© 2018 IJSRST | Volume 4 | Issue 9 | Print ISSN : 2395-6011 | Online ISSN : 2395-602X
Themed Section: Science and Technology

Logical acquisition of iPhone without Jail Breaking

Priyank Parmar[1], Dr. Ravi Sheth[2]
*[1]MTech, Cyber Security, Department of Information & Technology, Raksha Shakti University, Ahmedabad, Gujarat, India
*[2]Assistant Professor, Department of Information & Technology, Raksha Shakti University, Ahmedabad, Gujarat, India

ABSTRACT

With the tremendous growth of technology use of Smart phones are very common, but it is a bit complicated from the Forensic point of view. The various versions of operating system of device and manufactures customization leads the smartphone to the complication. Every manufacturer has its own security algorithm and own terms while designing a Mobile Operating System. Day by day level of encryption is increasing the complexity for the forensic expert during investigation. The iPhones and iPads are using iOS which has taken care of Users Privacy and Security on the top level of its architecture. Investigating such devices are very complicated as the every file is kept encrypted on the device. Cyber criminals are using latest gadgets to commit such crime and it will increase in near future. Author has tried to cover the forensically acceptable methods which can be achieved on an iOS Device.

Keywords : iPhone Forensics, iOS 11, Data acquisition without Jailbreak, logical acquisition of iPhone without Jailbreak

I. INTRODUCTION

Technology in smart phones is rising at tremendous way. Each and every release by Apple INC is containing a newer and secure technologies with the large data storage space. Email, SMS, Chat Messaging, Camera Images and Videos Calendar, Notes, Web Browsers and many more may contain some importance evidence. Author has analysed various artifacts retrieved from the device during forensic investigation. Author has performed logical methods of data extraction during the research to identify and analyse the valuable evidence. Retrieved data is in completely readable format and no jailbreak or rooting techniques has been applied on the iOS device. Author has follows the 6 general steps to perform forensic investigation process on the iOS device.

II. EXSISTING WORK

This section of research paper includes existing technology and methods available to perform the forensic analysis of the iOS based smart device. Apple has launched APFS (Apple file system) in version software version 10.2 or later one and discontinue with HFS (Hierarchical File system). To perform the forensic analysis, expert has to understand its file system and its architecture.

III. The Apple File System (APFS)

APFS is a new file system for every Operating system developed by Apple i.e. iOS, macOS, tvOS, and watchOS. It's a 64-bit file system which supports more than 9 quintillion files on a single volume. APFS is

structured in a single container, which may contain one or more volumes. [1]

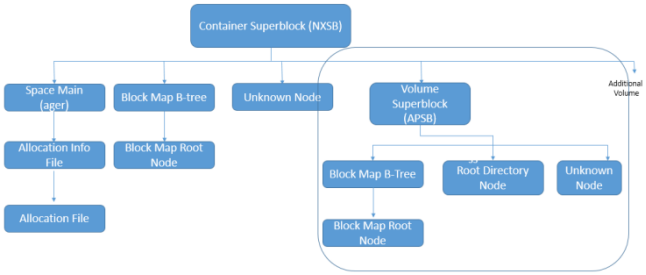


Figure 1. APFS File Structure.

Each element of this structure (except for the allocation file) starts with a 32 byte block header, which contains some general information about the block. The container superblock contains information on the number of blocks, the block size, and pointers to the space manager for this task, the block IDs of all volumes, and a pointer to a block map B-tree (contains entries for each volume with its ID and offset). Nodes are used for storing different kinds of entries. It can be part of a B-tree or exist on their own, and can either contain flexible or fixed-sized entries. The space manager manages allocated blocks in the APFS container, and stores the number of free blocks and a pointer to the allocation info file. The allocation info file stores the allocation file's length, version and the offset.

B-trees manage multiple nodes, and contain the offset of the root node. A volume superblock contains the name of the volume, an ID and a timestamp. As for allocation files, they are simple bitmaps, and do not have a block header and type ID.

Default, System is having two logical partition i.e. System Partition and User data Partition. The system partition contain all the system information which are pre-installed application by the iPhone. The System partition is updated only when a firmware upgrade is performed on the device. As the system partition was designed to remain in factory state for the entire life of the iPhone, there is typically little useful evidentiary information that can be obtained from it. However this require to jailbreak the iPhone and which is not advisable.

