

Mac/Unix Forensics [Industry Project]

by

Abhishek Gaur
MSc Cybersecurity



Supervisor of Dissertation

Dr. Eirini Anthi, Lecturer in Cybersecurity, Cardiff University

Industry Supervisor

Molly Betts, Digital Forensics and Incident Response Specialist
BAE Systems Applied Intelligence

2022

Contents

APFS	5
FileVault	5
T2 Security Chip	5
Apple Silicon (the M series)	6
Unified memory	6
Disk Arbitration	6
How to turn off FileVault	10
What format should be the image file	11
Customer Data Migration Tool	12
Target Disk Mode	13
dd, dcfldd and dc3dd	13
Imaging the M1 Mac	16
Differences	23

Acknowledgement

I would like to express my deepest gratitude to my supervisor Dr. Irene Anthi, who supported me at every point and also gave me the freedom to set my own goals in this project. I would also like to thank my industry supervisor Molly Betts for the initial assurance that I was going the right path when I felt I there was no way.

I am in debt of the graciousness of Steve Whalen, CFCE, Co-Founder - SUMURI LLC for his insights and my friend Vikas for his emotional support.

A big thank you to Cardiff University for providing me the opportunity to study Masters at this prestigious institution.

Lastly, I would thank my father, who believed in me even when I didn't believe in myself.

Abstract

The process of data acquisition and data analysis in Windows OS and Linux OS has seen an incremental progression. Premium software's and tools have been built to deal with the intricacy and problems that come when trying to extract and analyse the data. However, Mac OS has not seen the same progress. There are only a few tools that are specifically designed for data analysis and acquisition in Mac OS. The information regarding how to perform data acquisition, the variables that can influence the acquisition process and where to look for key evidence after acquisition is missing or unclear. This project sheds light on the factors that complicate the data acquisition process in Mac OS and then develop the most efficient method to perform this process. It further provides a comprehensive analysis of the available tools in the market to perform data analysis on the extracted data from a Mac OS.

The main objective of this project is to provide a method to perform key digital forensics on MAC OS.

Introduction

Since the launch of the first Macintosh along with the 'classic' Mac OS in 1984 [1], the Mac OS has gone through major development through the years with hardware changes to complement those updates. The Mac OS utilizes the XNU kernel [2] and its core set of components are based on Darwin [3], which is an open-source Unix-like operating system. Darwin is composed of code derived from NeXTSTEP [4], BSD [5], Mach [6], free software code and code developed by Apple.

Mac OS has not seen the same development in digital forensics as its counterparts Linux and Windows OS [7]. The hardware and software components of the Mac add on to the challenges faced in digital forensics in MAC.

APFS

Apple File System (APFS) is the file system used in Macs since Mac OS Sierra (10.12.4) [8]. APFS replaced the Mac OS Extended Filesystem (HFS+) [9] since APFS supported Snapshots, Data Checksums, concurrent access among many other features that were missing in Mac OS Extended [10].

The GUID Partition Table (GPT) [11] is utilized in APFS. At least one container is present under the GPT partition scheme.

APFS Volumes are present inside a container. The APFS Volumes inside an APFS container can grow or shrink on demand basis and the free space in a container is shared by APFS volumes within that container [12].

If there is only one internal APFS disk with a single user volume on it, a synthesized disk is created which stores important APFS Volumes such as Preboot, Recovery, VM, Macintosh HD, Macintosh HD - Data and other user created APFS Volumes [12].

The File System is important to understand to know where and how important data is stored on a Mac OS.

FileVault

FileVault 2 (now referred to as just FileVault and based on the original FileVault) is an in-built encryption software available in Macs since OS X Lion 10.7.5 [13]. FileVault provides full-disk encryption on Macs and uses the XTS-AES-128 encryption [14] with a 256-bit encryption key, which falls in line with NIST guidelines for using XTS-AES mode for protecting data confidentiality on storage devices [15].

T2 Security Chip

The T2 security chip is a second-generation custom silicon for Mac and introduced and available in all Macs since 2018 [16]. It provides data encryption and secure boot capabilities using hardware-accelerated AES engine built into it. The 256-bit keys tied to a unique identifier inside the T2 chip are used for encryption. The advanced encryption technology embedded in the T2 chip provides line-speed encryption [17].

Apple Silicon (the M series)

In November 2020, Apple discontinued the intel-based Macs and transitioned to the M series processors designed by the company itself, instead of relying on intel for their processors [18]. The M series processors are ARM based system on a chip (SoC) where the important components such as CPU, GPU, unified memory (RAM) and neural engine among others are available on the same chip [19]. Even the T2 security chip is integrated into the same chip instead of being present as a separate chip, making it an integral part of the processor itself.

Unified memory

With the launch of the M series SoC processors, slot-based RAMs that fit into the motherboard were also discontinued to be replaced by unified memory which is part of the chip on which the processor rests. It is termed as unified memory since instead of dedicating different portions of the RAM to various tasks, the same memory is used by GPU and CPU operations, the need to copy data from different sections is eliminated, making the process faster. Since no memory is reserved for any task, the allocation can be ramped up on a need basis by the processor [20].

Disk Arbitration

Disk Arbitration is a framework in Macs which detects mounts, unmounts, change mount points, mount with different flags or detect volume name changes [21]. This framework becomes crucial when a new drive is connected to a Mac during the data extraction process.

Aim and Objectives

The aim and objectives of this project can easily be encapsulated into these points

- Analyse the current imaging process
- Fill the gap of unclarity of instructions when it comes to imaging in Mac OS
- Make efforts to try and develop an imaging process or extend a current imaging process if it seems inadequate or does not bring likely results
- Analysis of current tools used for imaging and analysis for Mac OS
- Identify areas of core evidential data in the Mac OS
- Identify overlap and differences of Linux OS with Mac OS

Literature Review

dd, dcfldd and dc3dd are command line utility that have been utilized for imaging drives in intel-based Macs. PALADIN which is a software from SUMURI is also used for imaging intel-based Macs [22]. The FTK Imager for Mac from ACCESSDATA which is a command line interface can also image some intel-based macs since it only supports Mac OS 10.5 and 10.6x [23]. Target Disk Mode is another feature in intel-based Macs where a Mac disk can be imaged [24].

asr [28] is the only command line utility at this point which can image from silicon-based Macs. A guide from Yogesh Khatri and Alexandra Cartwright sheds light on using asr to image a silicon-based Mac [29].

Software's such as RECON ITR [25], Cellebrite Digital Collector [26] and Magnet AXIOM [27] are one of the very few tools that have the capability to extract an image from any intel or silicon Mac and analyse it too.

APOLLO [30] and mac_apr [31] are two artifact parsing tools which can also be used for image analysis after a drive is successfully imaged.

Given that different tools need to be used under different conditions, SUMURI has a decision tree which provides with next steps to be taken when a specific requirement is set [32].

Research has been done on where important data is stored, two research papers by Rathod, Digvijaysinh focus on Safari artifacts [37] and iMessage, Facetime and Apple Mail [38] on intel-based Macs.

The Problem

Imaging RAM in macs with Silicon processors has been impossible till date since the RAM is protected by Kernel Extensions and Apple's notarization process [33].

dd, dcfldd and dc3dd are command line utilities that can be used for imaging intel-based macs but the imaging process using these command line utilities lacks proper instructions.

The FTK Imager for Mac from ACESSDATA has not been updated since 2012 and only works for Mac OS versions 10.5 and 10.6X.

PALADIN from SUMURI also only works for intel-based Macs.

It is not clear whether Target Disk Mode works for silicon-based Macs since the silicon-based Macs do not have Target Disk Mode but Disk Sharing Mode. It is unclear whether the Disk Sharing Mode in the newer Macs provide Disk Imaging just like Target Disk Mode does. Another disadvantage of Target Disk Mode and Disk Sharing Mode is that it requires two Macs (target device and acquisition device) to image a Mac.

asr which can successfully image from silicon-based Macs excludes snapshots [34] from the final disk image which means that snapshots have to be imaged as a different image.

