Towards Live Forensic Acquisition & Analysis Of Mac OS Systems

Eby Prasad
ER & DCI Institute of Technology
Trivandrum, India
ebyprasad@gmail.com

Dija.S
CDAC
Trivandrum, India
dija@cdac.in

Dhana Lakshmi.M.P
ER & DCI Institute of Technology
Trivandrum, India
dhanalakshmi@cdac.in

***Abstract*—The Mac Operating System is a fork of UNIX OS that is widely used in Apple's computers known as Mac Books. The availability of tools for macOS forensics are in scarce. In addition to this, the macOS provides extra security to the user, which in turn leads to difficulties in forensic analysis. Nowadays the macOS is gaining popularity among cyber criminals because of its unique features. Hence there is a need for analyzing mac Systems in depth. The artefacts left out by the suspect must be retrieved and analyzed as part of cyber forensics investigation. The data from these artefacts are analyzed based on its forensic relevance. This paper presents a methodology for live forensics in macOS system. This involves acquisition of crucial artifacts from a suspect's machine and analyzing it later in an investigator's machine.**

Keywords— Mac, OS, forensics, live, acquisition, investigation, evidence,artefacts

I. INTRODUCTION

Cyber Forensics is the application of investigation and analysis techniques to gather and preserve evidence from a computing device in a way that is suitable for presentation in a court of law. Nowadays more and more cybercrimes are done on MAC machines compared to other Operating Systems. Mac is considered as a highly secure OS that is focused around user's privacy which is provided by the hybrid Darwin kernel of macOS which is forked from the Apple's old XNU project. Since MAC is tough to penetrate and gives high priority to user's privacy, the criminals make use of this to hide themselves and attack others[1]. Further, this same feature creates a problem for conducting forensics in a MAC machine. There is a lack of sophisticated tools available for carrying out forensics on a MAC machine especially after the release of macOS X version [2].

The macOS (previously Mac OS X and later OS X) is the current series of Unix-based graphical operating systems developed and marketed by Apple Inc. designed to run on Apple's Macintosh computers. It has been preinstalled on all Macs since 2002 [3]. Within the market of desktop, laptop and home computers, and by web usage, it is the second most widely used desktop OS after Microsoft Windows. It includes heavily used several programs by default, including Apple Mail, Apple Safari Web Browser, an Apple Address Book, and iCal [4].

Since macOS is a fork of the famous UNIX operating system, we can analyze various forensically relevant artefact locations like in other OSs. The uniqueness of Apple's macOS is the high-end security constraints included in it. These make the macOS forensics harder when compared to other OSs. The documentation materials and tools available for forensics of macOS are limited since it is hard to penetrate an apple MacBook. It is high time to fill the gap of this shortcoming and develop an all in one tool which could help the cyber forensic investigator in conducting analysis on a macOS seamlessly.

II. LITERATURE SURVEY

In the book named 'OS X Mountain Lion Pocket Guide', Chris Seibold covers all the basics of OS X mountain lion edition which is the previous version of macOS high sierra. The iCloud integration was first done in this edition. The features of the OS are explained well in this book along with the various services that are critical to the OS to execute properly [1]. Many troubleshooting guides are also included in it which helps users to quickly rectify an error or to recover from one. The basic terminal commands are also explained in the book that will help the investigator gain more information about the system.

The Forensic software support for OS X remains less mature than that of Windows or Linux. This is explained well in the paper OS X as a Forensic Platform – David M. Martin [2]. While many Linux forensic tools will work on OS X, instructions for how to configure the tool in OS X are often missing or confusing. The methods to configure tools and run it easily are explained here. Many tools for forensics and the data to be retrieved are mentioned with a brief explanation.

978-1-5386-4966-4/18/$31.00 ©2018 IEEE

The various methods and procedures to collect evidence from an UNIX system are explained in detail in the paper Electronic evidence collection and analysis based on UNIX system – Chen Xian gui, Hong Xiujuan & Wang Xin. Some basic commands used in terminal are mentioned [3]. Since the macOS is similar to the UNIX system, there are some similar locations from which various artefacts can be collected. The judicial data collection procedures are explained as the main concern in this paper.

III. Current Scenario

There is a lack of sophisticated macOS forensics tools currently available in the market. The two of the current tools used for macOS forensics are Macquisition by blackbagtech[9] and OSXCollector by Yelp. These tools are still in its nascent stage, lacking some crucial elements and features. Apart from these two tools, there is no other sophisticated tools available for conducting forensic analysis on a macOS machine. Hence it is high time to develop a sophisticated tool that could perform forensic analysis on macOS machines including all crucial features and elements.

IV. Proposed System

The proposed system contains two parts. The first part deals with the acquisition of various artefacts from a macOS machine. The details are collected and stored in a file folder structure. The artefact files are then populated in a user friendly graphical interface for analysis. The second part consists of analyzing the details acquired from the suspect machine to prepare detailed report. This is usually done in the investigator's machine.

The system is purely built in python language since the macOS has native python support by default and to minimize tampering. The acquisition tool collects the information by calling the required python scripts that in turn queries the system functions and libraries and stores the details to a USB device. The analysis tool, reads the output files, analyze it and finally displays it in a user-friendly manner to the investigator and prepares a report.

