors on it and walk away without any detection. This scenario is especially scary because no one can consider a user interacting with the mobile is actually targeting the network through the mobile phone.

The forensic investigator must know that hacking and cracking incidents involve evidence data to be spread and stored on multiple locations such as the wireless AP, the network router, as well as the attacked machine. Evidence needs to be recovered from all locations before a true picture of the attack can be deducted. Since iPhone is the attacking device is such case, recovering evidence from the device will establish the scope from where we need to recover the evidence further. Also, because most of the hacking and cracking tools require root privilege to be executed, if the iPhone is used as an attacking device, it means the device is already jailbreaked and the attacker has root privileges on the device. Therefore, the forensic investigator does not need to jailbreak the device.

A step by step procedure for investigating hacking and cracking incidents committed through an iPhone is described below.

1. Forensic investigator must verify that connection can be established with the OpenSSH server on the iPhone.
2. If connection cannot be established either because the OpenSSH server is not installed on the device or the password is unknown, use the procedures outlined in chapter 5, section 5.3.5 to install OpenSSH server and section 5.3.6 to recover the password of the OpenSSH root account.
3. When the connection is established, check the general.log file in /Library/Log/AppleSupport/ directory to determine the version of the iOS installed on the iPhone
4. Check and extract all subdirectories in /private/var/stash/ directory. This directory contains all the applications installed on the device through Cydia. All hacking tools are available only on the Cydia repository because root privileges are

required to execute such programs and AppStore does not have any applications that violate the Apple Inc terms of service.

5. Extract the Status file in /private/var/lib/dpkg/ directory. This file contain the installation status of all the applications installed through Cydia. If the user has uninstalled the hacking application after using it, the forensic investigator can check this file and detect the installation status.
6. Check the .bash_history file in /private/var/root/ directory. This file contains all the commands executed through the command shell of the iPhone. This will further proof whether any hacking and cracking tool are used on the device, what exact commands are executed, and which are the targets of those commands.

6.4 Conclusion

This chapter provides real world case studies of how the forensic investigation process on an iPhone should be conducted. The chapter provides the step by step solution for two kinds of criminal cases: when the iPhone is used in some traditional crimes such as murder or theft and the device contain correspondence and other evidence related to the crime; and when the iPhone is used in a hacking or cracking incident in an attack against someone's critical infrastructure. In this case the iPhone is used as a hacking device and tools and log data within the device can proof whether the iPhone is indeed used in the crime or not.

Chapter 7

Conclusion and Future Work

7.1 Overview

Since the usage of smart phones in daily life is increasing rapidly, it is important to investigate and analyze all the ways through which the power of these tiny devices can be used. iPhone is the leading smart phone used today which is based Unix operating system and is under the active scrutiny of the security research community. Many popular hacking and cracking tools have been ported to the device and it is reported to be used in some network attack incidents as well. Therefore it is important to analyze all the possible ways in which an iPhone can be used. The forensic investigation community needs to understand the architecture and file system of the device in order to find forensic artifacts stored on the device.

7.2 Objectives Achieved

The iPhone file system has been thoroughly analyzed and all the files have been identified that contain interesting data for the forensic investigation of the device. The iPhone has been analyzed from two angles: for the forensic investigation of police crimes such as murder, kidnapping and theft where evidence and information of the crime is stored on the iPhone; and for the investigation of hacking and cracking incidents where the iPhone is used as the attacking tool for exploitation of network level attacks. The toolkit designed for the extraction of data from the iPhone takes a bit by bit copy of the iPhone hard drive and transfer it to the computer for further analysis. The toolkit takes both the logical as well as physical bit by bit image of the iPhone. This is to ensure that information can be analyzed rapidly from the logically extracted files from the file system of the iPhone in such situations where rapid analysis of data is necessary, e.g. in case of kidnapping. The toolkit then make a complete bit level image and recover deleted data from the device so that not only the data stored on the iPhone file system but also the data deliberately deleted by the user can be recovered and analyzed.

A forensic investigation framework has also been adopted to establish a baseline for the forensic investigation of smart phones in general and iPhone in particular. The

framework can be followed for the extraction and analysis of the evidence as well as effectively reporting the findings so that non technical audience such as a judge or a corporate manager can also understand findings of the investigation. The tool developed as a part of this research initiative partially follows the methodology for the extraction of evidence.

7.3 Future Work

The work done here can be further extended in order to make the toolkit more effective against the latest smart phones. The iOS firmware including 4.0 and later encrypt all the data on the hard drive using hardware and software encryption (depending upon model of the phone). However, some latest research in this area have uncovered that the key for the encryption is hard coded within the phone and can be extracted to decrypted the data. Functionality could be added within the toolkit in order to extract the hard coded key on the fly and decrypt the encrypted data.