The Cellebrite Digital Collector can only acquire a disk image of a silicon-based Mac in the AFF4 format [35]. As per BlackBagTech's (now acquired by Cellebrite) statement, the image file still displays the encryption flag even after acquiring the decrypted image [36]. It is unsure if this has been resolved or not.

Apart from this, the cost of software/hardware products such as Cellebrite Digital Collector, RECON ITR and Magnet AXIOM is another issue since that means not everyone can get to use these tools for data acquisition and analysis.

The tool APOLLO has not seen regular maintenance and development.

The artifact parsing tool mac_apr, although regularly maintained cannot analyse Time Machine Local Snapshots as of yet.

The research done, especially in regards with where valuable data is stored, or where to look for valuable data especially in a silicon-based Mac has been limited too.

From the information above it is made clear that with different hardware and software present, the imaging process changes and there are no clear guidelines on what should be done in every different scenario.

The Imaging Process

It is now clear that the most fundamental aspect on which the imaging process depends is the processor, the security chip and FileVault encryption.

From that we create a table of the possibilities we might be presented with when imaging a Mac.

Processor	T2 Security Chip	FileVault
Intel	Not Present	Disabled
Intel	Not Present	Enabled
Intel	Present	Disabled
Intel	Present	Enabled
Silicon	Present	Disabled
Silicon	Present	Enabled

We already know that the Silicon based Macs have the T2 Security Chip integrated in the chip.

How to turn off FileVault

1. From Apple Menu -> System Preferences -> Security & Privacy -> FileVault
2. Click the lock icon, if locked to unlock the preference pane
3. Enter administrator name and password
4. Turn off FileVault
5. Turn off Encryption

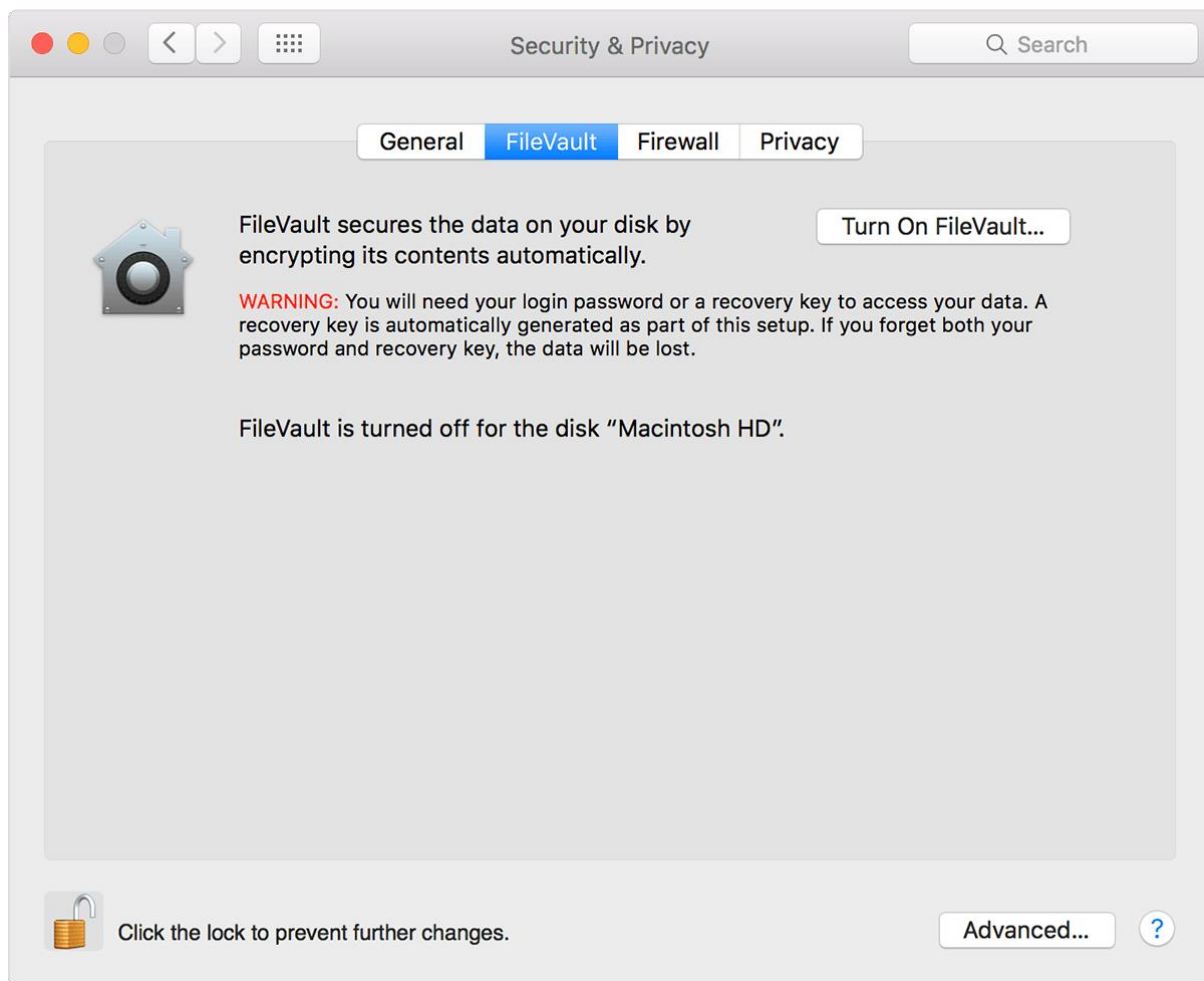
These commands are specific to Mac OS Monterey 12, commands for any other version can be found on macOS User Guide [39].

If possible, it is preferable to turn off Filevault if possible so that undue efforts are not made later on to decrypt the disk image file later on.

Since FileVault has no backdoor, user's login password or recovery key is required to decrypt the data.

If the system has the T2 security chip, it means that the data from the acquisition device must be decrypted before imaging, as it would not be possible to decrypt it later on since it is not just software encryption but hardware encryption on the data.

This means that the data in intel-based Macs with T2 and all silicon-based Macs must be decrypted before the imaging process starts.



What format should be the image file

These are the following formats in which the disk image file can be produced

1. AFF4 (The Advanced Forensics File Format), an open-source file format developed for digital evidence and data storage [40].
2. E01, where disk image storage and compression standards used in E01 format are used to encode the image file [41].
3. DD file extension, a compressed file archive which utilizes the LZ78 compression algorithm [42].
4. DMG (Apple Disk Images) file extension, which maintains disk image's integrity via checksums [43].

The best image file for an analysis depends on the capabilities of the tool used for analysis of the image file. For example, if the analysis tool only supports AFF4, then it is the best extension to create a disk image in.

However, if there is a choice and the analysis tool supports multiple disk image extensions, the DMG file extension is preferred since it is a file extension which is natively supported by the Mac OS.

