Computer and Internet Forensics
COSC2301-COSC3135 (Semester 2, 2025)
Assignment 1
S4100564

Overview

The objective of Assignment 1 is to evaluate your knowledge of the topics covered in Lectures, Tutorials, and Workshops from 2 to 5. Topics include Static data acquisition from the suspect's computer and forensics of static and live data artifacts. Assignment 1 will focus on developing your abilities in identifying the basic digital forensic tasks. Assignment 1 contains problems related to the topics mentioned above. You are required to prepare the solutions as a single PDF or MS Word file with the description of the step-by-step processes, with screenshots whenever required. Develop the solution to this assignment in an iterative fashion (as opposed to completing it in one sitting). By completing each week's tasks starting from Week 2 to Week 5, you should be able to solve at least one question in this assignment. This is an individual assignment.

Learning Outcomes

This assessment is relevant to the following course learning outcomes:

- **CLO 1**: Understand the principles and practices of computer and internet forensics, including the methods used to investigate and analyze digital evidence.

- **CLO 2**: Identify and apply appropriate forensic tools and techniques to recover, preserve, and examine data from various digital devices.

- **CLO 3**: Analyze digital evidence to reconstruct events, identify perpetrators, and understand the context of cyber incidents.

- **CLO 4**: Evaluate the legal and ethical considerations involved in digital forensic investigations, ensuring compliance with relevant laws and regulations.

- **CLO 5**: Communicate forensic findings effectively through clear, concise, and professional reports and presentations, suitable for both technical and non-technical audiences.

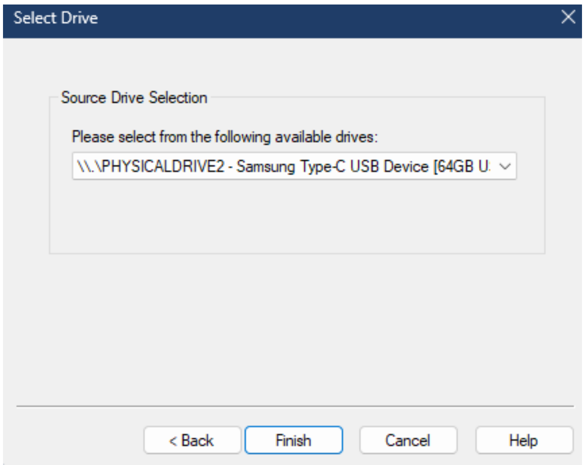
- **CLO 6**: Educate stakeholders about threats, possible mitigation approaches, and report actions taken in response to incidents

PART-1: DATA ACQUISITION (12 Marks)

Q1. Static Data Acquisition from a USB Disk (3 Marks)

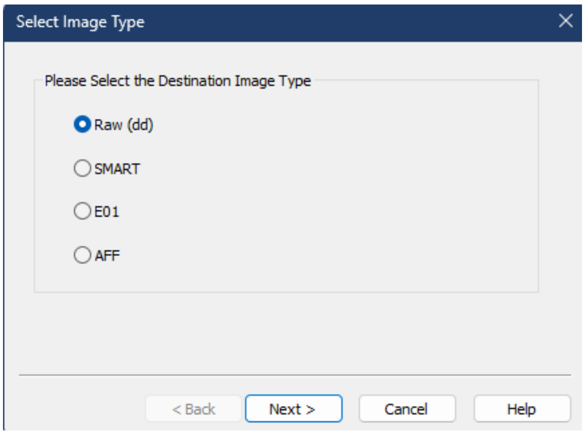
a) To create a disk image in Raw format, we will use FTK Imager as follows:

i. We will start by navigating to File → Create Disk Image in FTK Imager. When prompted, we'll select our usb drive as shown below:



Please select from the following available drives:
\\.\PHYSICALDRIVE2 - Samsung Type-C USB Device [64GB U

ii. To ensure that we are using the specified Raw (dd) format, we'll check this option when prompted, then hit next:



Please Select the Destination Image Type
● Raw (dd)
○ SMART
○ E01
○ AFF

iii. Now, we choose a destination for image, as well as a name (my student ID). We also ensure that the fragment size is set to 0 mb so that we get one segment.

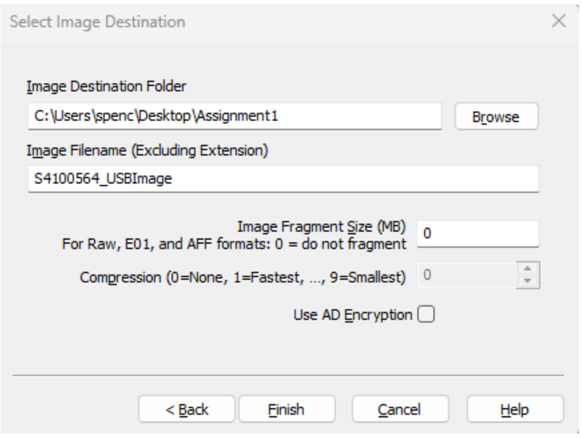


Image Destination Folder: C:\Users\spenc\Desktop\Assignment1

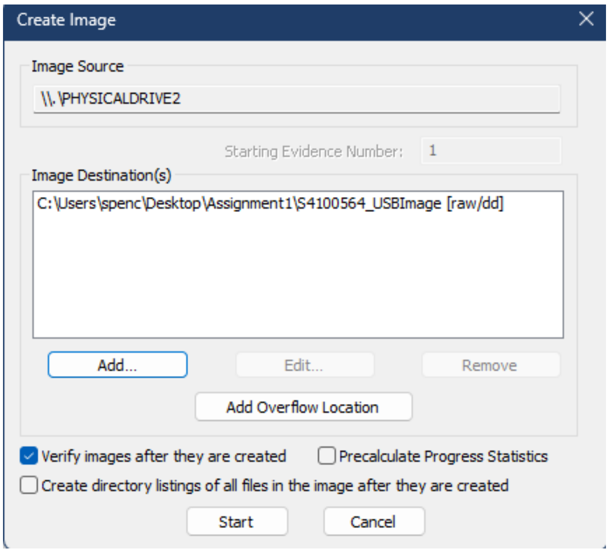
Image Filename (Excluding Extension): S4100564_USBImage

Image Fragment Size (MB)
For Raw, E01, and AFF formats: 0 = do not fragment: 0

Compression (0=None, 1=Fastest, ..., 9=Smallest): 0

Use AD Encryption □

iv. Back at the Create Image screen, we click start because we are only creating one image, and we also ensure that ”Verify images after they are created” is checked.



Create Image

Image Source: \\.\PHYSICALDRIVE2

Starting Evidence Number: 1

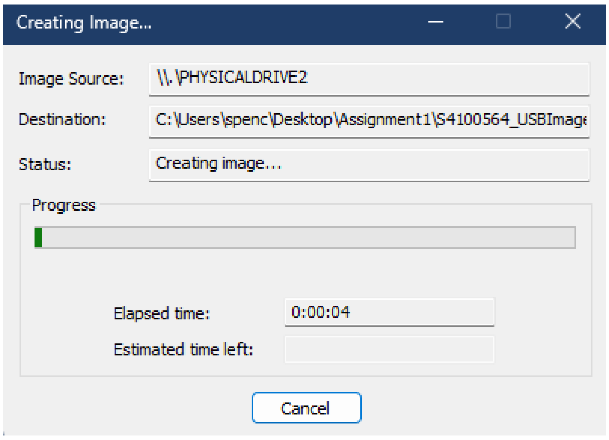
Image Destination(s): C:\Users\spenc\Desktop\Assignment1\S4100564_USBImage [raw/dd]

☑ Verify images after they are created

□ Precalculate Progress Statistics

□ Create directory listings of all files in the image after they are created

v. We will need to wait for a while for the image to be created.



Creating Image...

Image Source: \\.\PHYSICALDRIVE2

Destination: C:\Users\spenc\Desktop\Assignment1\S4100564_USBImage

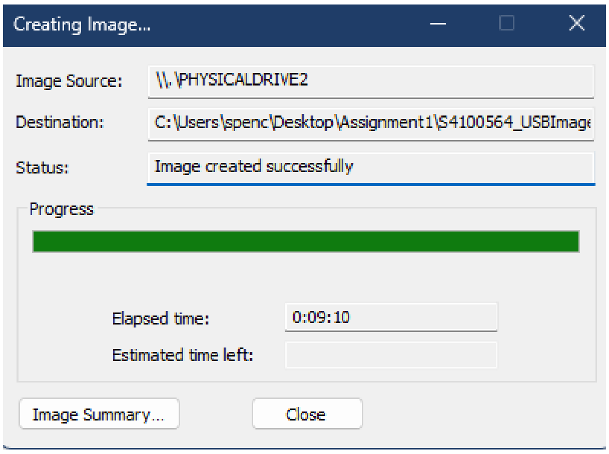
Status: Creating image...

Progress

Elapsed time: 0:00:04

Estimated time left:

vi. After it has been created, we see this screen:



Creating Image...

Image Source: \\.\PHYSICALDRIVE2

Destination: C:\Users\spenc\Desktop\Assignment1\S4100564_USBImage

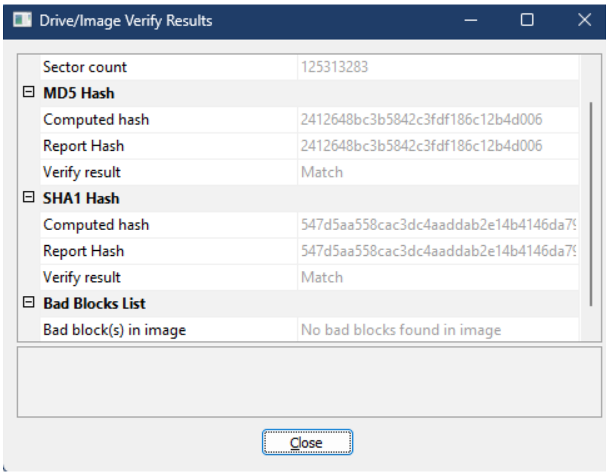
Status: Image created successfully

Progress

Elapsed time: 0:09:10

Estimated time left:

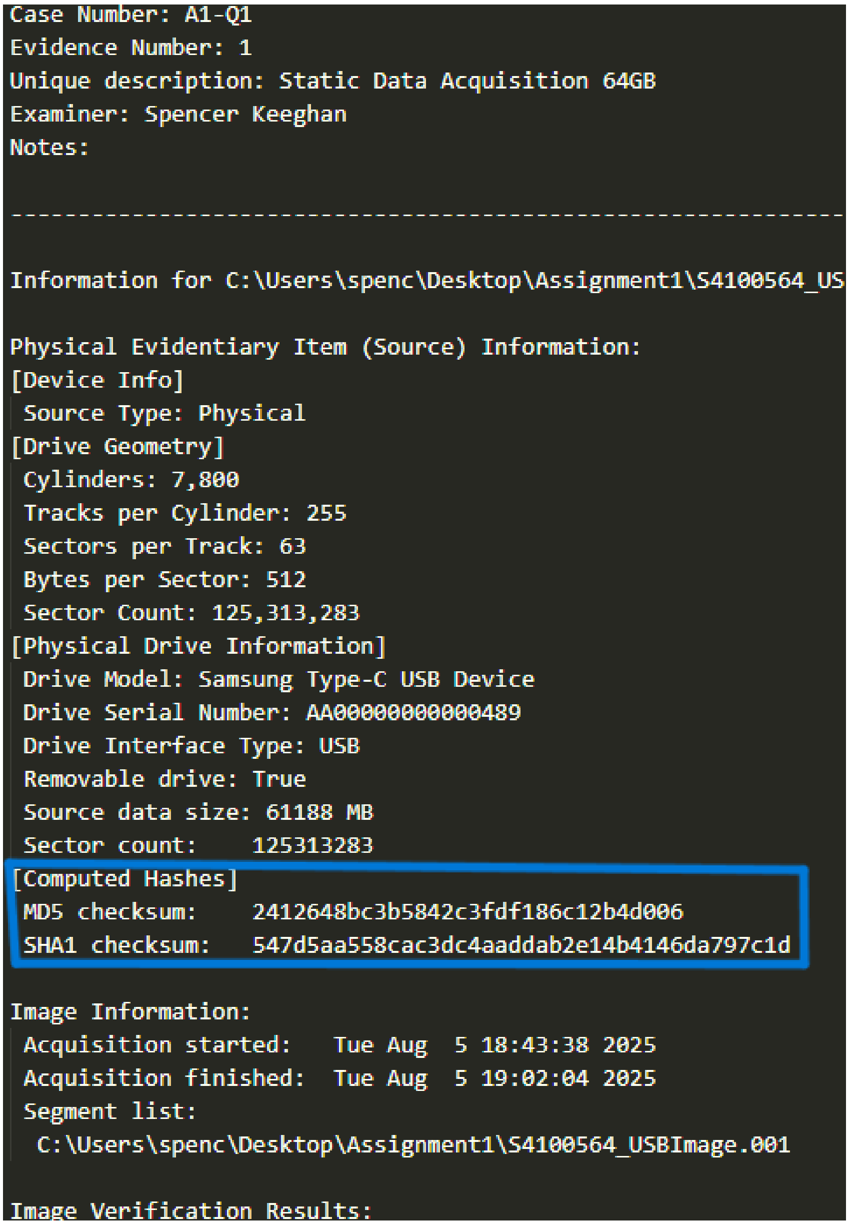
vii. We also see that the Verify result is ”Match” and that ”No Bad blocks found in image”, indicating a successful image creation.