The Data Partition contains all variety of data which the user has stored or used. The third party application data, contact, Chat messages, attachment everything is stored here and which plays important role for the forensic investigator for the detailed study of the case. The major portion of NAND is being used by this partition and this is mounted at /private/var on the device. As said this contains almost all information of the user we can go with the logical acquisition of the device and can get the almost all evidence.

IV. The iOS Architecture

iOS plays an intermediary role between the underlying hardware components and the applications that appear on the screen. The application needs to go through this communicate through a well-defined system interface that protects the applications from hardware changes. This abstraction makes it easy to build applications that work on devices with different hardware capabilities. The iOS architecture comprise of four layers—the Cocoa Touch layer, Media layer, Core Services layer, and Core OS layer—as shown in the following figure. Each layer is comprise of several frameworks that helps to build an application.

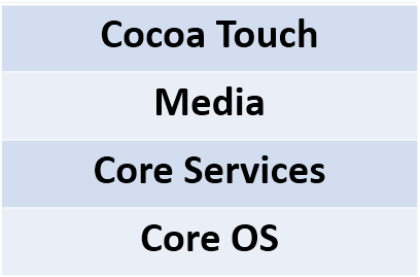


Figure 2. iOS Architecture.

The Cocoa Touch layer contains the key frameworks required to develop the visual interface for iOS applications. Frameworks in this layer provide the

basic application infrastructure and support key technologies, such as multitasking, touch-based input, and many high-level system services. The Media layer provides the graphics and audio and video frameworks to create the best multimedia experience available on a mobile device. The technologies in this layer help developers to build applications that look and sound great. The Core Services layer: The Core Services layer provides the fundamental system services that are required for the applications. Not all of these services are used by developers, though many parts of the system are built on top of them. This layer contains technologies to support features such as location, iCloud, and social media. The Core OS layer: The Core OS layer is the base layer and sits directly on top of the device hardware. This layer deals with low-level functionalities and provides services such as networking (BSD sockets), memory management, threading (POSIX threads), filesystem handling, external accessories access, and inter-process communication.

V. SQLite Database

SQLite is an open source, in-process library that implements a self-contained, zero-configuration, transactional SQL database engine. This is a complete database with multiple tables, triggers, and views that are contained in a single cross-platform file. As SQLite is portable, reliable, and small, it is a popular database format that appears in many mobile platforms like android operating system.

Most of smartphones are using SQLite database to store data. The applications like messages, phone, SMS, WhatsApp, calendar are storing data in SQLite database. To open this kind of database a forensic investigator may need to use SQLite Browser. [5]

VI. Plist

Plist is used as abbreviation for Property list/file. Earlier Apple was using NeXSTEP formats, however this was discontinued and new XML format i.e. plist was introduce. Tools like Plutil or Strings may be used to view the content on Windows or Linux system. [5]