Customer Data Migration Tool

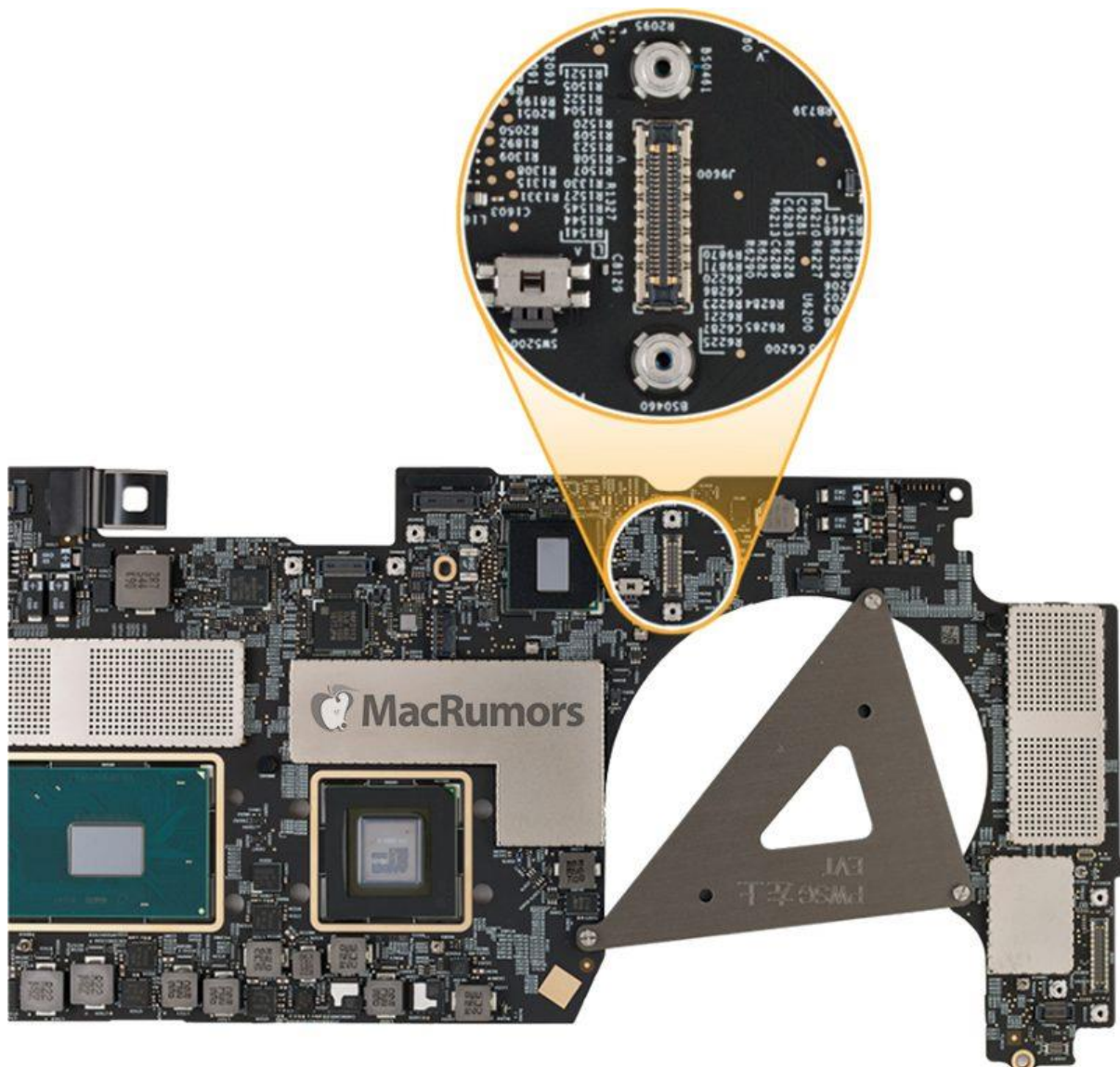
The 2016-2017 models of Macbook Pro (intel-based, non T2, FileVault enabled/disabled), the SSD was soldered to the logic board. These models came with a connector on the logic board which could connect to Customer Data Migration Tool Connector [44]. This tool would ultimately be connected to another Mac, which would then be able to image the drive using dd, dcfldd or dc3dd.

Acquisition device: device from which disk image is created

Target device: device where disk image is stored

Following are the steps to create an image using the Customer Data Migration Tool:

1. Open the back lid and remove the logic board from the Macbook.
2. Connect the logic board connector to the Customer Data Migration Tool Connector and the target device, which is the other Macbook.
3. Check if the acquisition drive is visible by running diskutil list.
4. Image the acquisition drive to target device using dd, dcfldd or dc3dd.



Customer Data Migration Tool connector

Source: MacRumors

The only drawback is that this process only works for Macs with the connector present in the logic boards which was 2016-2017 Macbook Pro models. The logic board connector was stripped away since the 2018 models. Also, the Customer Data Migration Tool connector is a costly toolkit [45], so financial constraints might be a reason to not pursue this imaging process.

Target Disk Mode

Target Disk Mode can be done on any intel based (T2 present/not present, FileVault turned off) Macs. It requires two Mac systems, a USB-C Cable and a thunderbolt cable if any of the systems has Mac OS Big Sur 11 or later installed [46].

1. Turn on the acquisition device and hold the letter T, the thunderbolt and USB symbol would pop up which means the device is in Target Disk Mode.
2. On the target device, we disable the disk arbitration utility so that the target device does not mount the acquisition device drive when connected.
3. Open terminal on target device and find the disk arbitration process by typing `sudo launchctl list | grep diskarbitrationd` and enter the password. This will provide us the number of the disk arbitration process
4. `sudo kill -SIGSTOP (enter the number of disk arbitration process)`
5. `ls /dev/disk*` to view the disks on the target device
6. Connect the acquisition device to the target device and run `ls /dev/disk*` again to view the drives of acquisition device
7. Select the acquisition drive to image and run `caffeinate dd if=/dev/(acquisition-drive) of=(target-drive-location)/imagefilename.dmg bs=1m status=progress`

Here we have used the caffeinate [47] command to stop our Target Device from going to sleep until the imaging process is complete.

The rest of the dd command is explained under the imaging using dd, dcfldd and dc3dd section below.

dd, dcfldd and dc3dd

The dd, dcfldd and the dc3dd command can be used to image all intel based (T2 present/not present, FileVault turned off) Macs.

I used a 64GB Dual Type-C USB 3.0 Flash Drive as the target device.

My acquisition device was an M1 Mac with FileVault enabled. However, the only target to achieve in this case was to image the drive using dd. I understood that the acquired image would be encrypted since it is an M1 Mac, but it would prove that if an encrypted image acquisition is possible on an M1 Mac using dd, decrypted image acquisition on intel-based Mac with FileVault disabled would be possible too.

Since my 500GB SSD of acquisition device only had 38GB of data and rest was free space, I decided to partition it so that I could only image the drive with data on it and not the free space, since my target device was only 64GB. Therefore, the size of the acquisition drive had to be less than that.

Steps on how to partition a physical disk in Mac can be found in the Disk Utility User Guide [48].

Firstly, I formatted my target drive to the APFS format [49]. This is because the acquisition drive was in APFS Format too and this was done to keep the environment filesystem same. This was done because I experienced that the disk image would sometimes get corrupted if the target filesystem was different than the acquisition filesystem. It is also done to make sure that sensitive data components such as Apple Extended Attributes [50] are conserved. Mac OS Extended (HFS+) [51] is another Filesystem natively supported by the Mac OS.