Drive/Image Verify Results

Sector count	125313283
⊟ **MD5 Hash**	
Computed hash	2412648bc3b5842c3fdf186c12b4d006
Report Hash	2412648bc3b5842c3fdf186c12b4d006
Verify result	Match
⊟ **SHA1 Hash**	
Computed hash	547d5aa558cac3dc4aaddab2e14b4146da7
Report Hash	547d5aa558cac3dc4aaddab2e14b4146da7
Verify result	Match
⊟ **Bad Blocks List**	
Bad block(s) in image	No bad blocks found in image

b) Details of the disk image information generated by FTK Imager



```
Case Number: A1-Q1
Evidence Number: 1
Unique description: Static Data Acquisition 64GB
Examiner: Spencer Keeghan
Notes:

--------------------------------------------------------------

Information for C:\Users\spenc\Desktop\Assignment1\S4100564_US

Physical Evidentiary Item (Source) Information:
[Device Info]
 Source Type: Physical
[Drive Geometry]
 Cylinders: 7,800
 Tracks per Cylinder: 255
 Sectors per Track: 63
 Bytes per Sector: 512
 Sector Count: 125,313,283
[Physical Drive Information]
 Drive Model: Samsung Type-C USB Device
 Drive Serial Number: AA00000000000489
 Drive Interface Type: USB
 Removable drive: True
 Source data size: 61188 MB
 Sector count:    125313283
[Computed Hashes]
 MD5 checksum:    2412648bc3b5842c3fdf186c12b4d006
 SHA1 checksum:   547d5aa558cac3dc4aaddab2e14b4146da797c1d

Image Information:
 Acquisition started:   Tue Aug  5 18:43:38 2025
 Acquisition finished:  Tue Aug  5 19:02:04 2025
 Segment list:
  C:\Users\spenc\Desktop\Assignment1\S4100564_USBImage.001

Image Verification Results:
```

c) i. The disk image was created on Tue Aug 5 19:02:04 2025

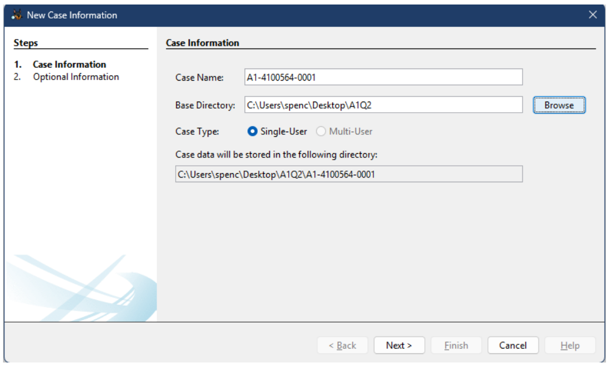
ii. MD5 checksum: 2412648bc3b5842c3fdf186c12b4d006. This is shown in the blue box in the screenshot for b).

iii. The checksum of the image matches the checksum of the physical disk. The SHA1 checksum of the image is 547d5aa558cac3dc4aaddab2e14b4146da797c1d. The verification was successful, as indicated by the "verified" status next to both the MD5 and SHA1 checksums in the report.

Q2. Analyzing Disk Image using Autopsy Tool (3 Marks)

a) We'll create a new case in Autopsy as follows:

i. First, we need to provide a case name and a base directory for this case:



New Case Information

Steps

1. Case Information
2. Optional Information

Case Information

Case Name: A1-4100564-0001

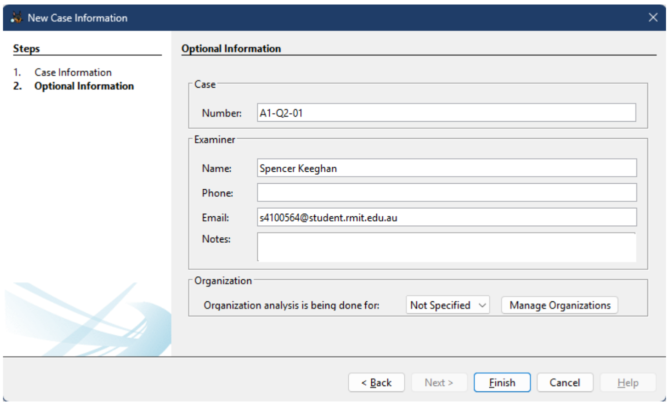
Base Directory: C:\Users\spenc\Desktop\A1Q2

Case Type: ● Single-User ○ Multi-User

Case data will be stored in the following directory:

C:\Users\spenc\Desktop\A1Q2\A1-4100564-0001

ii. Next, we will specify the case number, and I will provide my student email and full name in the Examiner details:



New Case Information

Steps

1. Case Information
2. Optional Information

Optional Information

Case

Number: A1-Q2-01

Examiner

Name: Spencer Keeghan

Phone:

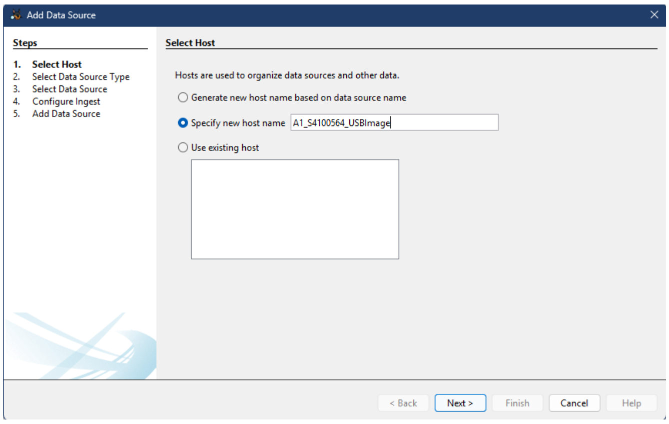
Email: s4100564@student.rmit.edu.au

Notes:

Organization

Organization analysis is being done for: Not Specified

iii. Our host will be the image that we created in Q1, so we'll specify that file as our host name:



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. Select Data Source
4. Configure Ingest
5. Add Data Source

Select Host

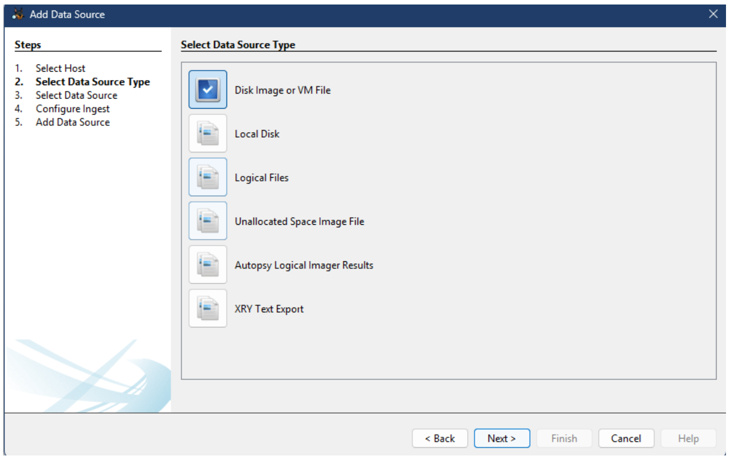
Hosts are used to organize data sources and other data.

○ Generate new host name based on data source name

● Specify new host name A1_S4100564_USBImage

○ Use existing host

iv. We will select Disk Image or VM File as our data source:



Add Data Source

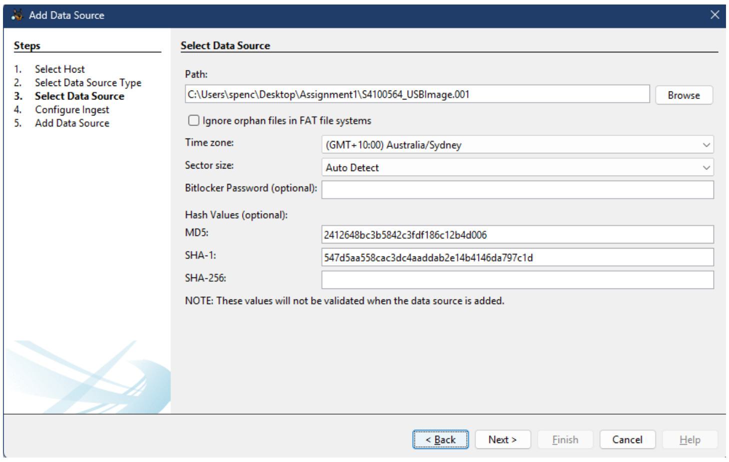
Steps

1. Select Host
2. **Select Data Source Type**
3. Select Data Source
4. Configure Ingest
5. Add Data Source

Select Data Source Type

- ☑ Disk Image or VM File
- Local Disk
- Logical Files
- Unallocated Space Image File
- Autopsy Logical Imager Results
- XRY Text Export

v. Now we need to provide the path to our USB image, and we will also provide the two hashes that were generated:



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. **Select Data Source**
4. Configure Ingest
5. Add Data Source

Select Data Source

Path: C:\Users\spenc\Desktop\Assignment1\S4100564_USBImage.001

□ Ignore orphan files in FAT file systems

Time zone: (GMT+10:00) Australia/Sydney

Sector size: Auto Detect

Bitlocker Password (optional):

Hash Values (optional):

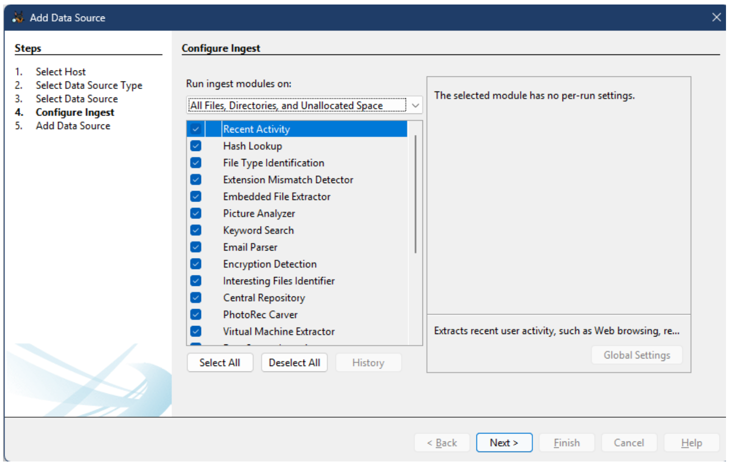
MD5: 2412648bc3b5842c3fdf186c12b4d006

SHA-1: 547d5aa558cac3dc4aaddab2e14b4146da797c1d

SHA-256:

NOTE: These values will not be validated when the data source is added.

vi. We will leave all of the default ingest modules selected:



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. Select Data Source
4. **Configure Ingest**
5. Add Data Source

Configure Ingest

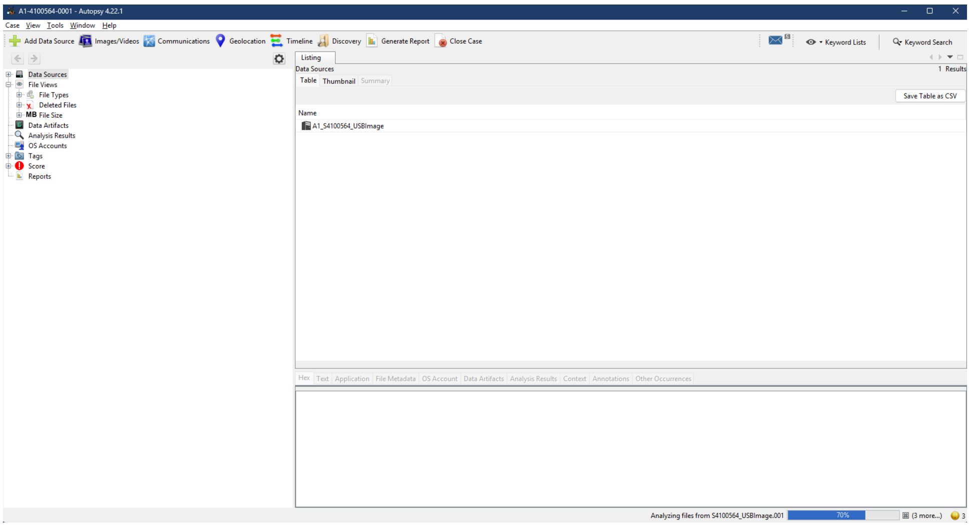
Run ingest modules on: All Files, Directories, and Unallocated Space

- ☑ Recent Activity
- ☑ Hash Lookup
- ☑ File Type Identification
- ☑ Extension Mismatch Detector
- ☑ Embedded File Extractor
- ☑ Picture Analyzer
- ☑ Keyword Search
- ☑ Email Parser
- ☑ Encryption Detection
- ☑ Interesting Files Identifier
- ☑ Central Repository
- ☑ PhotoRec Carver
- ☑ Virtual Machine Extractor

The selected module has no per-run settings.