The acquisition tool has the interface to key in the crime details, investigator details and suspect details. This information will be stored along with the artefacts collected for preparing the crime report later. The acquisition tool can select the required module as per the investigators needs. The tool then collects the artefacts and store them in a particular folder with the crime number.

The analysis tool is run on the investigator's machine and the artefact files available in the USB device. Then, it will be parsed and populated in a user-friendly manner. The investigator can then analyze the artefacts and correlate the occurrence of events and finally generate a report from it.

The investigator plugs in the USB drive containing the acquisition tool with a write blocker, which will prevent tampering of the system while running the tool. The artefacts will be collected in the same USB drive itself. The USB drive is removed once the acquisition process is completed. The USB drive with the acquired artefacts is then plugged into the investigator's machine. Then the analysis tool is executed for carrying out further analysis and detailed report generation.

A. Objectives Of The Proposed System

The various artefacts retrieved are:

- Basic System Configuration Information
- User Details of all user accounts
- Running Processes at the time of seizure
- Network Details of the live system
- Installed Applications in the system
- Storage Details of the system
- Open Files during the time of seizure
- The Shell History from terminal
- Event Logs of the system

The basic diagram of the proposed system is shown in Fig.1 below which shows the main functionality of the system as a whole. The main tools as well as their functionality are also shown in the diagram.

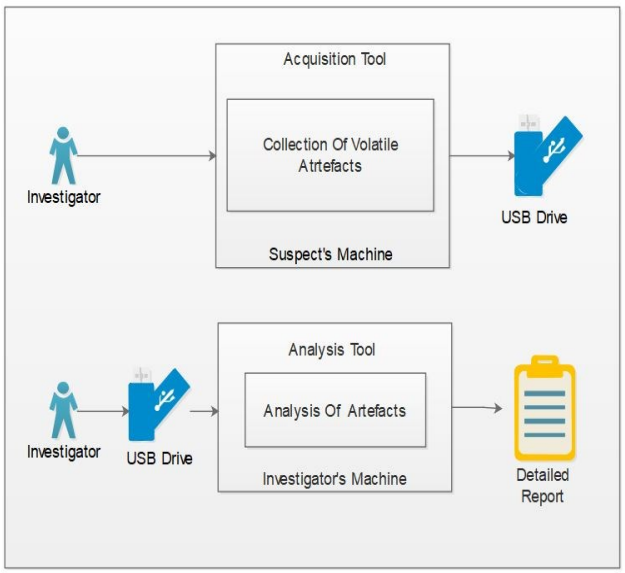

Fig.1 Proposed System

B. Algorithm

Step1: Select the required artefact extraction modules from a list of available modules.
Step2: Execute the selected modules to start the artefact extraction and retrieval.
Step3: Store the retrieved artefacts in a file folder structure which is located in the removable media.
Step4: Repeat step2 to step3 until all the required artefacts are collected.
Step5: Disconnect the removable device from suspect machine and re-connect it to the investigator's machine.
Step6: Execute the analysis tool for previewing the artefacts.
Step7: Select the required artefact to analyse by clicking on its name.
Step8: Analyse the artefact file from the removable drive.
Step9: Reconstruct the forensically relevant information from the file and display it in the viewer.
Step10: Generate a detailed report at the end of analysis.

V. Analysis

The various artifacts that can be collected from the MAC system are:

A. Basic System Information:

This involves the information about the system platform, OS version, OS fingerprint, Kernel name and Processor Name. This will help the investigator in knowing the main hardware and system software version installed in the suspect machine.

B. Event Logs:

The event logs must be analyzed to get information about the various events happened in the machine such as boot time, processor name, time at which certain events occurred, type of process and user name [5]. This information will help the investigator in finding out the processes that were running during the time of seizure and in correlating the events to reconstruct the possible crime scenario. The process type gives information on whether the user have started the process or not. The boot time is very useful in helping the investigator to further deepen the analysis around a particular time frame.

C. Installed applications

The installed applications in the MAC machine must be retrieved in order to analyze the various tools used by the suspect to commit the crime. It can be identified by their package name such as com.apple.pkg.x. Where x denotes the real application's name displayed by the operating system [10]. When the user installs an application, it will be registered in the macOS profiler. When we query the profiler, the application packages will be given back to us. The investigator will benefit from this information by knowing all the user and system application packages installed in the machine. This is useful when the investigator needs to target a particular tool or set of tools that could be used by the suspect in carrying out the crime.

D. Network Information

The network info can be obtained from the macOS machine which includes all the network adapter details, ip address, mac address, network status and mtu packet size [8]. These details will help the investigator to find out the network to which the suspect has connected, the network adapter used and to determine whether the mtu is tampered with for fragmentation attacks.

E. Open Files

From a macOS system, open files during the time of acquisition along with the details such as PID, user who ran it, file descriptor, file type, file size, inode entry, location and commands used to run it, can be retrieved. The investigator will benefit from this information for determining which files were used by the suspect recently and to locate possible file artifacts that would be of prime importance in correlating the events and proving the evidence.