1. Power on the system and keep holding the power button until it says "Loading Startup Options".
2. Click Options -> Select User -> Enter Password -> Utilities -> Terminal.
3. Connect your target device and run `diskutil list`
4. Now you'll be able to see all the disks and notice the disk we shrunk `/dev/disk3` is the drive where all the data is stored. We'll also `/dev/disk8` which is our target drive.
5. The default command for dd is `dd if=input drive of=output drive`
I made some changes to the dd command to optimize it for imaging the mac drive
`dd if=input drive of=(output drive mount point)/filename.dmg bs=1m status=progress && conv=fdatasync`
6. To know the output drive mount point, we run the command `mount`
7. Finally, we unmount the input drive `/dev/disk3` using diskutil `unmountDisk /dev/disk3`
8. Now we run the command
`dd if=/dev/disk3 of=/Volumes/sandisk/ddimage.dmg bs=1m status=progress && conv=fdatasync`
9. Acquisition drive imaged successfully.


```
Terminal Shell Edit View Window Help

-bash-3.2# diskutil list
/dev/disk0 (internal):
#  TYPE NAME          SIZE      IDENTIFIER
0:  GUID_partition_scheme 500.3 GB  disk0
1:  Apple_APFS_ISC        524.3 MB  disk0s1
2:  Apple_APFS Container disk3 40.3 GB  disk0s2
3:  Apple_APFS Container disk4 454.1 GB  disk0s3
4:  Apple_APFS_Recovery    5.4 GB   disk0s4

/dev/disk3 (synthesized):
#  TYPE NAME          SIZE      IDENTIFIER
0:  APFS Container Scheme - +40.3 GB  disk3
   Physical Store disk0s2
1:  APFS Volume Macintosh HD - Data 22.9 GB  disk3s1
2:  APFS Volume Macintosh HD       15.4 GB  disk3s3
3:  APFS Volume Preboot             550.7 MB disk3s4
4:  APFS Volume Recovery            944.1 MB disk3s5
5:  APFS Volume VM                  20.5 KB  disk3s6

/dev/disk4 (synthesized):
#  TYPE NAME          SIZE      IDENTIFIER
0:  APFS Container Scheme - +454.1 GB  disk4
   Physical Store disk0s3
1:  APFS Volume NEWAPFS           40.3 GB  disk4s1

/dev/disk5 (external, physical):
#  TYPE NAME          SIZE      IDENTIFIER
0:  GUID_partition_scheme *61.5 GB  disk5
1:  EFI EFI            209.7 MB  disk5s1
2:  Apple_APFS Container disk6 61.3 GB  disk5s2

/dev/disk6 (synthesized):
#  TYPE NAME          SIZE      IDENTIFIER
0:  APFS Container Scheme - +61.3 GB  disk6
   Physical Store disk5s2
1:  APFS Volume sandisk          24.6 KB  disk6s1

/dev/disk7 (disk image):
#  TYPE NAME          SIZE      IDENTIFIER
0:  GUID_partition_scheme +1.9 GB  disk7
1:  Apple_APFS Container disk8 1.9 GB  disk7s1

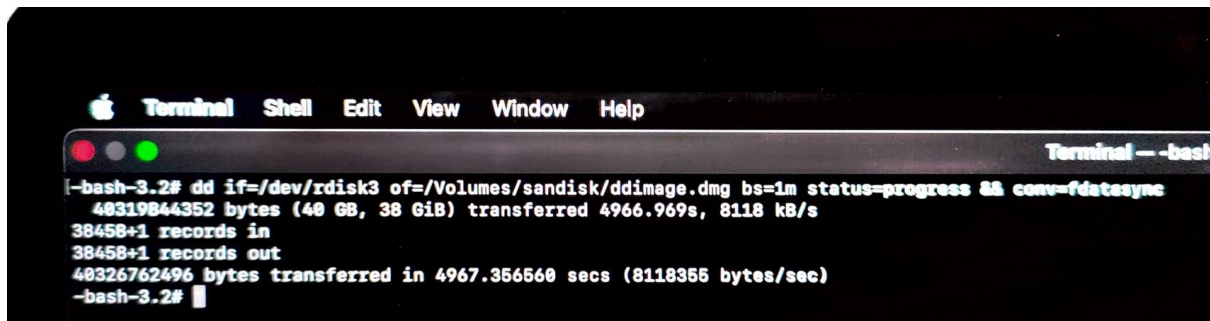
/dev/disk8 (synthesized):
#  TYPE NAME          SIZE      IDENTIFIER
0:  APFS Container Scheme - +1.9 GB  disk8
   Physical Store disk7s1
1:  APFS Volume macOS Base System 1.8 GB  disk8s1

-bash-3.2#
```

Output after running diskutil list

```
-bash-3.2# mount
/dev/disk8s1 on / (apfs, sealed, local, read-only, journaled)
devfs on /dev (devfs, local, nobrowse)
/dev/disk3s2 on /System/Volumes/Update (apfs, local, journaled, nobrowse)
/dev/disk1s2 on /System/Volumes/xarts (apfs, local, noexec, journaled, noatime, nobrowse)
/dev/disk1s1 on /System/Volumes/iSCPreboot (apfs, local, journaled, nobrowse)
/dev/disk1s3 on /System/Volumes/Hardware (apfs, local, journaled, nobrowse)
tmpfs on /System/Volumes/Data (tmpfs, local)
tmpfs on /Volumes (tmpfs, local)
/dev/disk3s5 on /System/Volumes/Data/private/tmp/Recovery (apfs, local, journaled, nobrowse)
tmpfs on /System/Volumes/Preboot (tmpfs, local)
/dev/disk4s1 on /Volumes/NEWAPFS (apfs, local, journaled, nobrowse, protect)
/dev/disk3s3 on /Volumes/Macintosh HD (apfs, sealed, local, read-only, journaled, nobrowse)
/dev/disk6s1 on /Volumes/sandisk (apfs, local, nodev, nosuid, journaled, noowners)
-bash-3.2#
```

Output after mount command

A screenshot of a macOS Terminal window. The title bar shows 'Terminal' and standard window controls. The menu bar includes 'Terminal', 'Shell', 'Edit', 'View', 'Window', and 'Help'. The terminal text shows a successful execution of the 'dd' command to create a disk image. The output includes the command, the number of bytes transferred (40319844352), the time taken (4966.969s), and the transfer rate (8118 kB/s).

```
-bash-3.2# dd if=/dev/rdisk3 of=/Volumes/sandisk/ddimage.dmg bs=1m status=progress && conv=fdatasync
40319844352 bytes (40 GB, 38 GiB) transferred 4966.969s, 8118 kB/s
38458+1 records in
38458+1 records out
40326762496 bytes transferred in 4967.356560 secs (8118355 bytes/sec)
-bash-3.2#
```

Output after dd command is successfully executed

I chose /dev/rdisk3 instead of /dev/disk3 because /dev/disk# is buffered device whereas /dev/rdisk# is raw path which is around 20 times faster on class 4 SD card [52].

Here bs is the block size. I chose bs=1m since it is not too big of a block size but not so small as the default block size 512 bytes.

Status=progress is used to display the status of progress.

conv=fdatasync guarantees file data will be written before dd exits [53].

To use dcfldd instead, in step 8 execute command

```
dc3dd if=/dev/rdisk3 hash=sha256 hashlog=hash.log of=/Volumes/sandisk/ddimage.dmg
bs=1m status=progress && conv=fdatasync
```

To use dc3dd instead, in step 8 execute command

```
dc3ddd if=/dev/rdisk3 of=/Volumes/sandisk/ddimage.dmg bs=1m hash=sha256
hashlog=hash.log log=image.log status=progress && conv=fdatasync
```

Here, hash=sha256 is on the fly hashing algorithm

hashlog=hash.log saves hash output to hash.log

log=image.log is log output path

The advantage dc3dd and dcfldd have over dd is that they provide on the fly hashing among many other features. Dcfldd has more hashing algorithms than dc3dd. However, dcfldd is a fork of dd and dc3dd is a patch. This means that dc3dd is updated every time dd is updated but dcfldd is running on an old version of dd [54]. This makes dcfldd dangerous as it can either misalign the data or fall into an infinite loop if a faulty sector is encountered [55].

For the reasons above, it is better to use dc3dd if on the fly hashing is needed or just use dd if there is no such requirement.

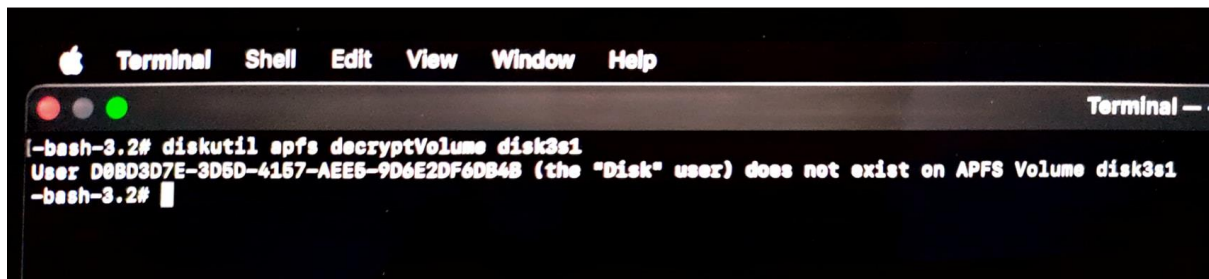
Imaging the M1 Mac

My first attempt at imaging the drive of an M1 Mac was using dd. I was researching ways the drive could be decrypted before I started imaging it with dd. If I were able to decrypt the

drive before running the dd command, I would successfully be able to image the entire drive and not an APFS partition like asr does.