Extracts recent user activity, such as Web browsing, re...

vii. After the case has successfully been generated, we are met with this screen:

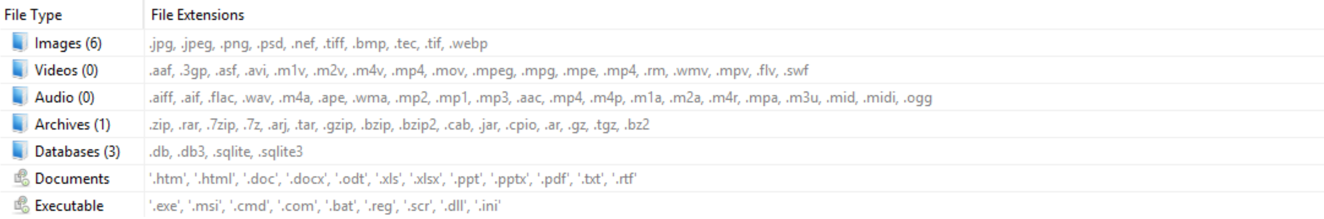


A1-4100564-0001 - Autopsy 4.22.1

A1_S4100564_USBImage

Analyzing files from S4100564_USBImage.001 70% (3 more...)

b) We can find the different types of files and their count by selecting File Views → File Types → By Extension in the top left navigation pane. Here we can see that the image contains 6 images, 1 archive and 3 databases:



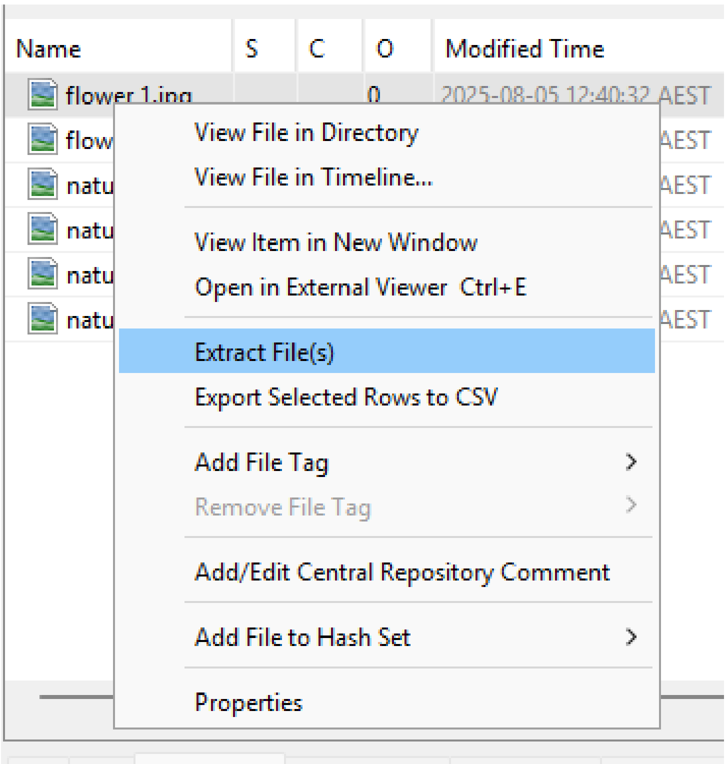
File Type	File Extensions
Images (6)	.jpg, .jpeg, .png, .psd, .nef, .tiff, .bmp, .tec, .tif, .webp
Videos (0)	.aaf, .3gp, .asf, .avi, .m1v, .m2v, .m4v, .mp4, .mov, .mpeg, .mpg, .mpe, .mp4, .rm, .wmv, .mpv, .flv, .swf
Audio (0)	.aiff, .aif, .flac, .wav, .m4a, .ape, .wma, .mp2, .mp1, .mp3, .aac, .mp4, .m4p, .m1a, .m2a, .m4r, .mpa, .m3u, .mid, .midi, .ogg
Archives (1)	.zip, .rar, .7zip, .7z, .arj, .tar, .gzip, .bzip, .bzip2, .cab, .jar, .cpio, .ar, .gz, .tgz, .bz2
Databases (3)	.db, .db3, .sqlite, .sqlite3
Documents	'.htm', '.html', '.doc', '.docx', '.odt', '.xls', '.xlsx', '.ppt', '.pptx', '.pdf', '.txt', '.rtf'
Executable	'.exe', '.msi', '.cmd', '.com', '.bat', '.reg', '.scr', '.dll', '.ini'

In tabular form:

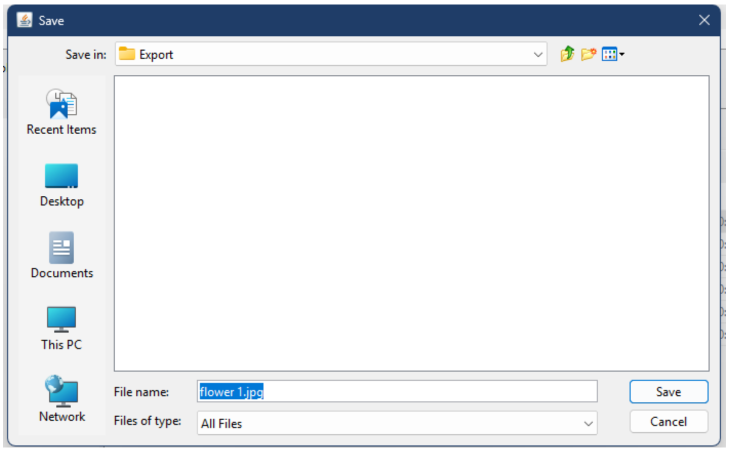
Images	**Archives**	**Databases**
.jpg	.zip	.db
.jpeg	.rar	.db3
.png	.7zip	.sqlite
.psd	.7z	.sqlite3
.nef	.arj	
.tiff	.tar	
.bmp	.gzip	
.tec	.bzip	
.tif	.bzip2	
.webp	.cab	
	.jar	
	.cpio	
	.ar	
	.gz	
	.tgz	
	.bz2	

c) To extract a file in our image, we'll follow these steps:

i. First, right click on a file and select "Extract File(s)".



ii. By default, it will save to the Export folder in the Base case directory:

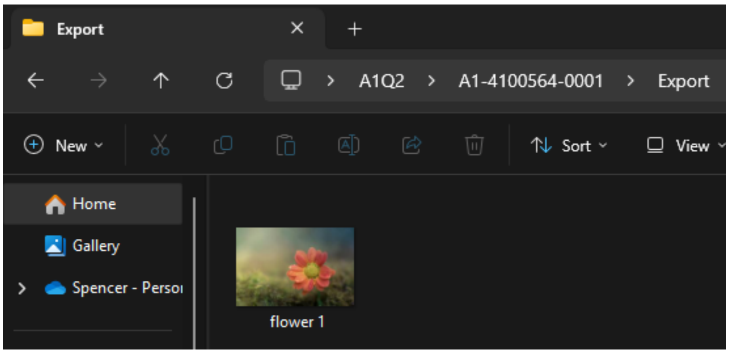


Save in: Export

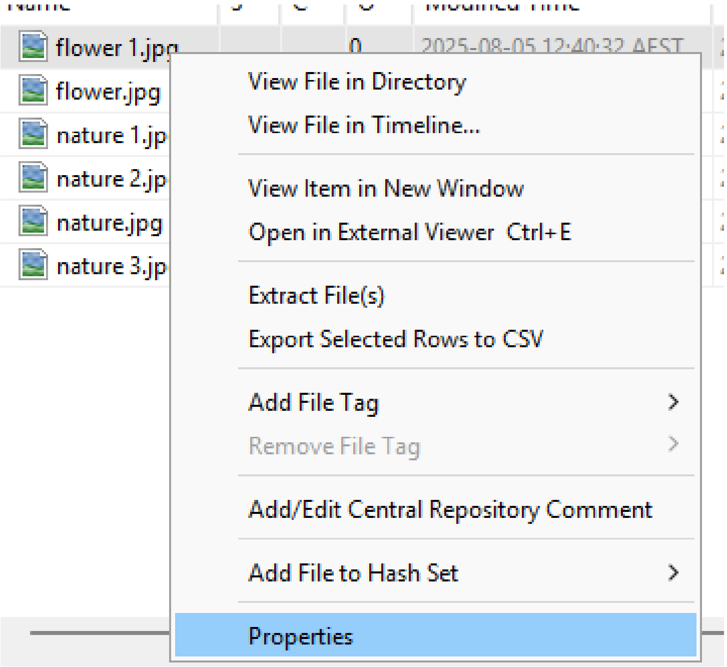
File name: flower 1.jpg

Files of type: All Files

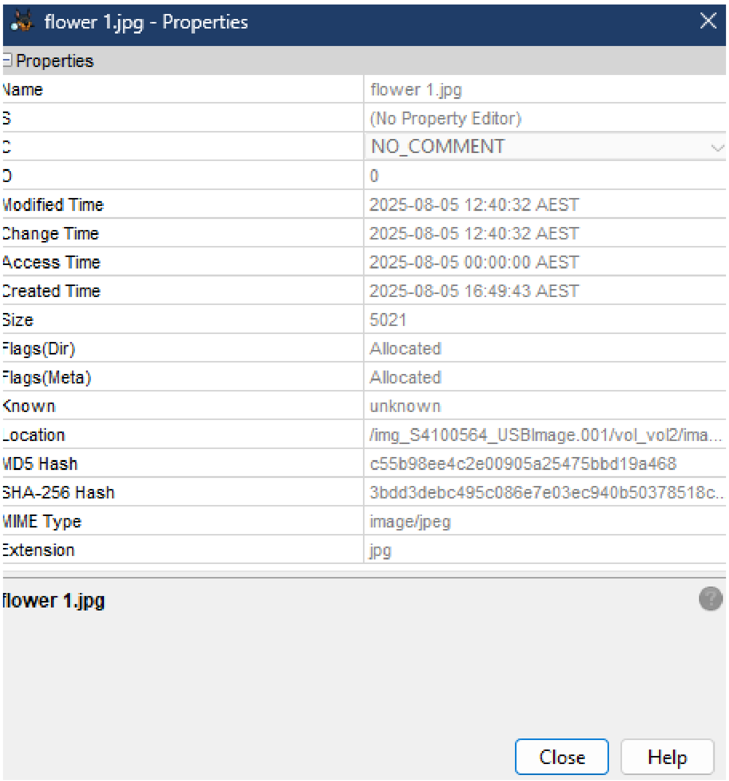
iii. We can navigate to the Export folder and see our extracted file there:



iv. We can also analyse the properties of the file by right clicking it and selecting ”Properties”



v. The properties are shown below:



Properties	
Name	flower 1.jpg
S	(No Property Editor)
C	NO_COMMENT
O	0
Modified Time	2025-08-05 12:40:32 AEST
Change Time	2025-08-05 12:40:32 AEST
Access Time	2025-08-05 00:00:00 AEST
Created Time	2025-08-05 16:49:43 AEST
Size	5021
Flags(Dir)	Allocated
Flags(Meta)	Allocated
Known	unknown
Location	/img_S4100564_USBImage.001/vol_vol2/ima...
MD5 Hash	c55b98ee4c2e00905a25475bbd19a468
SHA-256 Hash	3bdd3debc495c086e7e03ec940b50378518c...
MIME Type	image/jpeg
Extension	jpg

flower 1.jpg

Q3. Understanding of Different Digital Forensics Tools (6 Marks)

a) Overview of Static Data Acquisition Tools

ProDiscover Forensics (Windows): ProDiscover Forensics is a well-established commercial tool for the Windows platform, designed to perform comprehensive digital forensic investigations. Its core function is to create a forensically sound bit-for-bit copy of a storage device, such as a hard drive or USB stick, preserving the original evidence. It can also be used to analyze a live system's data without altering it [4, 7].

Advantages: One of its key strengths is the ability to perform acquisitions remotely over a network, which is useful in corporate environments. It also integrates a suite of analysis tools, allowing an examiner to not only image a disk but also to immediately begin searching for keywords, viewing files, and generating reports within the same application.

Disadvantages: The primary drawback is its cost. As a commercial product, the license can be expensive, making it less accessible for academic use or smaller labs. Additionally, like many commercial tools, its updates may not always keep pace with the latest developments in file systems or encryption, potentially leaving gaps in its capabilities.

Guymager (Linux): Guymager is a free, open-source forensic imager that comes pre-installed with Kali Linux. It provides a user-friendly GUI for creating disk images in various formats, making the process straightforward even for those less comfortable with command-line tools [5].

Advantages: Its main appeal is that it is free, open-source, and reliable. It is multi-threaded, meaning it can use multiple processor cores to speed up the imaging process significantly. It also supports several forensic image formats, including the standard raw 'dd' format and the more advanced E01 and AFF formats, giving the examiner flexibility.

Disadvantages: While excellent for imaging, it is not a full forensic suite; it does not include any analysis capabilities. An examiner must use other tools like Autopsy to analyze the images it creates. Furthermore, being open-source, it lacks official customer support, so users must rely on community forums for help [11].

Cellebrite Digital Collector (macOS): This is a specialized commercial tool, formerly known as MacQuisition, built specifically for acquiring data from Apple's macOS and iOS devices. It is renowned for its ability to handle Apple-specific hardware and software, such as Fusion Drives and the T2 security chip [3].

Advantages: Its most critical feature is its unique ability to create decrypted physical images from modern Macs equipped with the T2 security chip, something most other tools cannot do. It can also perform live acquisitions from running Macs and capture RAM, making it a versatile tool for Apple-focused investigations.