F. Running Processes

The running processes at the time of acquisition along with their PID, terminal number, time at which it ran and location of the executable can be retrieved [7]. This information is crucial for the investigator in determining the applications used by the suspect during the time of seizure which might lead to the main evidence of the crime committed. The running processes could be anything ranging from a single program or a set of programs. The list of running process retrieved in this way may include a malware.

G. Storage Details

The various details about the storage such as file system, total size, used capacity, available capacity, inodes used, inodes free and mount locations are collected. These details will enable the investigator to conduct analysis on a particular mount point which may be hidden by the suspect. The investigator can cross check the total size of the storage media against the one displayed in the MAC's finder application.

H. Terminal History

The terminal history can be a good clue to know what the suspect did with the MAC machine. Thus, the terminal history must be collected. The terminal command history is useful in determining what the suspect did with the machine recently [6]. The terminal history can also be used to prove the various applications installed via side loading that are not in apple store. Harmful applications (potential hack tools) are usually not available in apple store. So, the suspect must install then via the terminal to use them.

I. User Details

The user details such as current user name, home directory and host name can be obtained along with other users of the machine, their UID, home directory and privilege [9]. The current user and his permission levels will be useful in determining what type of applications are run by the suspect since many hack tools require root privileges. The investigator can narrow down the search to a particular area by knowing these kinds of details during analysis.

By collecting all these artefacts from the live machine, the investigator can reconstruct the state of the system at the time of seizure. The running processes present in the system at the time of seizure is very important to find out what happened and correlate the various events that might have occurred in the machine. Live forensics is important than offline forensics since the latter forensics there could be a huge amount of processes, which are not relevant to the current state of the system, which makes the forensics process tedious. In live methodology, the relevant running processes and associated details can be targeted easily and analyzed for finding evidence of committed crime. It could also point out potential malicious processes or applications that may be used by the suspect to commit the crime.

VI. Implementation Details

The acquisition tool has a GUI for entering various case related details such as investigator's name, investigator's rank, police station, crime number, seizure memo number, place of seizure, date of seizure, time of seizure, name of suspect, address of suspect and details about two witnesses. The second page consist of artefact selection where we can select the required artefacts to be retrieved from the system. The analysis tool is used at the suspect's machine running on macOS.

The analysis tool is used at the investigator's PC to analyze the artefacts retrieved from the suspect's machine. The analysis tool GUI has the tree view of the main case file containing the retrieved artefacts. A hash generator is provided to generate the hash of artefact files in order to check its integrity after acquisition. Each type of artefact can be viewed in the analysis tool by selecting the required one. The artefacts can be sorted in different ways. Finally, a report is generated.

VII. Challenges

One of the main challenges faced during the implementation of this methodology is that, few of the forensically relevant information can be acquired only if the user credentials are known. Terminal history can be forcefully disabled by the user in a macOS system and there is no other way to acquire system details other than calling the system libraries. To keep the tampering to a minimal level, an inbuilt programming language such as objective C or Python 2.7 must be used for retrieval of artifacts from the system.

VIII. Conclusion

Since the macOS systems are widely used for criminal activities when compared to other OSs, there is a need for conducting deep analysis of the machine. The availability of tools for macOS forensics are scarce in number. Live forensics is more useful in correlating evidence than offline forensics. Thus, there is a need for a sophisticated tool for conducting live forensics in a macOS machines. In addition to this the macOS provides extra security to the user, which in turn leads to difficulties for forensic analysis. The artefacts left out by the suspect can be retrieved and analyzed through the methodology described in this paper. This provides highly crucial evidence in a cyber forensics investigation.

Acknowledgment

The authors wish to thank ER & DCI Institute of Technology for providing an opportunity to undertake research on Live Forensics of macOS.

References

[1] Chris Seibold, "OS X Mountain Lion Pocket Guide", 2016

[2] David M. Martin, "OS X as a Forensic Platform", 2016

[3] Kenneth M. Ovens and Gordon Morison, "Identification and Analysis of Email and Contacts Artefacts on iOS and OSX", 2017

[4] Chen Xian gui, Hong Xiujuan & Wang Xin, "Electronic evidence collection and analysis based on UNIX system ", 2015

[5] Ashley Hammer,"Mac OS High Sierra", 2017, https://www.macrumors.com/roundup/macos-10-13

[6] Charlie Miller, "Owning the Fanboys: Hacking Mac OS X", 2017, www.blackhat.com/presentations/bh-jp-08/bh-jp-08-Miller/BlackHat-Japan-08-Miller-Hacking-OSX.pdf

[7] M. Geralds," The Secret Guide to High Sierra Sensitive Information Retrieval", 2017, www.25e9ytrvsajryd673nhdfema865ojiq.onion/secrets/apps

[8] H.Zhang, "The New Mac OS X High Sierra", 2017, https://www.apple.com/macos/macos/high-sierra

[9] "Macquisition-Acquisition tool for mac", 2017, https://www.blackbagtech.com

[10] Dave Grohl, "Macintosh Operating Systems", 2016, www.wikipedia.org/wiki/Macintosh_operating_systems