Therefore, after step 7 of the dd section I tried the following commands

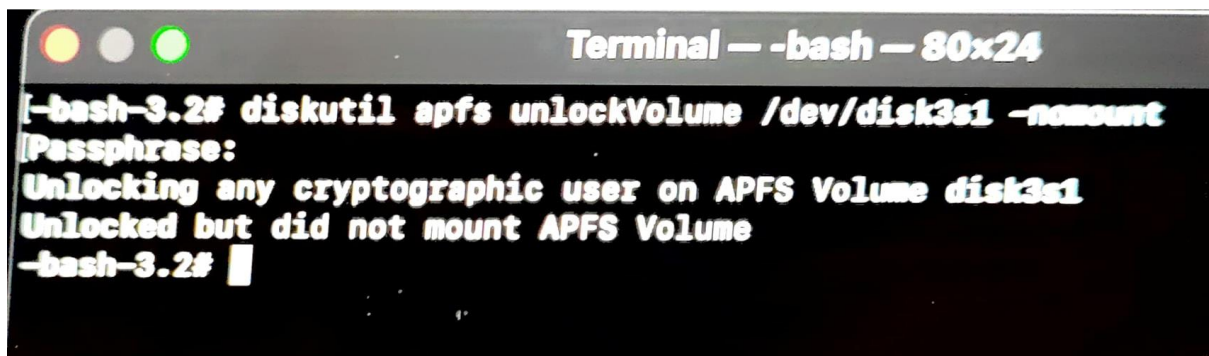
```
diskutil apfs decryptVolume disk3s1
```



Unfortunately, this did not work. I somehow tried to get an image file but it was still encrypted. I am speculated the reason was lack of privileges, since it reported that user did not exist on that APFS Volume.

Strangely enough though, I was able to unlock the volume using the command

```
diskutil apfs unlockVolume disk3s1 -nomount
```



But this was not enough as the primary objective was to be able to decrypt the APFS volume.

To try and decrypt it on a VMWare was an option but the results would not have been honest since a VMWare could not emulate the T2 chip which handles the hardware encryption/decryption in an M1 Mac.

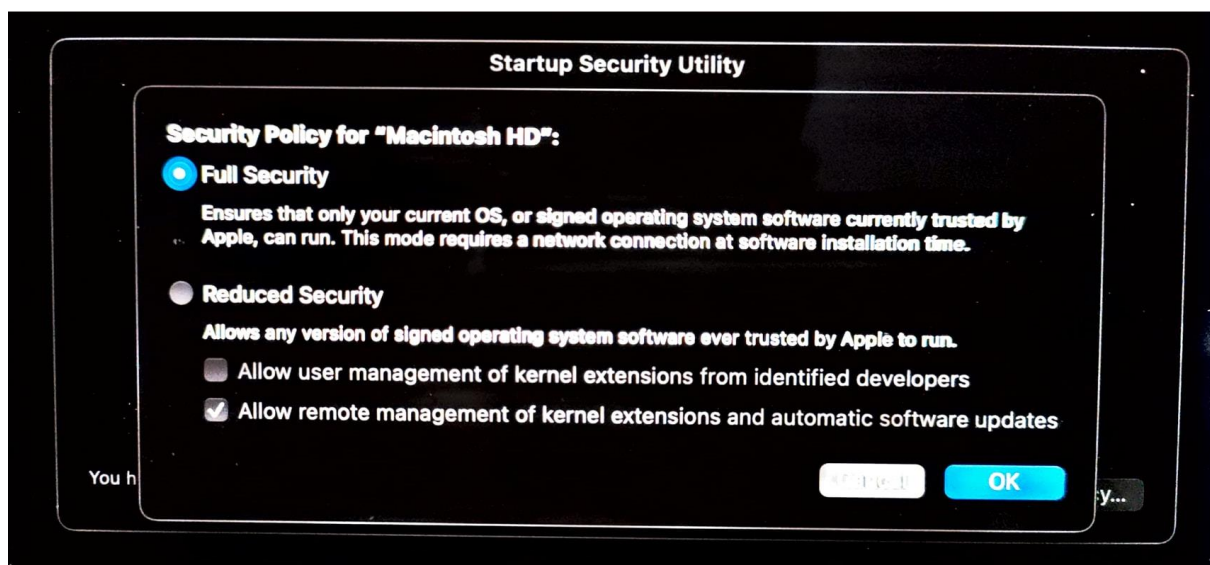
Therefore, I used the asr command line utility to image the M1 Mac.

Following are the steps to image an M1 Mac using asr.

1. Create a bootable installer for Mac OS [56], in this case in the 64GB flash drive.
2. The second step was to change settings in Startup Security Utility, to allow booting from external media. However, the features of Startup Security Utility, such as Secure Boot and External Boot was replaced with Full Security and Reduced Security.



Source: support.apple.com [58]



No option to external boot available

I did try rebooting through recovery but somehow it always ended up failing. A possible reason of that could be that LocalPolicy creation issues [57].

This means that the only way to image an M1 Mac at the point are the commercial software's such as Recon ITR, Cellebrite Digital Collector and Magnet Axiom which all have a price.

Areas of Core Evidential Data in Mac

Before delving deep into all the core evidential data that is stored in a Mac, my first approach was to look at all the applications and processes clearly visible in the Mac OS.

Here is a look at some of those applications:

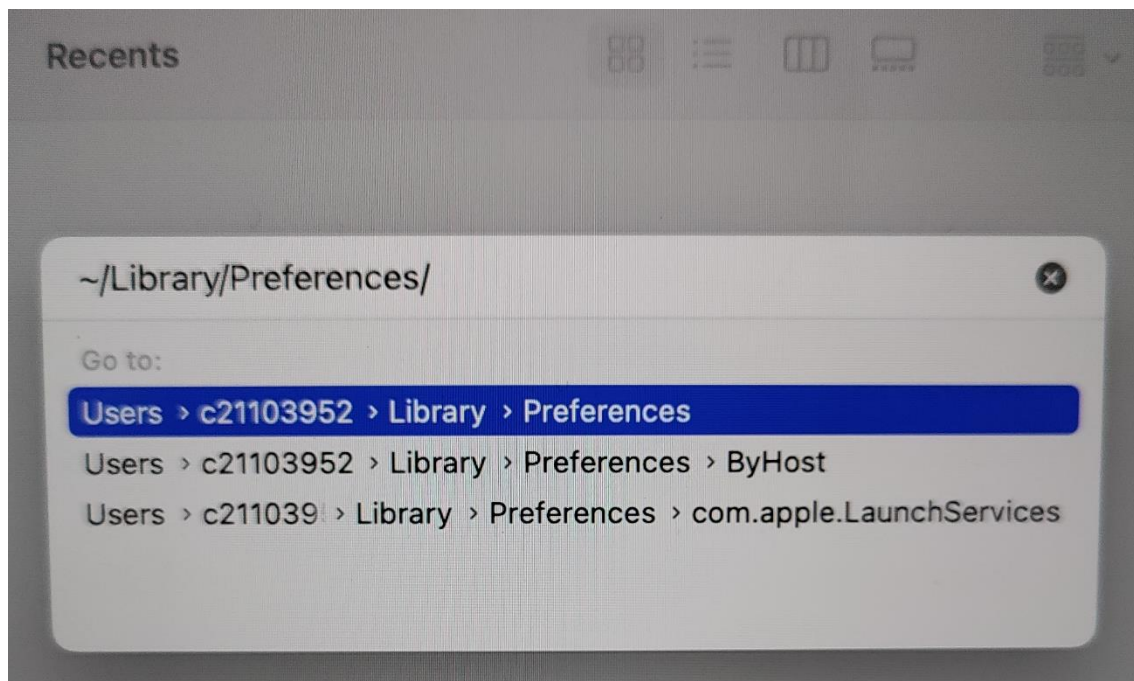
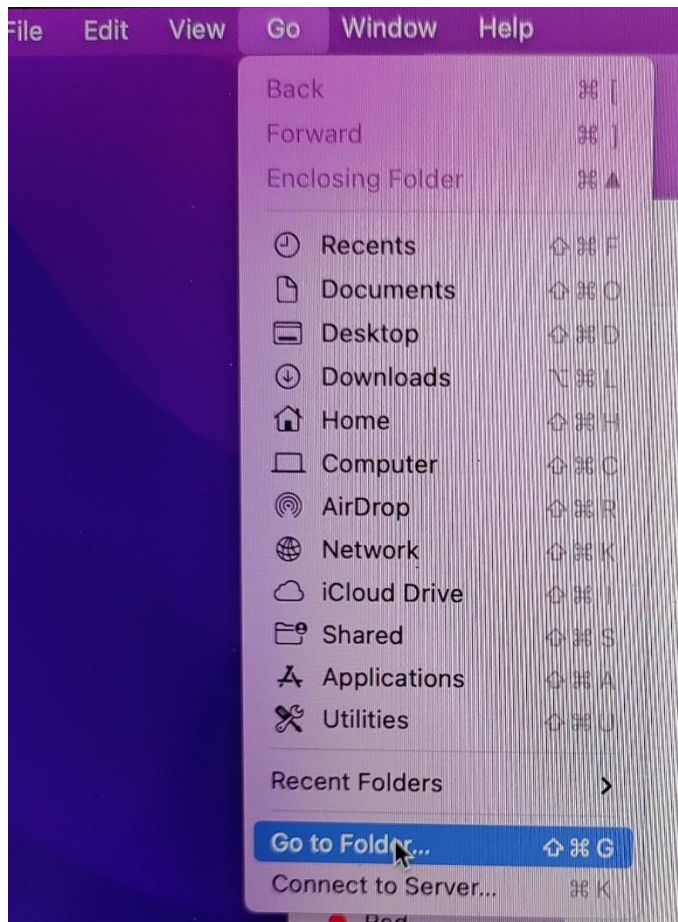
• Bluetooth	• Chrome
• iMessage	• MS Office
• Notes	• Safari
• Spotlight	• Wi-Fi
• Login	• Recent Items
• Install History	• Downloaded Files
• Mount/Unmount Log	• About this Mac
• Terminal	• Users
• Recent Files	• Keychain