Disadvantages: The tool is very expensive, placing it out of reach for many. Its focus is also exclusively on Apple products, so it cannot be used for Windows or standard Linux systems. The complexity of its features may also require specialized training to use effectively.

b) **Overview of Memory Acquisition Tools**

Magnet RAM Capture (Windows): This is a free and widely used tool from Magnet Forensics for capturing the volatile memory (RAM) from a live Windows computer. It's designed to be a simple, no-fuss utility that quickly dumps the contents of physical memory to a file for later analysis [6].

Advantages: It is completely free to use and has a very small memory footprint, which is crucial because it minimizes the tool's impact on the live data being captured. A significant technical advantage is its ability to acquire RAM from modern Windows 10 systems even when security features like Virtual Secure Mode (VSM) are enabled [6]. This is a vital capability, as many other acquisition tools can be blocked by these advanced security measures.

Disadvantages: Its simplicity is also a limitation. It is purely an acquisition tool and offers no analysis features. It also provides very few configuration options, so examiners who need more control over the acquisition process might find it too basic for their needs.

LiME (Linux Memory Extractor): LiME is a popular open-source tool for acquiring RAM from Linux and Android devices. It operates as a loadable kernel module, which allows it to access memory directly and efficiently [2].

Advantages: As a kernel module, it provides robust, low-level access to memory. A key feature is its ability to send the memory image over the network instead of saving it to the local disk, which is essential when dealing with systems that may not have enough free disk space or when trying to minimize interaction with the storage device.

Disadvantages: The main challenge with LiME is that it requires a kernel module that matches the specific kernel version of the target system. If a pre-compiled module is not available, one must be compiled on a similar system, which can be a complex and time-consuming process, especially during a live incident response.

OSXpmem (macOS): OSXpmem is a command-line tool for acquiring memory from macOS systems. It is part of the pmem suite of tools, which was developed alongside the Rekall memory analysis framework [8, 12].

Advantages: It is free and open-source. One of its main strengths is its support for the advanced AFF4 evidence container format, which can store memory images, metadata, and other forensic data in a single, verifiable file [8]. Being a command-line tool also makes it easy to script and use in automated workflows.

Disadvantages: Its reliance on the command line can be a hurdle for examiners who are more accustomed to GUIs. More significantly, it functions by loading a kernel extension, which can be blocked by macOS's built-in security features like System Integrity Protection (SIP), making it difficult to run on modern, fully updated Macs without first altering the security settings.

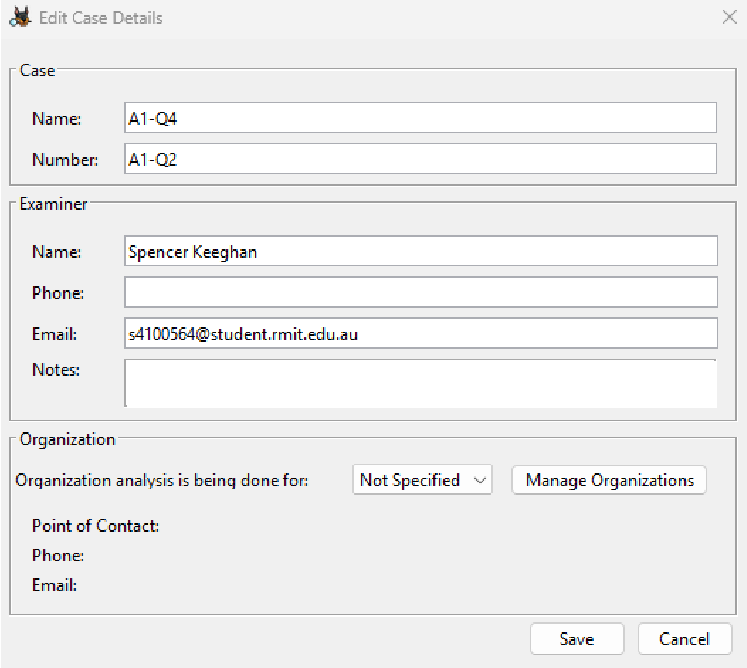
PART-2: FORENSIC ANALYSIS (18 Marks)

Q4. Disk Image Artifact Analysis (13 Marks)

a) *File Listing and carving*

i. We will begin by creating a case in Autopsy, following the steps we took in **Part 1, Question 2**:

First, we will decide on the initial case details:



Edit Case Details

Case

Name: A1-Q4

Number: A1-Q2

Examiner

Name: Spencer Keeghan

Phone:

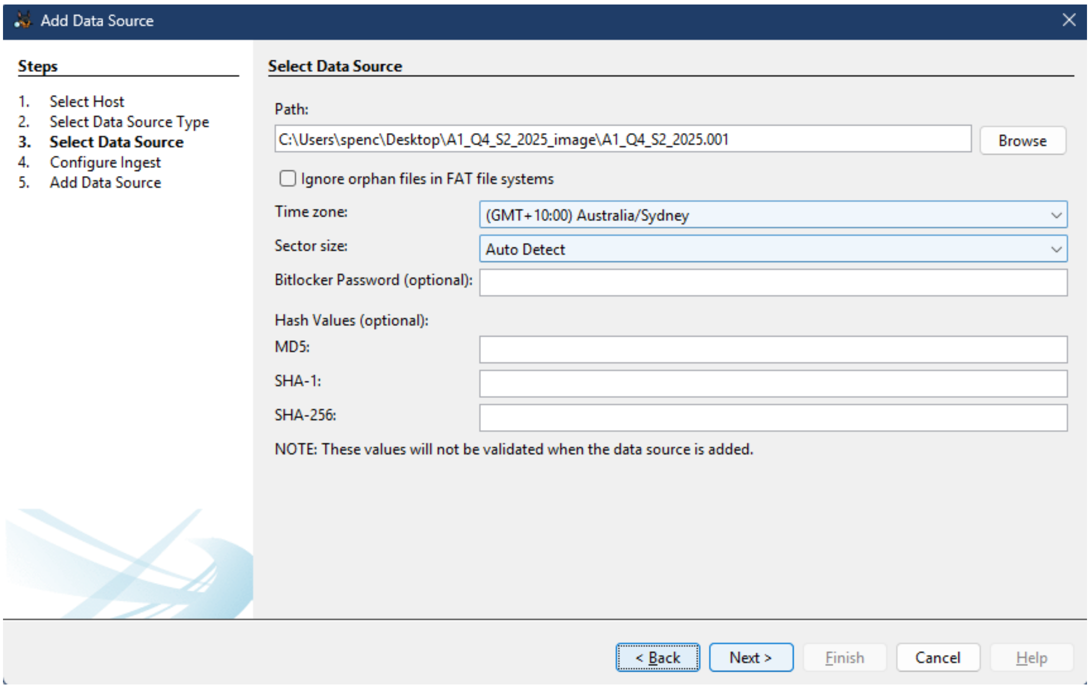
Email: s4100564@student.rmit.edu.au

Notes:

Organization

Organization analysis is being done for: Not Specified

For our data source, we will choose the disk image created from *Specter's* USB drive and external hard disk:



Add Data Source

Steps

1. Select Host
2. Select Data Source Type
3. **Select Data Source**
4. Configure Ingest
5. Add Data Source

Select Data Source

Path: C:\Users\spenc\Desktop\A1_Q4_S2_2025_image\A1_Q4_S2_2025.001

□ Ignore orphan files in FAT file systems

Time zone: (GMT+10:00) Australia/Sydney

Sector size: Auto Detect

Bitlocker Password (optional):

Hash Values (optional):

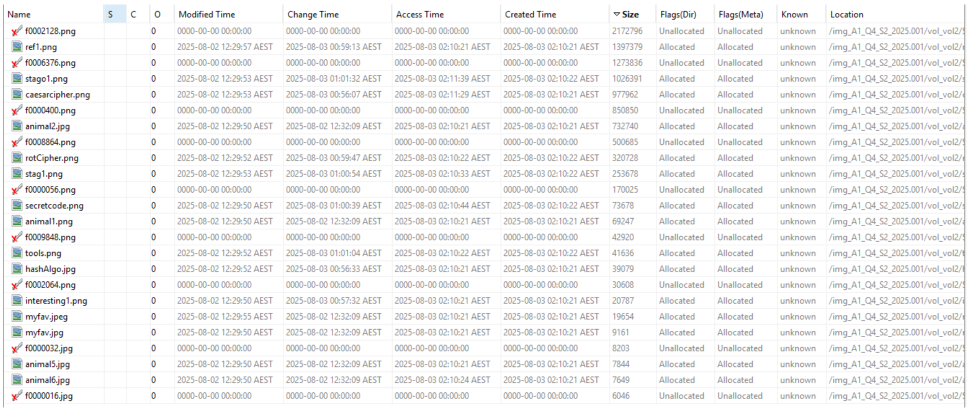
MD5:

SHA-1:

SHA-256:

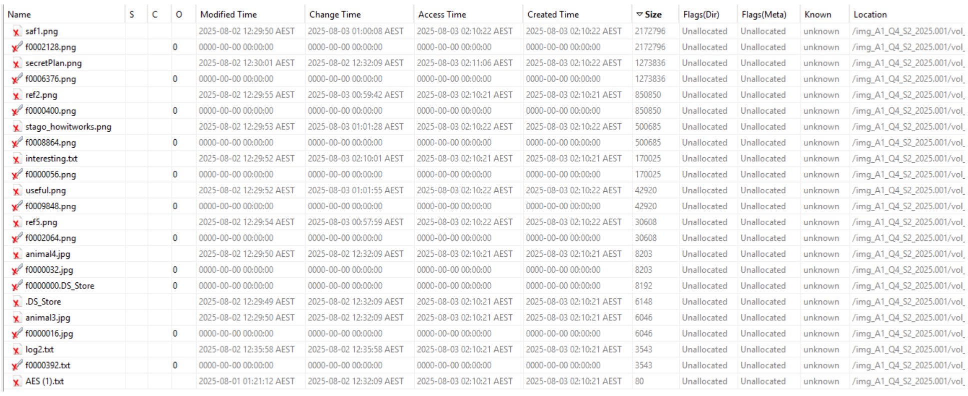
NOTE: These values will not be validated when the data source is added.

After creating the case, we can see the files on the image:



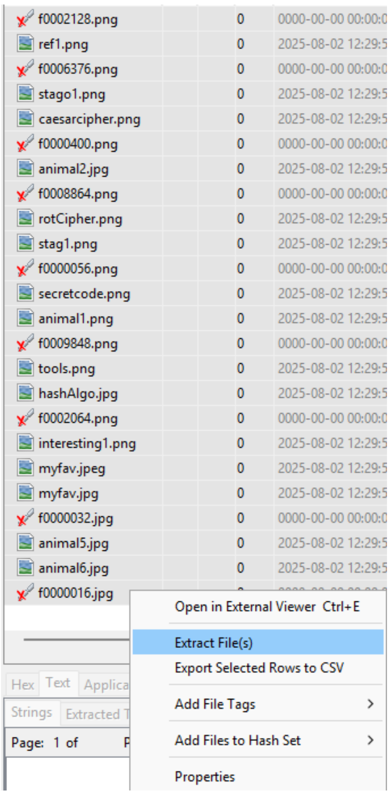
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	▽ Size	Flags(Dir)	Flags(Meta)	Known	Location
f0002128.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	2172796	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
ref1.png			0	2025-08-02 12:29:57 AEST	2025-08-03 00:59:13 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	1397379	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0006376.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	1273836	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
stago1.png			0	2025-08-02 12:29:53 AEST	2025-08-03 01:01:32 AEST	2025-08-03 02:11:39 AEST	2025-08-03 02:10:22 AEST	1026391	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
caesarcipher.png			0	2025-08-02 12:29:53 AEST	2025-08-03 00:56:07 AEST	2025-08-03 02:11:29 AEST	2025-08-03 02:10:21 AEST	977962	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0000400.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	850850	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
animal2.jpg			0	2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	732740	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0008864.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	500685	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
rotCipher.png			0	2025-08-02 12:29:52 AEST	2025-08-03 00:59:47 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	320728	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
stag1.png			0	2025-08-02 12:29:53 AEST	2025-08-03 01:00:54 AEST	2025-08-03 02:10:33 AEST	2025-08-03 02:10:22 AEST	253678	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0000056.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	170025	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
secretcode.png			0	2025-08-02 12:29:50 AEST	2025-08-03 01:00:39 AEST	2025-08-03 02:10:44 AEST	2025-08-03 02:10:22 AEST	73678	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
animal1.png			0	2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	69247	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0009848.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	42920	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
tools.png			0	2025-08-02 12:29:52 AEST	2025-08-03 01:01:04 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	41636	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
hashAlgo.jpg			0	2025-08-02 12:29:52 AEST	2025-08-03 00:56:33 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	39079	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0002064.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	30608	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
interesting1.png			0	2025-08-02 12:29:50 AEST	2025-08-03 00:57:32 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	20787	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
myfav.jpeg			0	2025-08-02 12:29:55 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	19654	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
myfav.jpg			0	2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	9161	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0000032.jpg			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	8203	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
animal5.jpg			0	2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	7844	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
animal6.jpg			0	2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:24 AEST	2025-08-03 02:10:21 AEST	7649	Allocated	Allocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/
f0000016.jpg			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	6046	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_vol2/

We can also view the deleted files:



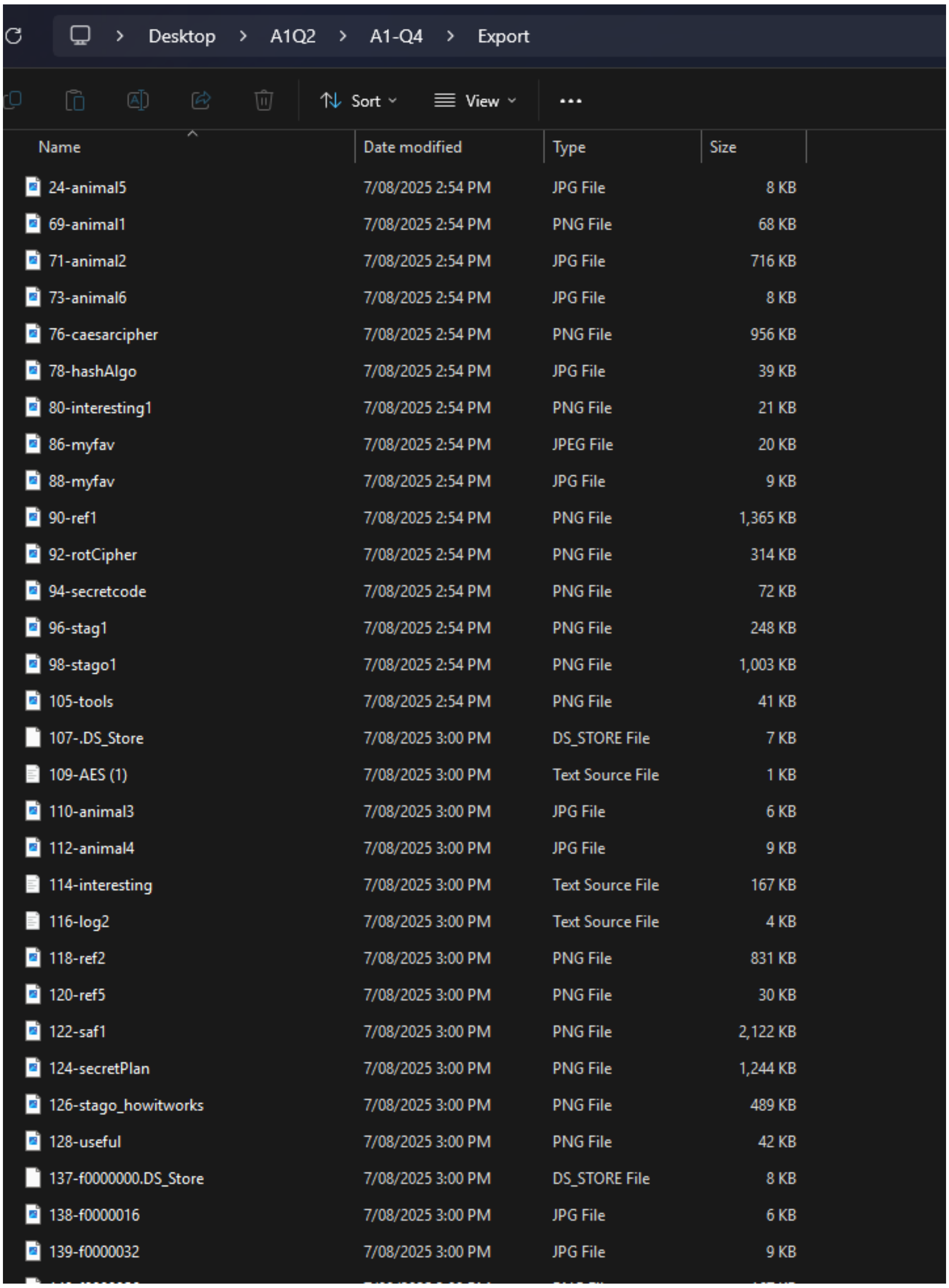
Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	▽ Size	Flags(Dir)	Flags(Meta)	Known	Location
saf1.png				2025-08-02 12:29:50 AEST	2025-08-03 01:00:08 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	2172796	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0002128.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	2172796	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
secretPlan.png				2025-08-02 12:30:01 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:11:06 AEST	2025-08-03 02:10:22 AEST	1273836	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0006376.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	1273836	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
ref2.png				2025-08-02 12:29:55 AEST	2025-08-03 00:59:42 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	850850	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000400.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	850850	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
stago_howitworks.png				2025-08-02 12:29:53 AEST	2025-08-03 01:01:28 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	500685	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0008864.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	500685	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
interesting.txt				2025-08-02 12:29:52 AEST	2025-08-03 02:10:01 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	170025	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000056.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	170025	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
useful.png				2025-08-02 12:29:52 AEST	2025-08-03 01:01:55 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	42920	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0009848.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	42920	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
ref5.png				2025-08-02 12:29:54 AEST	2025-08-03 00:57:59 AEST	2025-08-03 02:10:22 AEST	2025-08-03 02:10:22 AEST	30608	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0002064.png			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	30608	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
animal4.jpg				2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	8203	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000032.jpg			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	8203	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000000.DS_Store			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	8192	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
.DS_Store				2025-08-02 12:29:49 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	6148	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
animal3.jpg				2025-08-02 12:29:50 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	6046	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000016.jpg			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	6046	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
log2.txt				2025-08-02 12:35:58 AEST	2025-08-02 12:35:58 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	3543	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
f0000392.txt			0	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	3543	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_
AES (1).txt				2025-08-01 01:21:12 AEST	2025-08-02 12:32:09 AEST	2025-08-03 02:10:21 AEST	2025-08-03 02:10:21 AEST	80	Unallocated	Unallocated	unknown	/img_A1_Q4_S2_2025.001/vol_

ii. To recover all of the files, including hidden and deleted files, we will right click them and select ”Extract”:



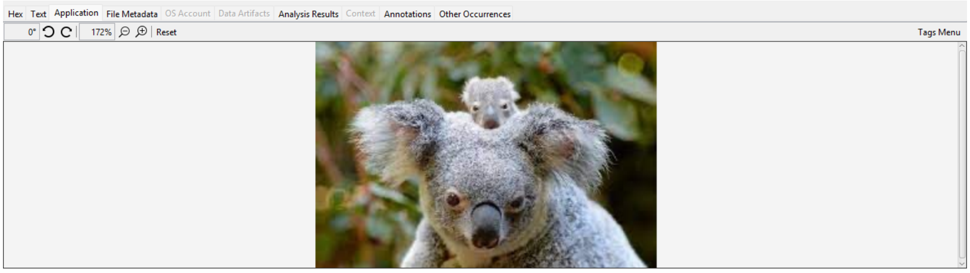
f0002128.png			0	0000-00-00 00:00:0
ref1.png			0	2025-08-02 12:29:5
f0006376.png			0	0000-00-00 00:00:0
stago1.png			0	2025-08-02 12:29:5
caesarcipher.png			0	2025-08-02 12:29:5
f0000400.png			0	0000-00-00 00:00:0
animal2.jpg			0	2025-08-02 12:29:5
f0008864.png			0	0000-00-00 00:00:0
rotCipher.png			0	2025-08-02 12:29:5
stag1.png			0	2025-08-02 12:29:5
f0000056.png			0	0000-00-00 00:00:0
secretcode.png			0	2025-08-02 12:29:5
animal1.png			0	2025-08-02 12:29:5
f0009848.png			0	0000-00-00 00:00:0
tools.png			0	2025-08-02 12:29:5
hashAlgo.jpg			0	2025-08-02 12:29:5
f0002064.png			0	0000-00-00 00:00:0
interesting1.png			0	2025-08-02 12:29:5
myfav.jpeg			0	2025-08-02 12:29:5
myfav.jpg			0	2025-08-02 12:29:5
f0000032.jpg			0	0000-00-00 00:00:0
animal5.jpg			0	2025-08-02 12:29:5
animal6.jpg			0	2025-08-02 12:29:5
f0000016.jpg				

By default, the files will save to the "Export" folder in our base case directory:

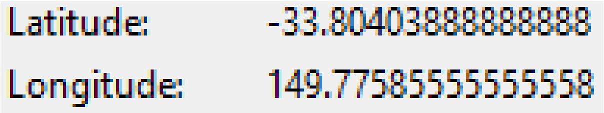


Name	Date modified	Type	Size
24-animal5	7/08/2025 2:54 PM	JPG File	8 KB
69-animal1	7/08/2025 2:54 PM	PNG File	68 KB
71-animal2	7/08/2025 2:54 PM	JPG File	716 KB
73-animal6	7/08/2025 2:54 PM	JPG File	8 KB
76-caesarcipher	7/08/2025 2:54 PM	PNG File	956 KB
78-hashAlgo	7/08/2025 2:54 PM	JPG File	39 KB
80-interesting1	7/08/2025 2:54 PM	PNG File	21 KB
86-myfav	7/08/2025 2:54 PM	JPEG File	20 KB
88-myfav	7/08/2025 2:54 PM	JPG File	9 KB
90-ref1	7/08/2025 2:54 PM	PNG File	1,365 KB
92-rotCipher	7/08/2025 2:54 PM	PNG File	314 KB
94-secretcode	7/08/2025 2:54 PM	PNG File	72 KB
96-stag1	7/08/2025 2:54 PM	PNG File	248 KB
98-stago1	7/08/2025 2:54 PM	PNG File	1,003 KB
105-tools	7/08/2025 2:54 PM	PNG File	41 KB
107-.DS_Store	7/08/2025 3:00 PM	DS_STORE File	7 KB
109-AES (1)	7/08/2025 3:00 PM	Text Source File	1 KB
110-animal3	7/08/2025 3:00 PM	JPG File	6 KB
112-animal4	7/08/2025 3:00 PM	JPG File	9 KB
114-interesting	7/08/2025 3:00 PM	Text Source File	167 KB
116-log2	7/08/2025 3:00 PM	Text Source File	4 KB
118-ref2	7/08/2025 3:00 PM	PNG File	831 KB
120-ref5	7/08/2025 3:00 PM	PNG File	30 KB
122-saf1	7/08/2025 3:00 PM	PNG File	2,122 KB
124-secretPlan	7/08/2025 3:00 PM	PNG File	1,244 KB
126-stago_howitworks	7/08/2025 3:00 PM	PNG File	489 KB
128-useful	7/08/2025 3:00 PM	PNG File	42 KB
137-f0000000.DS_Store	7/08/2025 3:00 PM	DS_STORE File	8 KB
138-f0000016	7/08/2025 3:00 PM	JPG File	6 KB
139-f0000032	7/08/2025 3:00 PM	JPG File	9 KB

b) The image ”myfav.jpeg” contains the text ”Taken near my House”. This was found by performing a keyword search for the string ”House”. We will use this to determine the possible hideout location of the suspect:

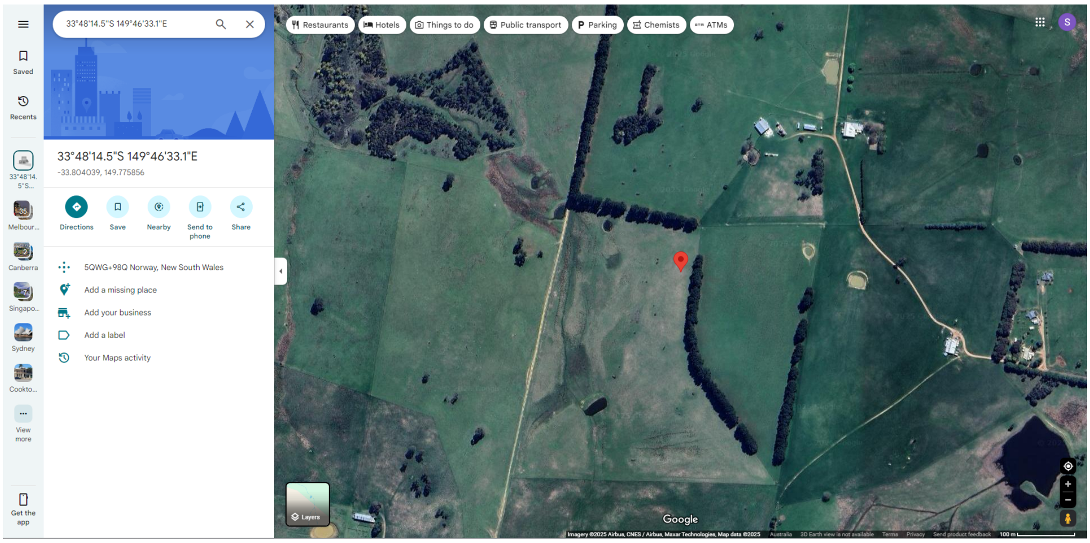


Analysing the image, we see the following geolocation data under the ”Analysis results” tab in Autopsy:



Latitude:	-33.80403888888888
Longitude:	149.77585555555558

Putting this into Google Maps, we see that the image was taken in Norway, New South Wales. The nearby house could be the possible hideout location of *Specter*.