Further, currently the toolkit does not have any graphical user interface. Although command line is more suitable because of the speed and reliability of the execution, it requires some familiarity with the command line. Therefore, a web interface for the tool will be helpful where multiple forensic investigators can work on a single case simultaneously by access the interface from multiple locations.

Lastly, reporting capability could be added in the toolkit.

7.4 Summary

Analysis and understanding of smart phones is very important because of the emerging use of these devices in every field. In this research, iPhone has been analyzed for the forensic recovery of evidence and analysis of the recovered data. Since iPhone is closed source, therefore, understanding the architecture and file system of the device is very important to make sure no damage is done to the stored evidence and forensic integrity of the evidence remain intact. All important files on the iPhone file system have been identified so that the forensic investigator knows exactly where a piece of data is stored on the file system. A framework has been adopted so that the whole investigation process is executed in a systematic way and every step is defined clearly. This study concluded with the development of the forensic toolkit that extracts the evidence from the iPhone so that forensic investigator works with a copy of the evidence and the original data remain intact. The tool is

capable of not only recovering data stored on the iPhone file system but also the data deleted by the user. The limitations and missing features of the forensic tool are discussed at the end of the previous chapter. Adding those features within the toolkit will make it more effective for easy extraction and analysis of the evidence data.

Bibliography

1. R. P. Mislan. (2010, Jul.) Cellphone crime solvers. [Online]. Available: http://spectrum.ieee.org/computing/software/cellphone-crime-solvers
2. Hackaday.com (2010) Make iPhone a Penetration Testing Too. [Online]. Available: http://hackaday.com/2010/08/18/make-iphone-a-penetration-testing-tool/
3. Nmap.org (2011). Network Scanner. [Online]. Available: http://nmap.org
4. Metasploit.com (2011). Metasploit Exploitation Framework. [Online]. Available: http://www.metasploit.com
5. Aircrack-ng.org (2011). 802.11 Security Cracker. [Online]. Availble: http://www.aircrack-ng.org
6. Ettercap.sourceforge.net (2011). Man in the Middle Suite on LAN. [Online]. Available: http://ettercap.sourceforge.net/index.php
7. Secmaniac.org (2011). Social Engineering Toolkit. [Online]. Available: http://secmaniac.org
8. D. Compton. (2010, Oct.) Own with an iphone. [Online]. Available: http://www.youtube.com/watch?v=zBm1UXmgz1k
9. K. G. Andrew Hoog. (2009, Mar.) iphone forensic.[Online]. Available: http://viaforensics.com/wpinstall/wpcontent/uploads/2009/03/iPhone-Forensics-2009.pdf
10. S. Institute. (2008) iphone forensic sans. [Online]. Available: http://www.http://files.sans.org/summit/forensics08/PDFs/iPhoneForensics-SANs.pdf
11. J. Zdziarski, iPhone Forensics Recovering Evidence, Personal Data, and Corporate Assets. O'Reilly Media, 2008.
12. Elcomsoft.com. (2011) Elcomsoft phone password breaker. [Online]. Available: http://www.elcomsoft.com/eppb.html
13. J. Schmidt. (2011, Jul.) iopener how safe is your iphone data?[Online]. Available: http://www.h-online.com/security/features/iOpener-How-safe-is-your-iPhone-data-1266713.html

14. D. D. Zovi. (2011) ios 4 security evaluation. [Online]. Available: http://trailofbits.files.wordpress.com/2011/08/apple-ios-4-securityevaluation-whitepaper.pdf
15. (2011, Sep.) Mono. [Online]. Available: http://www.mono-project.com
16. Comex. (2011) Jailbreak me. [Online]. Available: http://www.jailbreakme.com
17. Sogeti Esec Lab (2011, Oct) Analysis of the jailbreakme v3 exploit. [Online]. Available: http://esec-lab.sogeti.com/post/Analysis-of-the-jailbreakme-v3-font-exploit
18. The Space Station Blog (2011, Oct), Python on iPhone & iPad. [Online]. Available: http://the-space-station.com/2011/7/8/python-on-iphone-ipad

Appendix A

View Assigned IP Address

1. Start the iPhone and wait until the default desktop screen (also known as the SpringBoard) is fully loaded.
2. Locate the Settings icon and tap it.
3. In the new screen, look for the Wi-Fi option. In the default iPhone theme, it is usually the second option on the screen. If it is not visible on the screen, scroll down the screen until the Wi-Fi option is visible.
4. If the Wi-Fi is off, tap it so that the Wi-Fi Networks screen gets open. Here, tap the OFF toggle button so that it changes to ON. When the Wi-Fi is ON, it will search for any available wireless network which is broadcasting its ESSID Extended Services Set Identifier (the wireless network name). Select the trusted network that is under the control of the forensic examiner.