A lot of relevant information about these services and applications are stored in plist files [59]. Plist files store application preferences, saving data, file log, etc.

Plist Files in Mac are equivalent to Windows Registry Files [60] in Windows OS.

One way to view most of the plist files is:

1. Open Finder
2. Select Go -> Go to Folder
3. Enter ~Library/Preferences
4. These files can be opened through Xcode, PLIST Editor or even TextEdit. However, it might not be very readable in TextEdit.
5. To convert plist files from binary to xml, type in terminal
`plutil -convert xml1 filename.plist`
6. And to convert from xml to binary
`plutil -convert binary1 filename.plist`, where filename is the name of the plist file in both cases.



However, the mac_apt parsing tool, with more than 40 plugins automatically parses plist files from a disk image and supports popular image formats such as DD, DMG, E01, VMDK, AFF4, SPARSEIMAGE and mounted images among a few more.

mac_apt can be installed on any of Mac, Windows or Linux Systems.

mac_apr installation guide can be found on their GitHub [61].

Below is a table of applications and processes and their plist file locations.

App/Processes	plist File Location
Bluetooth	/Library/Preferences/com.apple.Bluetooth.plist
Install History	/Library/Receipts/InstallHistory.plist
Notes	/Users/[user]/Library/Group Containers/group.com.apple.notes/NoteStore.sqlite
Recent Items	/Users/{USER}/Library/Preferences/com.apple.sidebarlists.plist
Safari	/Users/{USER}/Library/Preferences/com.apple.safari.plist
WiFi	/Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist.
Users	/private/var/db/dslocal/nodes/Default/users/username.plist
System Updates	/Library/Preferences/com.apple.SoftwareUpdate.plist
Keychain	/Library/Keychains/System.keychain

Swapfile is another really important artifact found in */private/var/vm*. Swapfile is a temporary storage on a secondary storage device where RAM data is stored when the RAM runs low on memory.

Bootrom, stored in flash memory chips is also important since it provides memory verification, hardware initialization and boot partition selection.

Log Files in *private/var/log* where important log files such as system log, authentication log, firewall log and security log among other important log information is stored is another core evidence data in Mac OS.

Time Machine Local Snapshots [62] is another area of core evidence in Macs. However, there is no open-source software at the moment that does Snapshot analysis.

Overlap and Differences with Linux Distributions

Linux and Mac OS are both great operating systems which have the same heritage which is Unix. However different their user interface looks, with Linux having more freedom to change their user interface to any liking, their base directory structure is unsurprisingly similar.

Below is a comparison of base structure of Mac OS and Linux

Mac OS	Linux
/bin	/bin
/Library	/lib
/System	/sys
/usr	/usr
/	/
/etc	/etc
/dev	/dev
/sbin	/sbin
/tmp	/tmp
/var	/var

The swapfile is another similarity between Mac OS and Linux. However, Linux may have a swap partition instead of just having a swapfile.

Bootrom in Mac is another file which is similar to the data in EFI partition in Linux [63], which does the same job as Bootrom of setting up the system to boot and setting up the boot file.

Differences

Despite being so similar in base architecture, Mac OS and Linux are not that similar.

The biggest difference between them being that Mac OS is proprietary while Linux is open-source.

Another thing to note is how almost all plist files in Mac are stored in a specific location and the location changes with the nature of the plist file and not the application itself. This means log files in one place, plist files in one place. In Linux however, the location of the config files stored is specific to the application. On top of that, most of the times, the config files are human readable [64].

Analysis

When it comes to analysis of the current imaging processes in Mac OS, I think the project was a success in terms of defining and explaining the factors that make imaging a Mac such a complicated task. The current imaging process were critically analysed and the lack of instructions which were an issue and the small factors which needed to be taken into consideration when imaging a Mac were taken care of. For example, caffeinate during imaging using the Target Disk Mode is a very small factor but is crucial since if the target system sleeps during the imaging process, it might break down the process or might even end up corrupting the acquisition drive. The fact that every generation of Mac with every major hardware change was taken into consideration and the focus was not only kept on the latest generation of Mac or just one type of imaging process is also a success. Imagine if a system cannot turn on so you cannot use dd for imaging but if applicable, the customer data migration tool connector can be used.

The research project hit a setback when the imaging of an M1 Mac was unsuccessful. I do believe though that if I had full administrative rights, I would have been able to find a way to decrypt it after turning off FileVault, since that would have made things easier. If I had access to another Mac, I could've given a try to Share Disk Mode to try and image the drive similar to how it was done in target Disk Mode. These two options would've provided me two more shots at an attempt to decrypt and image an M1 Mac.

Despite the setback, I still believe the project fills much of the lack of instructions when it comes to imaging a Mac.

When it came to analysis of the current tools, it was another setback to see that external boot option has been removed now and LocalPolicy is required.

When it came to analysis after imaging, the scarcity of analysis tools did not surprise me since I already knew much of the tools are paid and the open source are not on that level of development yet. It was because of this though that my identification of areas of core evidential data in Macs suffered a bit since I did not have a proper analysis tool to analyse the several disk image files I had acquired with during the course of my project. Still, with only mac apt and some raw data lookup, I was able to identify plist files, important partitions and log files which are crucial data in case of an analysis/investigation. It is important to note that Mac OS generally has around 20000 plist files at the moment so a lot of thought needs to be put to determine which plist file should be considered as core evident data and which should be ranked lower.

The overlap and differences in Linux were something I believe I could have done better had I taken the approach of trying to image and analysing the Linux OS as well.

Conclusions

The project fulfils the goal of providing guidelines to the imaging process. The project also, with partial success is able to identify core evidential data in Mac OS, although not to the extent it could have if there were no financial constraints or there was access to the few analysis tools that dominate the Mac Forensics market.

I also believe that using a Flash Drive was not a smart idea and using an SSD would've been a better decision since the APFS Volumes and Mac OS is optimized for SSD. With a 1TB SSD the entire drive could've been imaged and there wouldn't have been any need to shrink the volumes. However, the cost of 1TB SSD was another factor, given a flash drive only costed me 1/10th of the price of an SSD.