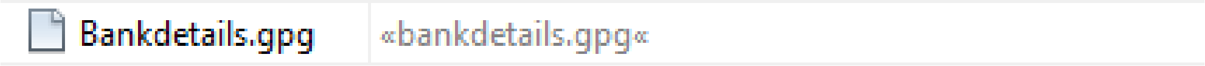
33°48'14.5"S 149°46'33.1"E

33°48'14.5"S 149°46'33.1"E

-33.804039, 149.775856

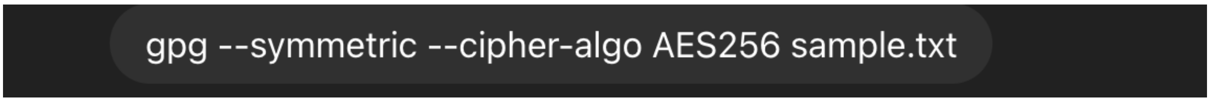
5QWG+98Q Norway, New South Wales

c) To begin the decryption of the bank details, we first need to find the encrypted file. A keyword search for ”bank” returns this .gpg file:



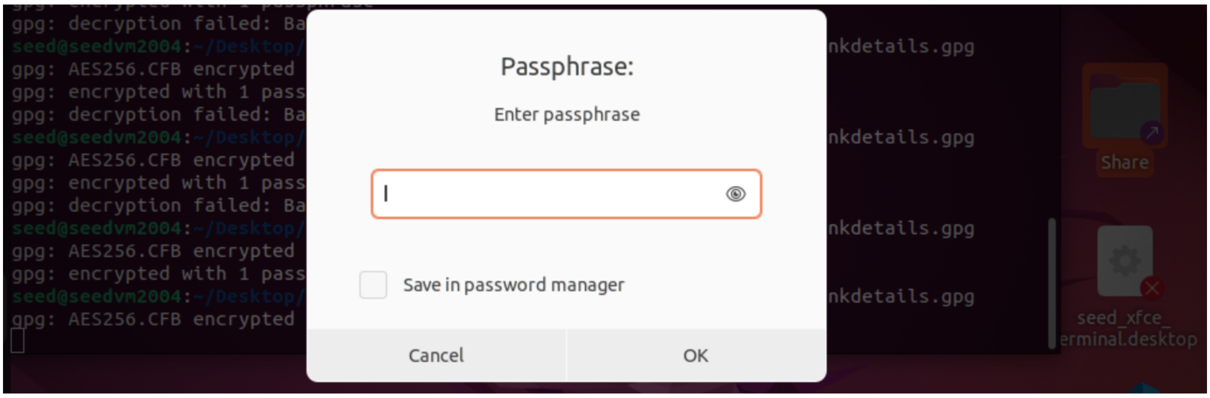
Bankdetails.gpg «bankdetails.gpg«

Moreover, we can assume that symmetric encryption was used to encrypt this file, thanks to this hint:



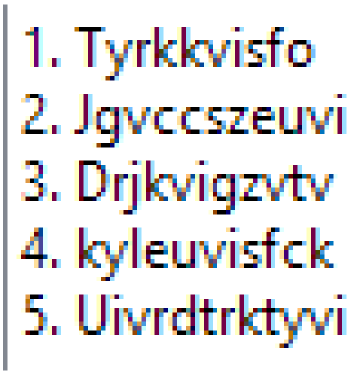
```
gpg --symmetric --cipher-algo AES256 sample.txt
```

Next, we need to find the password so that we can decrypt the file:



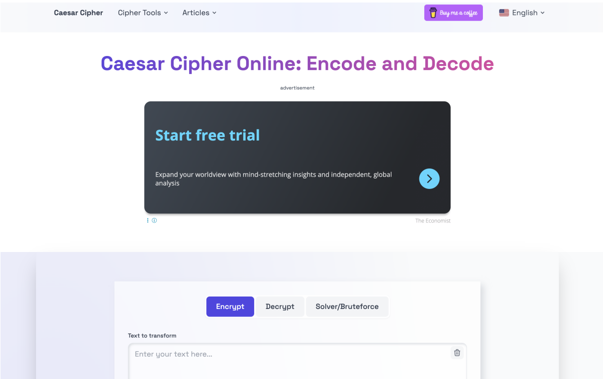
```
gpg: decryption failed: Ba
seed@seedvm2004:~/Desktop/                              nkdetails.gpg
gpg: AES256.CFB encrypted
gpg: encrypted with 1 pass
gpg: decryption failed: Ba
seed@seedvm2004:~/Desktop/                              nkdetails.gpg
gpg: AES256.CFB encrypted
gpg: encrypted with 1 pass
gpg: decryption failed: Ba
seed@seedvm2004:~/Desktop/                              nkdetails.gpg
gpg: AES256.CFB encrypted
gpg: encrypted with 1 pass
seed@seedvm2004:~/Desktop/                              nkdetails.gpg
gpg: AES256.CFB encrypted
```

The most obvious place for the password to be stored would be the file named ”AES(1).txt”, that contains 5 seemingly random strings of characters:



1. Tyrkkvisfo
2. Jgvccszeuvi
3. Drjkvigzvtv
4. kyleuvisfck
5. Uivrdtrktyvi

After trying each of these passwords and learning that none of them work, I assume that the passwords themselves must be somehow encrypted. So i re-examine the files on the image and find a few that mention caesar ciphers or just ROT ciphers in general:

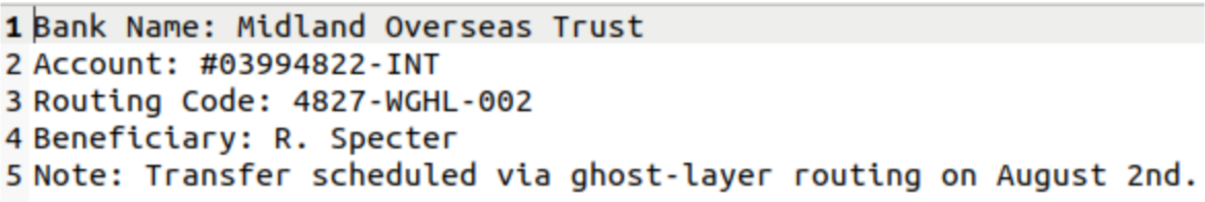


Using an online ROT Cipher tool [1], I discovered that the five words from the `AES.txt` file were encrypted using a ROT-9 cipher. Decrypting them by shifting each letter forward by 9 positions revealed the following English words:

- `Tyrkkvisfo` → **Chatterbox**
- `Jgvccszeuvi` → **Spellbinder**
- `Drjkvigzvtv` → **Masterpiece**
- `kyleuvisfck` → **thunderbolt**

- `Uivrdtrktyvi` → **Dreamcatcher**

And after trying each of these passwords again, *thunderbolt* was the one to decrypt the bank details, giving us the following information:



```
Bank Name: Midland Overseas Trust
Account: #03994822-INT
Routing Code: 4827-WGHL-002
Beneficiary: R. Specter
Note: Transfer scheduled via ghost-layer routing on August 2nd.
```

d) Now, we need to recover the Vault PIN and Drawer Password. There are a few clues that hint towards the information being hidden via steganography, such as the image below and the question itself!

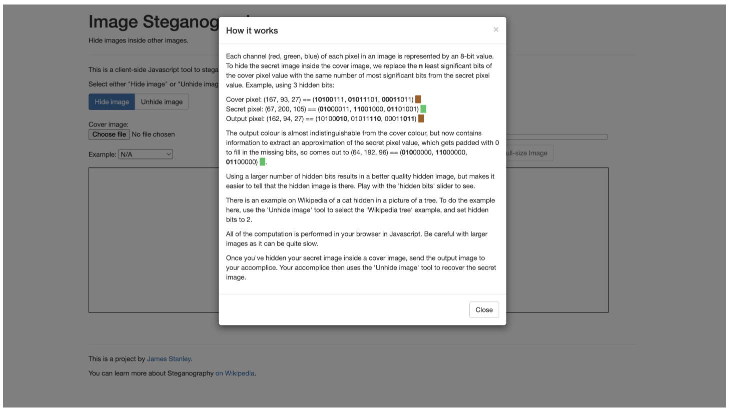


Image Steganog
Hide images inside other images.
This is a client-side Javascript tool to stega
Select either "Hide image" or "Unhide imag
Hide image
Unhide image
Cover image:
Choose file
No file chosen
Example: N/A
ull-size Image
×
Close

How it works

Each channel (red, green, blue) of each pixel in an image is represented by an 8-bit value. To hide the secret image inside the cover image, we replace the **n** least significant bits of the cover pixel value with the same number of most significant bits from the secret pixel value. Example, using 3 hidden bits:

Cover pixel: (167, 93, 27) == (**10100**111, **01011**101, **00011**011) 🟫
Secret pixel: (67, 200, 105) == (**010**00011, **110**01000, **011**01001) 🟩
Output pixel: (162, 94, 27) == (10100**010**, 01011**110**, 00011**011**) 🟫

The output colour is almost indistinguishable from the cover colour, but now contains information to extract an approximation of the secret pixel value, which gets padded with 0 to fill in the missing bits, so comes out to (64, 192, 96) == (**010**00000, **110**00000, **011**00000) 🟩.

Using a larger number of hidden bits results in a better quality hidden image, but makes it easier to tell that the hidden image is there. Play with the 'hidden bits' slider to see.

There is an example on Wikipedia of a cat hidden in a picture of a tree. To do the example here, use the 'Unhide image' tool to select the 'Wikipedia tree' example, and set hidden bits to 2.

All of the computation is performed in your browser in Javascript. Be careful with larger images as it can be quite slow.

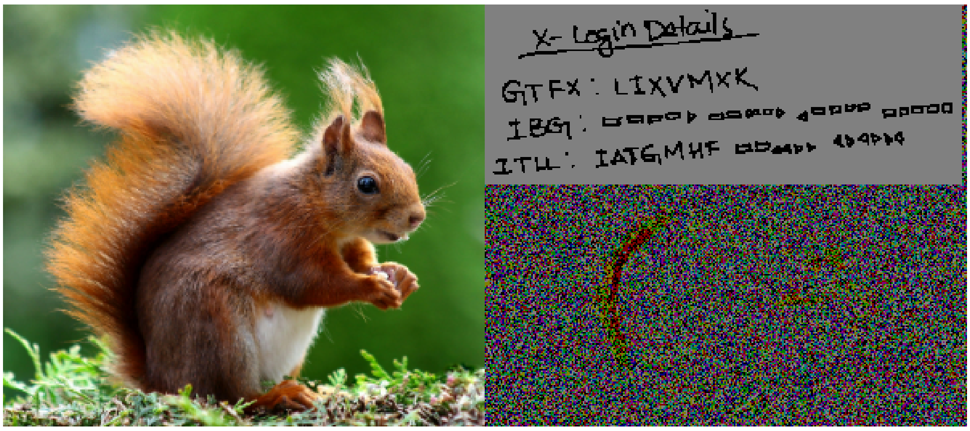
Once you've hidden your secret image inside a cover image, send the output image to your accomplice. Your accomplice then uses the 'Unhide image' tool to recover the secret image.

This is a project by James Stanley.

You can learn more about Steganography on Wikipedia.

1. https://www.edchart.com/free-online-converters/steganographic-decoder.php
2. https://incoherency.co.uk/image-steganography/#

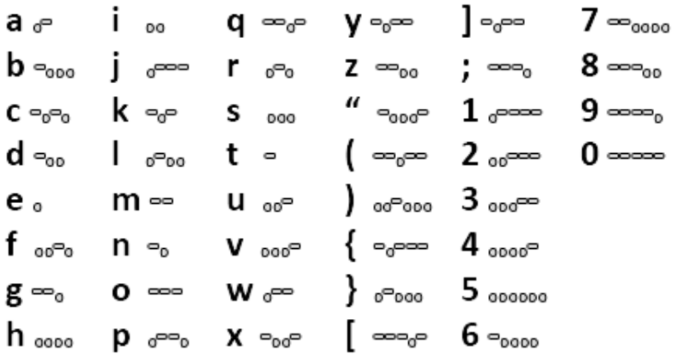
Due to these hints, I assume that I need to use online tools [9, 10] to recover the PIN and password. So I ran each .pdf or .jpg file through these tools until I found a deleted image of a squirrel named ”f0000056”, and when putting this into the online steganography tool, we see this message appear:



Because the text seemed random, I ran it through the same caesar cipher tool used earlier and got this:

NAME: SPECTER
PASS: PHANTOM
PIN:

However, its clear that the password and pin are encrypted using another strategy. There were various images relating to text encryption methods, such as a polybius square, braille and morse code. As the first two methods don't work with numbers, I focused on the image that contained morse code:



a	·–	i	··	q	––·–	y	–·––	]	–·––	7	––····
b	–···	j	·–––	r	·–·	z	––··	;	–––·	8	–––··
c	–·–·	k	–·–	s	···	“	–···–	1	·––––	9	––––·
d	–··	l	·–··	t	–	(	––·––	2	··–––	0	–––––
e	·	m	––	u	··–	)	··–···	3	···––		
f	··–·	n	–·	v	···–	{	–·–––	4	····–		
g	––·	o	–––	w	·––	}	·–···	5	······		
h	····	p	·––·	x	–··–	[	–––·–	6	–····		

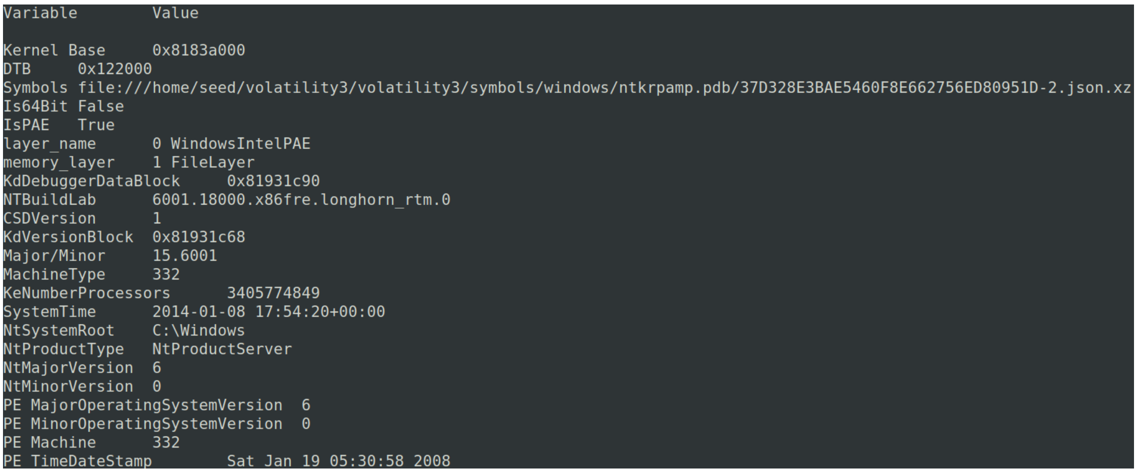
From this, the password and pin become:

PASS: PHANTOM75
PIN : 9910

Therefore, we have finally found Specter's Vault Pin (9910) and Drawer Password (PHANTOM75). A combination of image steganography, caesar ciphers and morse code was used to hide these details.