   It is very important to connect to a trusted network. Connection to non-trusted wireless networks such as free Wi-Fi hotspots at airport and coffee shops must be avoided at all costs. Transferring sensitive evidential data through non-trusted networks can compromise both the confidentiality and integrity of the forensic evidence.

   It is also important that the forensic investigator should setup a key-protected wireless network so that only authorized devices can connect to it. WPA/WPA2 with AES (Advanced Encryption Standard) encryption and a strong protection key should be selected. WEP should be avoided because of its security weaknesses.

5. Once the iPhone is connected to the wireless network indicated by a tick mark on the selected wireless network, press the right pointing arrow to open the wireless network properties. Here, IP address, subnet mask, default gateway (identified by Router), and DNS server details are visible. A snapshot of the iPhone wireless network details can be viewed below.

Figure A-1 View IP Address in iPhone

Appendix B

Jailbreaking iPhone

iPhone can be jailbreaked using a variety of tools and techniques. However, the jailbreaking process can be roughly divided into two categories: the automated technique and the manual technique. Below are the steps for each one.

Automatic Jailbreak Process

The automated jailbreaking technique is normally used by less technical people because it requires no user interaction. The user just needs to go to http://jailbreakme.com website from the built-in Safari browser in the iPhone and the index.html page of the website exploits vulnerability within the PDF plugin of the browser using a specially crafted html page consisting of JavaScript exploit code [16]. When remote code execution is achieved, the shell code activates the root account within the iPhone to gain full control over the device. It is also important for the forensic investigator to understand what is going on behind the scene in order to fully understand the jailbreaking process as well as to defend the method in a court of law should the need arise in the future. For a full technical understanding of how the jailbreakme exploit work and how remote code execution is achieved on the iOS platform, please refer to [17].

Manual Jailbreaking Process

There are a lot of jailbreak and unblock solutions available for iPhone. Blackra1n, GreenPois0n, Limera1n, Purplera1n, QuickPwn, Redsn0w, Blacksn0w, Sn0wbreeze, and Winpwn are some of the most popular jailbreak and unlock solutions. The process of jailbreak exploits vulnerability within the iOS and gain root privilege over the device. Unlocking is the process of exploiting vulnerability within the baseband of the device to make it operate with incompatible GSM and CDMA networks. Majority of the tools mentioned above can perform both jailbreaking and unlocking of the device. Redsn0w and Limerain/Blackra1n/Blacksn0w should be used for jailbreaking because these are the most reputed solutions for jailbreaking and unlocking the device. Below is the step by step procedure for how to jailbreak the device using the popular redsn0w jailbreak tool from the iPhone Dev Team.

1. As specified in section 2.3, the forensic investigator should have all the iOS firmware versions stored offline. If it is not stored offline, download the original iOS firmware version installed on the iPhone and store it offline.
2. Launch the Redsn0w jailbreak tool. It is available for both Mac and Windows operating systems.
3. Select the downloaded iOS file. iPhone iOS firmware is a proprietary IPSW file format.
4. Select Cydia to be installed with the firmware as well. This option is important to ensure that if the forensic investigator needs to install some other tools such as OpenSSH or command shell on the iPhone device, it can be easily done through the Cydia repository.
5. When the custom IPSW file is created, restart the iPhone device is DFU (Device Firmware Upload/Update). The procedure for DFU mode is simple:
   i. Restart the iPhone
   ii. When the device screen turns black, hold down the wakeup/power and home button simultaneously for exactly 10 seconds
   iii. After 10 seconds, release the wakeup/power button but keep holding the home button.
6. Start the iTunes. A message will be displayed on the screen indicating that an iPhone in recovery state is detected.
7. Hold down the shift key on the PC keyboard (Option key on Mac) and select the IPSW file created by the Redsn0w.
8. Wait for the installation to complete.

The forensic investigator also needs to understand the underlying process of how the jailbreaking process actually words under the hood. The process of a normal boot sequence within the iPhone is shown through the figure below.

Figure 1- 3 iPhone Jailbreaking Process

The first step of the booting process is to initialize the Boot ROM. The purpose of the Boot ROM is like a computer BIOS. Then next step is to load the Low Level Bootloader (LLB). LLB is responsible for transfer of low level memory and content transfer between the iPhone RAM and flash memory. Next is the iBoot. iBoot is used for security within the boot process. It checks the integrity of the firmware that is being loaded in the next step. The final step is to load the user applications.