The biggest deficiency of the project remains the lack of testing of Share Disk Mode on M1 Macs and efforts to decrypt and image after turning off FileVault. Even if these two options didn't work, we'd have known that at least there's two more options that don't work on M1 Macs.

The second deficiency, and I feel like I am reiterating here is the lack of the commercial analysis tools.

My recommendation on the basis of the research I have done is that dd remains the most reliable method of imaging a Mac and the fact that it can image in the dmg format is another big point. The analysis part still remains a bit raw but I would say that plist files seem to be the most important artifact in Mac Image Analysis.

I would also say that preparation is needed since the use of Macs have increased on corporate/industrial and educational level, hackers have also turned their attention towards the growing Mac OS ecosystem. It only makes sense that the number of Macs on which evidence acquisition and analysis needs to be performed is only going to grow.

Learning

One of the best research skills I have learned from this dissertation project is that how deeply you are willing to research on a project is directly proportional to how passionate you are about the project. It takes time and patience to look for answers because if you get irritated with the process, you won't see the answer even if it is right in front of your eyes. And when you really want an answer, you'll understand it was your own mental gymnastics that made it tough to get to the answer.

The lessons I learnt especially understanding the mac OS at a Filesystem level are amazing. The fact that I chose to dive into almost all the available imaging processes and not stick with just one process was a really good decision. However, I surprised myself when I let go and didn't make any more attempts at trying to decrypt the M1 Mac. I made a rational decision and decided to focus further on the project since I realized I was spending too much time on decryption which was hurting my overall project. This decision surprised because I had a habit of not letting go of things until I could fix them. This decision proved that sense prevails after all. End of September I had to looking for a new place and the time management I put in place with the 2-minute rule was what led to my dissertation project to be finished on time. The 2-minute rule states that if it can be done in two minutes, do it now and that saved me a lot from procrastination.

I do understand with this that I need better skills when it comes to identification of core evidential data or the evidence analysis aspect of data forensics. Another flaw that I need to work upon is that once I start working, I find it very difficult to stop since I fear it would break my flow which leads to me working for hours without any break. I need to learn to take short breaks without fearing that it would break my work rhythm.

Bibliography

1. Web.stanford.edu. 2022. *First Macintosh Press Release*. [online] Available at: <<https://web.stanford.edu/dept/SUL/sites/mac/primary/docs/pr1.html>> [Accessed 5 July 2022].
2. GitHub. 2022. *GitHub - apple-oss-distributions/xnu*. [online] Available at: <<https://github.com/apple-oss-distributions/xnu>> [Accessed 5 July 2022].
3. GitHub. 2022. *GitHub - apple/darwin-xnu: The Darwin Kernel (mirror). This repository is a pure mirror and contributions are currently not accepted via pull-requests, please submit your contributions via https://developer.apple.com/bug-reporting/*. [online] Available at: <<https://github.com/apple/darwin-xnu>> [Accessed 5 July 2022].
4. O'Reilly Online Learning. 2022. *Running Mac OS X Tiger*. [online] Available at: <<https://www.oreilly.com/library/view/running-mac-os/0596009135/ch01s04.html>> [Accessed 5 July 2022].
5. Bsd.org. 2022. *www.bsd.org*. [online] Available at: <<https://www.bsd.org/>> [Accessed 5 July 2022].
6. Developer.apple.com. 2022. *Mach Overview*. [online] Available at: <<https://developer.apple.com/library/archive/documentation/Darwin/Conceptual/KernelProgramming/Mach/Mach.html>> [Accessed 5 July 2022].
7. Joyce, R., Powers, J. and Adelstein, F., 2008. MEGA: A tool for Mac OS X operating system and application forensics. *Digital Investigation*, [online] 5, pp.S83-S90. Available at: <<https://www.sciencedirect.com/science/article/pii/S1742287608000376>> [Accessed 5 July 2022].
8. Apple Support. 2022. *File system formats available in Disk Utility on Mac*. [online] Available at: <<https://support.apple.com/en-gb/guide/disk-utility/dsku19ed921c/mac>> [Accessed 15 July 2022].
9. Developer.apple.com. 2022. *Technical Note TN1150: HFS Plus Volume Format*. [online] Available at: <<https://developer.apple.com/library/archive/technotes/tn/tn1150.html>> [Accessed 15 July 2022].
10. MiniTool. 2022. *APFS vs Mac OS Extended – Which Is Better & How To Format*. [online] Available at: <<https://www.minitool.com/data-recovery/apfs-vs-mac-os-extended-difference-and-format.html>> [Accessed 15 July 2022].
11. MiniTool. 2022. *What Is GPT or GUID Partition Table (Complete Guide)*. [online] Available at: <<https://www.minitool.com/lib/gpt.html>> [Accessed 15 July 2022].

12. Medium. 2022. *Get around your Mac Disks*. [online] Available at: <<https://medium.com/macoclock/get-around-your-mac-disks-892db0a671af>> [Accessed 15 July 2022].
13. Apple Support. 2022. *Use FileVault to encrypt the startup disk on your Mac*. [online] Available at: <<https://support.apple.com/en-us/HT204837>> [Accessed 16 July 2022].
14. Xilinx.github.io. 2022. [online] Available at: <https://xilinx.github.io/Vitis_Libraries/security/2019.2/guide_L1/internals/xts.html> [Accessed 16 July 2022].
15. Dworkin, M., 2010. *Recommendation for Block Cipher Modes of Operation: The XTS-AES Mode for Confidentiality on Storage Devices*. [online] NIST. Available at: <<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38e.pdf>> [Accessed 16 July 2022].
16. Apple Support. 2022. *Mac models with the Apple T2 Security Chip*. [online] Available at: <<https://support.apple.com/en-gb/HT208862>> [Accessed 17 July 2022].
17. Apple Support. 2022. *About encrypted storage on your new Mac*. [online] Available at: <<https://support.apple.com/en-gb/HT208344>> [Accessed 17 July 2022].
18. Apple Support. 2022. *Mac computers with Apple silicon*. [online] Available at: <<https://support.apple.com/en-gb/HT211814>> [Accessed 17 July 2022].
19. Clover, J., 2022. *Apple M1 Chip: Everything You Need to Know*. [online] MacRumors. Available at: <<https://www.macrumors.com/guide/m1/>> [Accessed 17 July 2022].
20. Xda-developers.com. 2022. [online] Available at: <<https://www.xda-developers.com/apple-silicon-unified-memory/>> [Accessed 18 July 2022].
21. Developer.apple.com. 2022. *About Disk Arbitration*. [online] Available at: <<https://developer.apple.com/library/archive/documentation/DriversKernelHardware/Conceptual/DiskArbitrationProgGuide/Introduction/Introduction.html>> [Accessed 19 July 2022].
22. *Paladin edge (64-bit) (2021) SUMURI*. Available at: <https://sumuri.com/product/paladin-edge-64-bit/> (Accessed: July 18, 2022).
23. *Mac OS 10.5 and 10.6X version – 3.1.1* (no date) *AccessData*. Available at: <https://accessdata.com/product-download/mac-os-10-5-and-10-6x-version-3-1-1> (Accessed: July 18, 2022).
24. *Transfer files between two Mac computers using Target Disk Mode* (no date) *Apple Support*. Available at: <https://support.apple.com/en-gb/guide/mac-help/mchlp1443/mac> (Accessed: July 18, 2022).