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/
DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>EthernetBuiltIn</key>
        <string>y</string>
        <key>Timeout</key>
        <string>5</string>
        <key>ForceHPET</key>
        <string>n</string>
        <key>GUI</key>
        <string>y</string>
        <key>Default Partition</key>
        <string>hd(0,2)</string>
        <key>Graphics Mode</key>
        <string>1024x600x32</string>
        <key>GraphicsEnabler</key>
        <string>n</string>
        <key>Kernel</key>
        <string>mach_kernel_atom2</string>
        <key>Kernel Flags</key>
        <string>arch=I386</string>
</dict>
</plist>
```

Figure 3. Screenshot for plist

Acquisition

There are two different methods of mobile device data extraction: logical and physical. In this paper Logical extraction method is covered and Physical extraction requires jailbreak and which is not advisable. The steps to be taken care the Investigator should get the passcode from the User and if the device is in unlocked condition Investigator should disable the passcode requirement. Auto lock feature should be change to “never” so that the device will not get locked automatically. The one more important step is to put phone into Airoplane mode, which prevent iPhone being connected remotely or wiping device remotely.

VII. Forensic Methodology

A. Logical Acquisition using Customized Tool

It’s a Linux based tool which includes certain libraries that talks the protocols to support iPhone®, iPod Touch®, iPad® and Apple TV® devices. The tool has been developed using python programming platform. Python is open source cross-platform framework

which supports windows and linux environment. This tool runs on the iDevices without Jailbreaking. It allows other software to easily access the device's filesystem, retrieve information about the device and it's internals, backing up the device, retrieve address book/calendars/notes and bookmarks synchronize music and video to the device.

The backup taken by the customized tools contain various interesting files of evidence all filename is SHA1 hash value of the original filename. The root directory contain four files for every device i.e. status.plist, info.plist, Manifest.plist and Manifest.db. These files contains all basic information about device and backup.

B. **Files need attention**

As said above, the all interesting data are stored either in sqlitedb or plist. This data can be analysed for getting a evidence. The two examples for the accessing such file are explained here.

1) Safari Browser

The Safari browser is pre-installed on Apple device which allows users to bookmark their favorite websites. The bookmarks database is a HomeDomain file, which can be found at */private/var/mobile/Library/Safari/Bookmarks.db*. The Safari browser stores the recently downloaded and cached data in a database. The database is a HomeDomain file and it can be found at */private/var/mobile/Library/Caches/com.apple.mobilesafari/Cache.db*.

The file stores cached URLs and the web server's responses along with the timestamps. Apart from this, Safari stores information from various sites in the WebKit database that is located in the */private/var/mobile/Library/WebKit/LocalStorage/directory*.

2) RootDomain plist Files

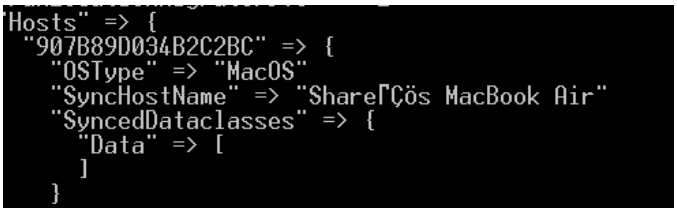
The following RootDomain files listed should be examined for relevance to your investigation:

/private/var/root/Library/Preferences/com.apple.preferences.network.plist

Status of Airplane mode can be retrieved from the following file:

/private/var/root/Library/Preferences/com.apple.MobileBackup.plist

It contains the timestamp of when the device was last restored from the backup, the device build version, and the backup build version



/private/var/root/Library/Caches/locationd/clients.plist: This contains the location settings for applications and system services

/private/var/root/Library/SystemConfiguration/com.apple.wifi.plist This contains the all information about Wifi connected last time, data usage, last password updated, BSSID, SSID.

VIII. Conclusion and Future work

It is important to note, that the some expensive commercial tools are also not capable to retrieve such information from iOS device without jailbreaking. Author has found some very important and crucial information from the iDevice using developed tool without jailbreaking shown in Table 1. There is no question of integrity here so for forensic point of view this methodology should be adapted in future. Findings could be represented in an attractive manner using some GUI application and reports will be prepared with hash value. This artifacts should be valid and liable to represent in judicial procedure.

IX. REFERENCES

[1] "Whitepaper of iOS Security Guideline from Apple" https://www.apple.com/business/docs/iOS_Security_Guide.pdf (Last Accessed: 26th April 2018)

[2] Guidelines on Mobile-Device Forensics, NIST Special Publication 800-101

[3] Kyle D. Lutes, Richard P. Mislan, "Challenges in Mobile Phone Forensics"

[4] Moore, Jason; Baggili, Ibrahim; and Breitinger, Frank, "Find Me If You Can: Mobile GPS Mapping Applications Forensic Analysis & SNAVP the Open Source, Modular, Extensible Parser," Journal of Digital Forensics, Security and Law: Vol. 12 : No. 1 , Article 7

[5] Tim Proffitt," Forensic Analysis on iOS Device, SANS Institute Author"

[6] Agrawal, V, Bhattacharyya, C, Niranjan, T & Susarla, S, 'Discovering Rules from Disk Events for Predicting Hard Drive Failures', In International Conference on Machine Learning and pplications (ICMLA 2009), pp. 782–786, IEEE Press, 13–15 December 2009.

[7] Hoog, A & Gaffaney, K (2009), iPhone Forensics, via Forensics