Jailbreak process patches the iBoot so that the integrity and security checks are not performed and hence a custom firmware is loaded on the device.

Appendix C

Check the default SSH port

OpenSSH server runs on the default TCP port 22. The forensic investigator can check the iPhone for the TCP port 22 and if the port is open, it can be reasonably assumed that the OpenSSH server is installed on the device. To check TCP port 22 on the iPhone, follow the procedure given below:

1. Obtain the IP address of the iPhone described in the section 5.1.
2. Connect a computer to the same network to which the iPhone is connected. The computer can be connected to any available network as long as there is no firewall restriction between the iPhone and the computer system. However, for reliable and fast port scanning, both the devices (iPhone and computer) should be on the same network.
3. Install a port scanner utility on the computer. The most popular port scanner is Nmap which is an open source port scanner available at http://nmap.org. Here, we will assume that the forensic examiner is using Nmap for port scanning.
4. Run the following command from the command window:

   nmap –p 22 <IP address of the iPhone>

For example, if IP address of the iPhone is 192.168.10.104, the forensic investigator can use the following command to check if TCP port 22 is open on the iPhone:

nmap –p 22 192.168.10.104

If the result shows port 22 as open, it means OpenSSH is installed on the iPhone.

Appendix D

Check OpenSSH Server through Cydia

The forensic examiner can also check the OpenSSH server through the Cydia repository. To check the installation status of the OpenSSH server through Cydia, follow the procedure given below:

1. Locate the Cydia App icon on the desktop SpringBoard and tap to launch it.
2. Press the Search button on the bottom strip and open the search screen once the Cydia App loading is complete
3. Tap the search dialogue box and write OpenSSH in it. The search is case insensitive.
4. When the OpenSSH package is located, if a green ticking arrow is visible beside the package name, it means it is installed on the iPhone.

Appendix E

Installing OpenSSH on iPhone

If OpenSSH is not installed on the iPhone, follow the procedure below to install it:

1. Make sure the iPhone is jailbreaked. If iPhone is not jailbreaked, please follow the procedure outlined in section 5.2.
2. Locate Cydia App icon on the desktop SpringBoard and tap to launch it.
3. Press the search button in the strip below to open the search dialogue box and search for OpenSSH. Search is case insensitive so case of the words does not matter.
4. Once the OpenSSH package is located, press it and confirm the installation to install it.

Appendix F

Resetting OpenSSH Password

By default, the OpenSSH server installed on the iPhone has the username “root” and password “alpine”. The forensic investigator can use these credentials for the logical and physical acquisition. However, if the user has changed the password of the OpenSSH server, the forensic investigator has two choices: either reinstall the OpenSSH server or use the following procedure to reset its password:

1. Locate and tap the Cydia App repository icon on the desktop SpringBoard.
2. Go to Search dialogue by pressing the Search button on the bottom stip.
3. Locate a package named iFile and install it. iFile is a graphical file manager which can be used to navigate and edit the whole file system of the iPhone using an easy to use graphical user interface.
4. Launch the iFile application
5. Go to /etc directory and locate a file called master.passwd. The master.passwd file is the Linux equivalent of /etc/shadow file and contains password hashes of all user accounts configured on the iPhone.
6. Press the Edit button in the top right corner of the iFile app. It will change the display interface.
7. Select the master.passwd file by tapping on it. When the file is selected, a red dot sign should appear on the left side of the file.
8. Once the file is selected, press the Cut/Copy button on the button right side and select Copy.
9. Go to any other location on the file system, press Edit and then press the Paste button. It will paste the file there.
10. Press the master.passwd file that just got copied and press edit. This will open the master.passwd file in edit mode so the forensic investigator can edit the password hashes.
11. When the file is opened in iFile, it has the username and hash of the password stored within it. Passwords in iPhone are hashed using the crypt function normally available in Linux libraries. A lot of online websites are also available for generating crypt hashes of text. Locate any resource and generate crypt hash of a known text.

12. Once hash is generated for a known text, copy the hash over to the master.passwd file and replace the old hash with the new one. For example, to reset password for the root account, locate the root account within the iFile and replace the hash with the newly generated hash that the forensic investigator generate from a known text.
13. Save the file and replace the old master.passwd file with the edited file.

To make sure nothing goes wrong, don't delete or replace the old master.passwd file. Rename it to master.passwd.old and paste the edited file in /etc directory. This will ensure that if anything goes wrong, the forensic investigator can revert back to the previous settings.

14. Now login to the OpenSSH server using the root username and the known text from which the hash is generated.