25. *Recon ltr* (2022) SUMURI. Available at: <https://sumuri.com/product/recon-ltr/> (Accessed: July 21, 2022).
26. *Digital Collector* (2022) Cellebrite. Available at: <https://cellebrite.com/en/digital-collector/> (Accessed: July 21, 2022).
27. *Magnet axiom: Digital forensic software* (2022) Magnet Forensics. Available at: <https://www.magnetforensics.com/products/magnet-axiom/> (Accessed: July 21, 2022).
28. *Mac OS X in a Nutshell* (no date) O'Reilly Online Learning. O'Reilly Media, Inc. Available at: <https://www.oreilly.com/library/view/mac-os-x/0596003706/re146.html> (Accessed: July 21, 2022).
29. *Presentations/macOS forensics-mus2020.pdf at master · Ydkhatri ...* (no date). Available at: <https://github.com/ydkhatri/Presentations/blob/master/macOS%20Forensics-MUS2020.pdf> (Accessed: July 21, 2022).
30. *mac4n6* (no date) *Mac4n6/apollo: Apple Pattern of Life lazy output'er, GitHub*. Available at: <https://github.com/mac4n6/APOLLO> (Accessed: July 21, 2022).
31. Ydkhatri (no date) *Ydkhatri/mac_apt: MacOS (& IOS) artifact parsing tool, GitHub*. Available at: https://github.com/ydkhatri/mac_apt (Accessed: July 21, 2022).
32. *MAC Imaging Guide: Sumuri guide* (2022) SUMURI. Available at: <https://sumuri.com/mac-imaging-guide/> (Accessed: July 21, 2022).
33. *Notarizing macOS software before distribution* (no date) *Apple Developer Documentation*. Available at: https://developer.apple.com/documentation/security/notarizing_macos_software_before_distribution (Accessed: August 1, 2022).
34. *About Time Machine local snapshots on Mac* (no date) *Apple Support*. Available at: <https://support.apple.com/en-gb/guide/mac-help/mh35933/mac#:~:text=Time%20Machine%20makes%20copies%2C%20each,is%20needed%20on%20the%20disk>. (Accessed: August 10, 2022).
35. Justin Matsuhara - Team Lead, S.E.at C.and S.T.- S.E.at C. (2021) *Cellebrite Digital Collector - taking away the guess work, Cellebrite*. Available at: <https://cellebrite.com/en/cellebrite-digital-collector-taking-away-the-guess-work/> (Accessed: August 10, 2022).
36. Sylve, J. (2020) *Encryption Flag in hardware encrypted APFS containers, Twitter*. Twitter. Available at: https://twitter.com/jtsylve/status/1255168538714746882?s=20&t=A8B_8N_9EbcAhjX5B9ywow (Accessed: August 10, 2022).
37. Rathod, Digvijaysinh. (2017). MAC OSX Forensics. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 6. 1240. (Accessed: August 15, 2022).
38. Rathod, Digvijaysinh. (2017). MAC OSX: iMessage, Face Time, Apple Mail Application Forensics. JOURNAL OF INFORMATION, KNOWLEDGE AND

RESEARCH IN COMPUTER ENGINEERING. 4. 1000. (Accessed: August 15, 2022).

39. *Turn off filevault encryption on Mac* (no date) *Apple Support*. Available at: <https://support.apple.com/en-gb/guide/mac-help/mchlp2560/mac> (Accessed: August 15, 2022).
40. *The Advanced Forensics File Format* (no date) *AFF4*. Available at: <http://www2.aff4.org/> (Accessed: August 21, 2022).
41. *ReviverSoft* (no date) *File extension search, ReviverSoft*. Available at: <https://www.reviversoft.com/en/file-extensions/e01> (Accessed: August 21, 2022).
42. *.DD file extension* (2018) *DD File Extension - What is a .dd file and how do I open it?* Available at: <https://fileinfo.com/extension/dd> (Accessed: August 21, 2022).
43. *.DMG file extension* (2022) *DMG File Extension - What is a .dmg file and how do I open it?* Available at: <https://fileinfo.com/extension/dmg> (Accessed: August 21, 2022).
44. Subramaniam, V. (2018) *Apple removes the customer data migration tool connector in the 2018 Macbook pro with touch bar, Notebookcheck*. Notebookcheck. Available at: <https://www.notebookcheck.net/Apple-removes-the-Customer-Data-Migration-Tool-connector-in-the-the-2018-MacBook-Pro-with-Touch-Bar.318186.0.html> (Accessed: August 25, 2022).
45. *New Apple Customer Data Migration Tool Kit 076-00236 for MacBook Pro 2016-2017* (1970) *eBay*. Available at: <https://www.ebay.co.uk/itm/282989664130> (Accessed: August 25, 2022).
46. *Transfer files between two Mac computers using Target Disk Mode* (no date) *Apple Support*. Available at: <https://support.apple.com/en-gb/guide/mac-help/mchlp1443/mac> (Accessed: August 30, 2022).
47. Pickerill, C. (2022) *Caffeinate your mac, The Apple Geek*. The Apple Geek. Available at: <https://www.theapplegeek.co.uk/blog/caffeinate> (Accessed: August 30, 2022).
48. *Partition a physical disk in Disk utility on mac* (no date) *Apple Support*. Available at: <https://support.apple.com/en-gb/guide/disk-utility/dskutl14027/mac> (Accessed: September 1, 2022).
49. *How to format your drive APFS on macos 11 (big sur) and later: Seagate Support Us* (no date) *Seagate.com*. Available at: <https://www.seagate.com/gb/en/support/kb/how-to-format-your-drive-apfs-on-macos-big-sur-and-later/> (Accessed: September 1, 2022).
50. *ExtendedAttributes*. (no date) *Apple Developer Documentation*. Available at: <https://developer.apple.com/documentation/fileprovider/nsfileprovideritem/3074511-extendedattributes> (Accessed: September 1, 2022).

51. *File system formats available in Disk Utility on mac* (no date) Apple Support. Available at: <https://support.apple.com/lt-it/guide/disk-utility/dsku19ed921c/mac> (Accessed: September 1, 2022).
52. *Rpi Easy SD card setup* (no date) RPi Easy SD Card Setup - eLinux.org. Available at: https://elinux.org/RPi_Easy_SD_Card_Setup#Flashing_the_SD_card_using_Mac_OS_X (Accessed: September 1, 2022).
53. *Informit* (no date) InformIT. Available at: <https://www.informit.com/articles/article.aspx?p=23618&seqNum=5> (Accessed: September 1, 2022).
54. *DC3DD* (no date) Dc3dd - Forensics Wiki. Available at: <https://forensicswiki.xyz/wiki/index.php?title=Dc3dd> (Accessed: September 1, 2022).
55. *DCFLDD* (no date) Dcfldd - Forensics Wiki. Available at: <https://forensicswiki.xyz/wiki/index.php?title=Dcfldd> (Accessed: September 1, 2022).
56. *How to create a bootable installer for macos* (2022) Apple Support. Available at: <https://support.apple.com/en-gb/HT201372> (Accessed: September 5, 2022).
57. Oakley, H. (2021) *Owners and users: Primary and secondary systems on M1 Macs*, The Eclectic Light Company. Available at: <https://eclecticlight.co/2021/07/21/owners-and-users-primary-and-secondary-systems-on-m1-macs/> (Accessed: September 10, 2022).
58. *About startup security utility on a Mac with the Apple T2 Security Chip* (2021) Apple Support. Available at: <https://support.apple.com/en-gb/HT208198> (Accessed: September 10, 2022).
59. *.Plist file extension* (2019) PLIST File Extension - What is a .plist file and how do I open it? Available at: <https://fileinfo.com/extension/plist> (Accessed: September 21, 2022).
60. Freda, A. (2021) *What is the windows registry and how does it work?* Avast. Available at: <https://www.avast.com/c-windows-registry> (Accessed: September 21, 2022).
61. Ydkhatri (no date) *Installation for python3* · Ydkhatri/mac_apt wiki, GitHub. Available at: https://github.com/ydkhatri/mac_apt/wiki/Installation-for-Python3 (Accessed: October 21, 2022).
62. *About Time Machine local snapshots on Mac* (no date) Apple Support. Available at: <https://support.apple.com/en-gb/guide/mac-help/mh35933/mac> (Accessed: September 30, 2022).
63. *EFI System Partition* (no date) EFI system partition - ArchWiki. Available at: https://wiki.archlinux.org/title/EFI_system_partition (Accessed: October 1, 2022).

64. *Linux equivalent of windows registry* (1958) *Super User*. Available at: <https://superuser.com/questions/295635/linux-equivalent-of-windows-registry> (Accessed: October 1, 2022).