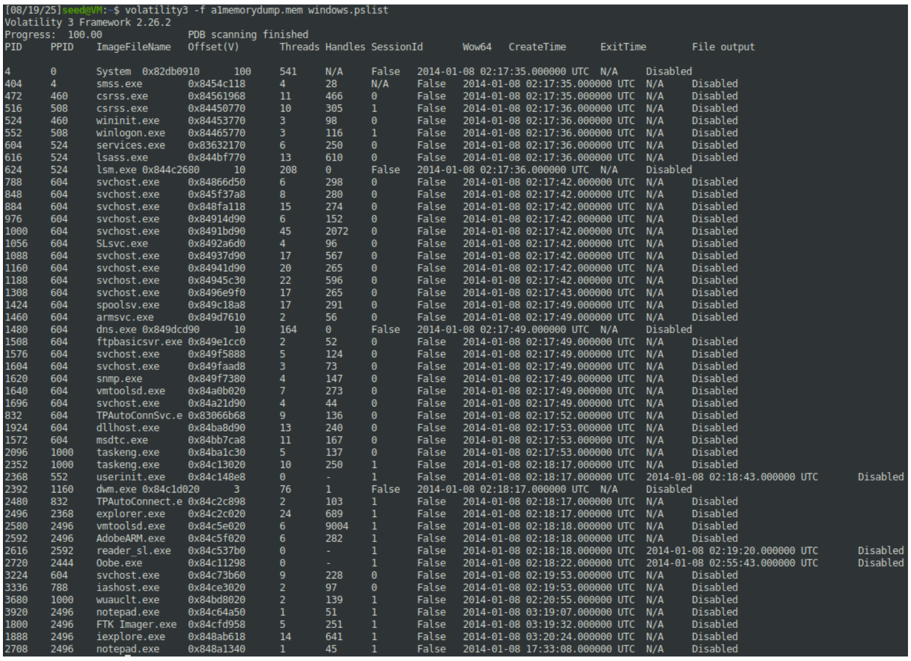
Q5. Memory Data Analysis from Computer Memory (5 Marks)

a) To get some basic information about the memory dump, like its OS and version, we use the command `volatility3 -f a1memorydump.mem windows.info`. This information is shown below:



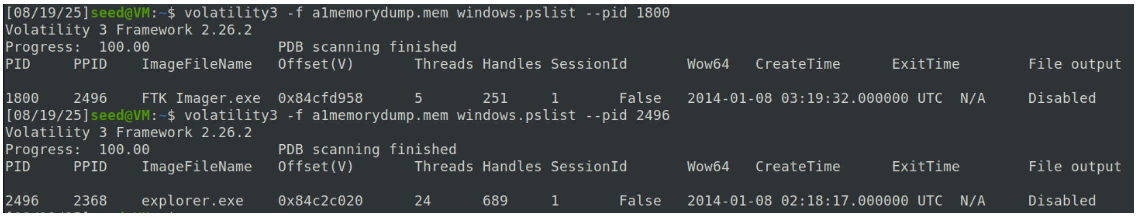
```
Variable        Value

Kernel Base     0x8183a000
DTB     0x122000
Symbols file:///home/seed/volatility3/volatility3/symbols/windows/ntkrpamp.pdb/37D328E3BAE5460F8E662756ED80951D-2.json.xz
Is64Bit False
IsPAE   True
layer_name      0 WindowsIntelPAE
memory_layer    1 FileLayer
KdDebuggerDataBlock     0x81931c90
NTBuildLab      6001.18000.x86fre.longhorn_rtm.0
CSDVersion      1
KdVersionBlock  0x81931c68
Major/Minor     15.6001
MachineType     332
KeNumberProcessors      3405774849
SystemTime      2014-01-08 17:54:20+00:00
NtSystemRoot    C:\Windows
NtProductType   NtProductServer
NtMajorVersion  6
NtMinorVersion  0
PE MajorOperatingSystemVersion  6
PE MinorOperatingSystemVersion  0
PE Machine      332
PE TimeDateStamp        Sat Jan 19 05:30:58 2008
```

b) To see the list of processes that were running when the memory was captured, we use the command `volatility3 -f a1memorydump.mem windows.pslist`. The active processes are shown here:



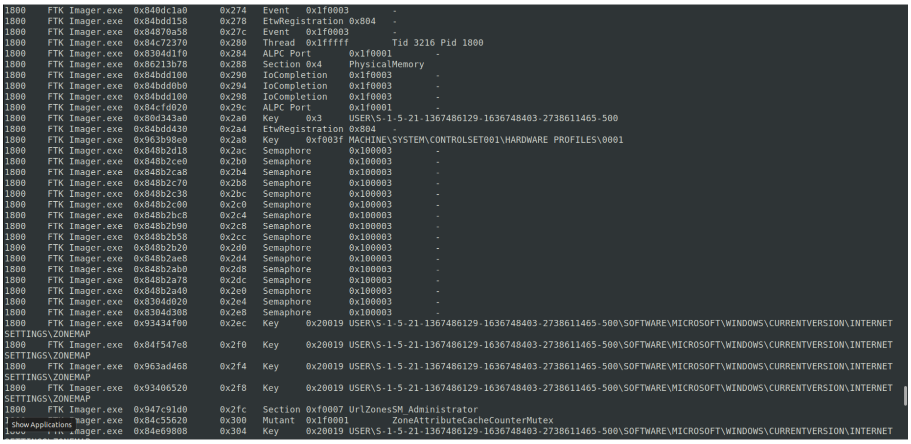
```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.pslist
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

4       0       System  0x82db0910      100     541     N/A     False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
404     4       smss.exe        0x8454c118      4       28      N/A     False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
472     460     csrss.exe       0x84561968      11      466     0       False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
516     508     csrss.exe       0x84450770      10      305     1       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
524     460     wininit.exe     0x84453770      3       98      0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
552     508     winlogon.exe    0x84465770      3       116     1       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
604     524     services.exe    0x83632170      6       250     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
616     524     lsass.exe       0x844bf770      13      610     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
624     524     lsm.exe 0x844c2680      10      208     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
788     604     svchost.exe     0x84866d50      6       298     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
848     604     svchost.exe     0x845f37a8      8       280     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
884     604     svchost.exe     0x848fa118      15      274     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
976     604     svchost.exe     0x84914d90      6       152     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1000    604     svchost.exe     0x8491bd90      45      2072    0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1056    604     SLsvc.exe       0x8492a6d0      4       96      0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1088    604     svchost.exe     0x84937d90      17      567     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1160    604     svchost.exe     0x84941d90      20      265     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1188    604     svchost.exe     0x84945c30      22      596     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1308    604     svchost.exe     0x8496e9f0      17      265     0       False   2014-01-08 02:17:43.000000 UTC  N/A     Disabled
1424    604     spoolsv.exe     0x849c18a8      17      291     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1460    604     armsvc.exe      0x849d7610      2       56      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1480    604     dns.exe 0x849dcd90      10      164     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1508    604     ftpbasicsvr.exe 0x849e1cc0      2       52      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1576    604     svchost.exe     0x849f5888      5       124     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1604    604     svchost.exe     0x849faad8      3       73      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1620    604     snmp.exe        0x849f7380      4       147     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1640    604     vmtoolsd.exe    0x84a0b020      7       273     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1696    604     svchost.exe     0x84a21d90      4       44      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
832     604     TPAutoConnSvc.e 0x83066b68      9       136     0       False   2014-01-08 02:17:52.000000 UTC  N/A     Disabled
1924    604     dllhost.exe     0x84ba8d90      13      240     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
1572    604     msdtc.exe       0x84bb7ca8      11      167     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
2096    1000    taskeng.exe     0x84ba1c30      5       137     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
2352    1000    taskeng.exe     0x84c13020      10      250     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2368    552     userinit.exe    0x84c148e8      0       -       1       False   2014-01-08 02:18:17.000000 UTC  2014-01-08 02:18:43.000000 UTC       Disabled
2392    1160    dwm.exe 0x84c1d020      3       76      1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2480    832     TPAutoConnect.e 0x84c2c898      2       103     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2496    2368    explorer.exe    0x84c2c020      24      689     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2580    2496    vmtoolsd.exe    0x84c5e020      6       9004    1       False   2014-01-08 02:18:18.000000 UTC  N/A     Disabled
2592    2496    AdobeARM.exe    0x84c5f020      6       282     1       False   2014-01-08 02:18:18.000000 UTC  N/A     Disabled
2616    2592    reader_sl.exe   0x84c537b0      0       -       1       False   2014-01-08 02:18:18.000000 UTC  2014-01-08 02:19:20.000000 UTC       Disabled
2720    2444    Oobe.exe        0x84c11298      0       -       1       False   2014-01-08 02:18:22.000000 UTC  2014-01-08 02:55:43.000000 UTC       Disabled
3224    604     svchost.exe     0x84c73b60      9       228     0       False   2014-01-08 02:19:53.000000 UTC  N/A     Disabled
3336    788     iashost.exe     0x84ce3020      2       97      0       False   2014-01-08 02:19:53.000000 UTC  N/A     Disabled
3680    1000    wuauclt.exe     0x84bd8020      2       139     1       False   2014-01-08 02:20:55.000000 UTC  N/A     Disabled
3920    2496    notepad.exe     0x84c64a50      1       51      1       False   2014-01-08 03:19:07.000000 UTC  N/A     Disabled
1800    2496    FTK Imager.exe  0x84cfd958      5       251     1       False   2014-01-08 03:19:32.000000 UTC  N/A     Disabled
1888    2496    iexplore.exe    0x848ab618      14      641     1       False   2014-01-08 03:20:24.000000 UTC  N/A     Disabled
2708    2496    notepad.exe     0x848a1340      1       45      1       False   2014-01-08 17:33:08.000000 UTC  N/A     Disabled
```

c) To dig deeper into a specific process, we can run the command `volatility3 -f a1memorydump.mem windows.psscan --pid <processID>`. We'll check the details for two processes that look suspicious, PID 1800 for FTK Imager.exe and PID 2496 for explorer.exe:



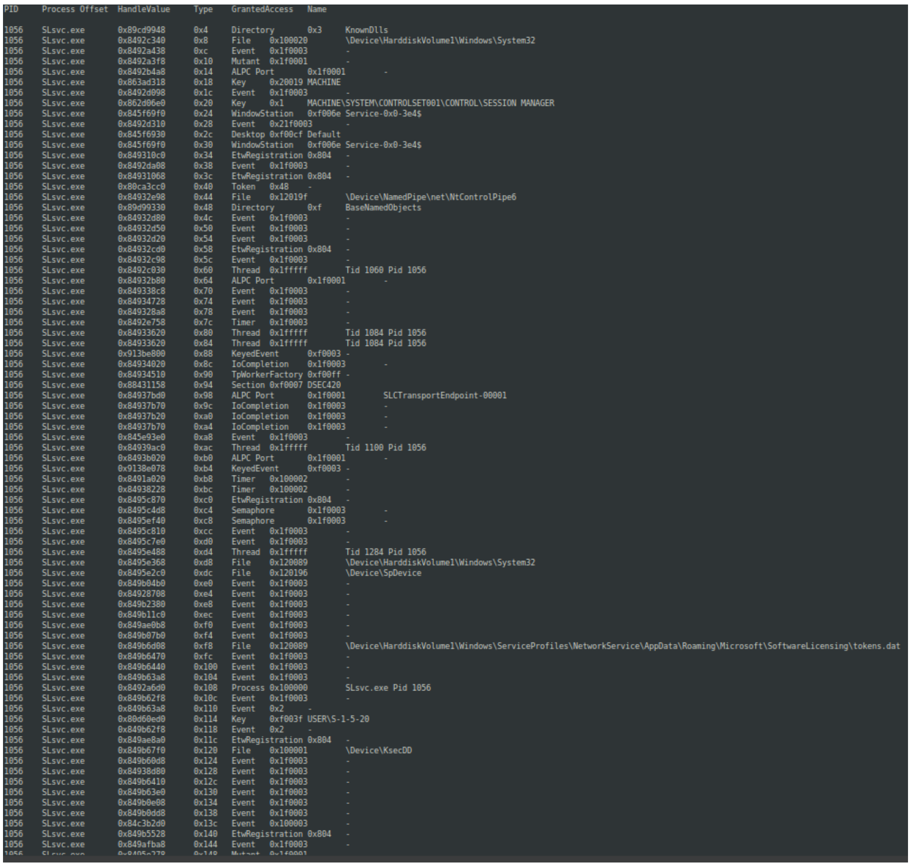
```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.pslist --pid 1800
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

1800    2496    FTK Imager.exe  0x84cfd958      5       251     1       False   2014-01-08 03:19:32.000000 UTC  N/A     Disabled
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.pslist --pid 2496
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

2496    2368    explorer.exe    0x84c2c020      24      689     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
```

d) To view the handles for all processes, which show things like open files and registry keys, we use the command `volatility3 -f a1memorydump.mem windows.handles`. The output is quite long, so just a portion of it is shown below:



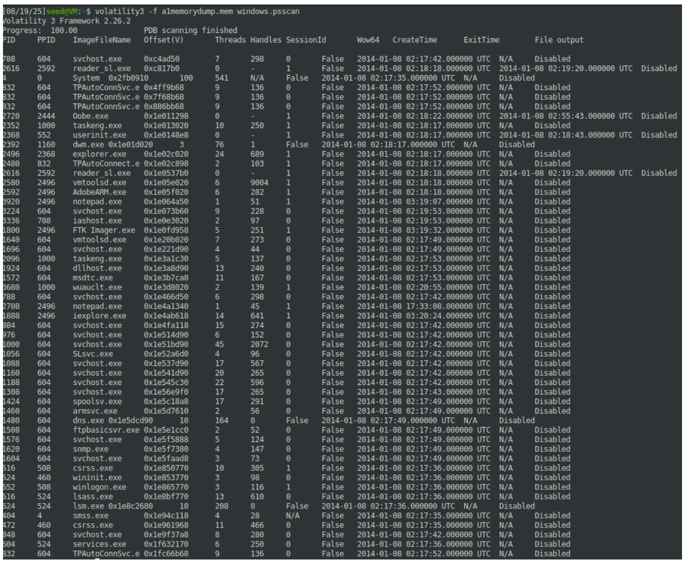
```
1800    FTK Imager.exe  0x840dc1a0      0x274   Event   0x1f0003        -
1800    FTK Imager.exe  0x84bdd158      0x278   EtwRegistration 0x804   -
1800    FTK Imager.exe  0x84870a58      0x27c   Event   0x1f0003        -
1800    FTK Imager.exe  0x84c72370      0x280   Thread  0x1fffff        Tid 3216 Pid 1800
1800    FTK Imager.exe  0x8304d1f0      0x284   ALPC Port       0x1f0001        -
1800    FTK Imager.exe  0x86213b78      0x288   Section 0x4     PhysicalMemory
1800    FTK Imager.exe  0x84bdd100      0x290   IoCompletion    0x1f0003        -
1800    FTK Imager.exe  0x84bdd0b0      0x294   IoCompletion    0x1f0003        -
1800    FTK Imager.exe  0x84bdd100      0x298   IoCompletion    0x1f0003        -
1800    FTK Imager.exe  0x84cfd020      0x29c   ALPC Port       0x1f0001        -
1800    FTK Imager.exe  0x80d343a0      0x2a0   Key     0x3     USER\S-1-5-21-1367486129-1636748403-2738611465-500
1800    FTK Imager.exe  0x84bdd430      0x2a4   EtwRegistration 0x804   -
1800    FTK Imager.exe  0x963b98e0      0x2a8   Key     0xf003f MACHINE\SYSTEM\CONTROLSET001\HARDWARE PROFILES\0001
1800    FTK Imager.exe  0x848b2d18      0x2ac   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2ce0      0x2b0   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2ca8      0x2b4   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2c70      0x2b8   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2c38      0x2bc   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2c00      0x2c0   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2bc8      0x2c4   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2b90      0x2c8   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2b58      0x2cc   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2b20      0x2d0   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2ae8      0x2d4   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2ab0      0x2d8   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2a78      0x2dc   Semaphore       0x100003        -
1800    FTK Imager.exe  0x848b2a40      0x2e0   Semaphore       0x100003        -
1800    FTK Imager.exe  0x8304d020      0x2e4   Semaphore       0x100003        -
1800    FTK Imager.exe  0x8304d308      0x2e8   Semaphore       0x100003        -
1800    FTK Imager.exe  0x93434f00      0x2ec   Key     0x20019 USER\S-1-5-21-1367486129-1636748403-2738611465-500\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\ZONEMAP
1800    FTK Imager.exe  0x84f547e8      0x2f0   Key     0x20019 USER\S-1-5-21-1367486129-1636748403-2738611465-500\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\ZONEMAP
1800    FTK Imager.exe  0x963ad468      0x2f4   Key     0x20019 USER\S-1-5-21-1367486129-1636748403-2738611465-500\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\ZONEMAP
1800    FTK Imager.exe  0x93406520      0x2f8   Key     0x20019 USER\S-1-5-21-1367486129-1636748403-2738611465-500\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
SETTINGS\ZONEMAP
1800    FTK Imager.exe  0x947c91d0      0x2fc   Section 0xf0007 UrlZonesSM_Administrator
1800    FTK Imager.exe  0x84c55620      0x300   Mutant  0x1f0001        ZoneAttributeCacheCounterMutex
1800    FTK Imager.exe  0x84e69808      0x304   Key     0x20019 USER\S-1-5-21-1367486129-1636748403-2738611465-500\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET
```

e) To see the handles for just one process, we can filter by its ID. For process 1056, the command is `volatility3 -f a1memorydump.mem windows.handles --pid 1056`. These handles are shown below:



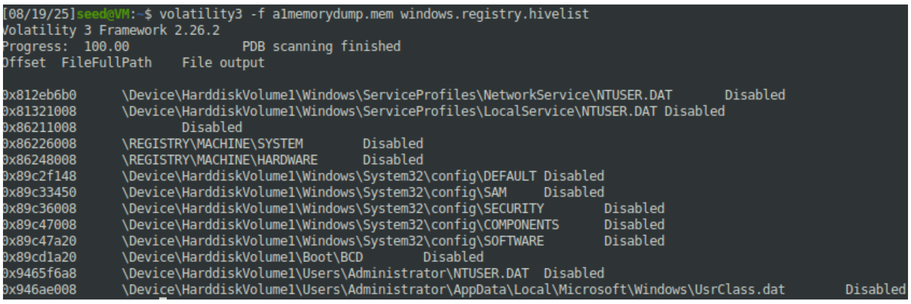
```
PID     Process Offset  HandleValue     Type    GrantedAccess   Name

1056    SLsvc.exe       0x89cd9948      0x4     Directory       0x3     KnownDlls
1056    SLsvc.exe       0x8492c340      0x8     File    0x100020        \Device\HarddiskVolume1\Windows\System32
1056    SLsvc.exe       0x8492a438      0xc     Event   0x1f0003        -
1056    SLsvc.exe       0x8492a3f8      0x10    Mutant  0x1f0001        -
1056    SLsvc.exe       0x8492b4a8      0x14    ALPC Port       0x1f0001        -
1056    SLsvc.exe       0x863ad318      0x18    Key     0x20019 MACHINE
1056    SLsvc.exe       0x8492d098      0x1c    Event   0x1f0003        -
1056    SLsvc.exe       0x862d06e0      0x20    Key     0x1     MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
1056    SLsvc.exe       0x845f69f0      0x24    WindowStation   0xf006e Service-0x0-3e4$
1056    SLsvc.exe       0x8492d310      0x28    Event   0x21f0003       -
1056    SLsvc.exe       0x845f6930      0x2c    Desktop 0xf00cf Default
1056    SLsvc.exe       0x845f69f0      0x30    WindowStation   0xf006e Service-0x0-3e4$
1056    SLsvc.exe       0x849310c0      0x34    EtwRegistration 0x804   -
1056    SLsvc.exe       0x8492da08      0x38    Event   0x1f0003        -
1056    SLsvc.exe       0x84931068      0x3c    EtwRegistration 0x804   -
1056    SLsvc.exe       0x80ca3cc0      0x40    Token   0x48    -
1056    SLsvc.exe       0x84932e98      0x44    File    0x12019f        \Device\NamedPipe\net\NtControlPipe6
1056    SLsvc.exe       0x89d99330      0x48    Directory       0xf     BaseNamedObjects
1056    SLsvc.exe       0x84932d80      0x4c    Event   0x1f0003        -
1056    SLsvc.exe       0x84932d50      0x50    Event   0x1f0003        -
1056    SLsvc.exe       0x84932d20      0x54    Event   0x1f0003        -
1056    SLsvc.exe       0x84932cd0      0x58    EtwRegistration 0x804   -
1056    SLsvc.exe       0x84932c98      0x5c    Event   0x1f0003        -
1056    SLsvc.exe       0x8492c030      0x60    Thread  0x1fffff        Tid 1060 Pid 1056
1056    SLsvc.exe       0x84932b80      0x64    ALPC Port       0x1f0001        -
1056    SLsvc.exe       0x849338c8      0x70    Event   0x1f0003        -
1056    SLsvc.exe       0x84934728      0x74    Event   0x1f0003        -
1056    SLsvc.exe       0x849328a8      0x78    Event   0x1f0003        -
1056    SLsvc.exe       0x8492e758      0x7c    Timer   0x1f0003        -
1056    SLsvc.exe       0x84933620      0x80    Thread  0x1fffff        Tid 1084 Pid 1056
1056    SLsvc.exe       0x84933620      0x84    Thread  0x1fffff        Tid 1084 Pid 1056
1056    SLsvc.exe       0x913be800      0x88    KeyedEvent      0xf0003 -
1056    SLsvc.exe       0x84934020      0x8c    IoCompletion    0x1f0003        -
1056    SLsvc.exe       0x84934510      0x90    TpWorkerFactory 0xf00ff -
1056    SLsvc.exe       0x88431158      0x94    Section 0xf0007 DSEC420
1056    SLsvc.exe       0x84937bd0      0x98    ALPC Port       0x1f0001        SLCTransportEndpoint-00001
1056    SLsvc.exe       0x84937b70      0x9c    IoCompletion    0x1f0003        -
1056    SLsvc.exe       0x84937b20      0xa0    IoCompletion    0x1f0003        -
1056    SLsvc.exe       0x84937b70      0xa4    IoCompletion    0x1f0003        -
1056    SLsvc.exe       0x845e93e0      0xa8    Event   0x1f0003        -
1056    SLsvc.exe       0x84939ac0      0xac    Thread  0x1fffff        Tid 1100 Pid 1056
1056    SLsvc.exe       0x8493b020      0xb0    ALPC Port       0x1f0001        -
1056    SLsvc.exe       0x9138e078      0xb4    KeyedEvent      0xf0003 -
1056    SLsvc.exe       0x8491a020      0xb8    Timer   0x100002        -
1056    SLsvc.exe       0x84938228      0xbc    Timer   0x100002        -
1056    SLsvc.exe       0x8495c870      0xc0    EtwRegistration 0x804   -
1056    SLsvc.exe       0x8495c4d8      0xc4    Semaphore       0x1f0003        -
1056    SLsvc.exe       0x8495ef40      0xc8    Semaphore       0x1f0003        -
1056    SLsvc.exe       0x8495c810      0xcc    Event   0x1f0003        -
1056    SLsvc.exe       0x8495c7e0      0xd0    Event   0x1f0003        -
1056    SLsvc.exe       0x8495e488      0xd4    Thread  0x1fffff        Tid 1284 Pid 1056
1056    SLsvc.exe       0x8495e368      0xd8    File    0x120089        \Device\HarddiskVolume1\Windows\System32
1056    SLsvc.exe       0x8495e2c0      0xdc    File    0x120196        \Device\SpDevice
1056    SLsvc.exe       0x849b04b0      0xe0    Event   0x1f0003        -
1056    SLsvc.exe       0x84928708      0xe4    Event   0x1f0003        -
1056    SLsvc.exe       0x849b2380      0xe8    Event   0x1f0003        -
1056    SLsvc.exe       0x849b11c0      0xec    Event   0x1f0003        -
1056    SLsvc.exe       0x849ae0b8      0xf0    Event   0x1f0003        -
1056    SLsvc.exe       0x849b07b0      0xf4    Event   0x1f0003        -
1056    SLsvc.exe       0x849b6d08      0xf8    File    0x120089        \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Microsoft\SoftwareLicensing\tokens.dat
1056    SLsvc.exe       0x849b6470      0xfc    Event   0x1f0003        -
1056    SLsvc.exe       0x849b6440      0x100   Event   0x1f0003        -
1056    SLsvc.exe       0x849b63a8      0x104   Event   0x1f0003        -
1056    SLsvc.exe       0x8492a6d0      0x108   Process 0x100000        SLsvc.exe Pid 1056
1056    SLsvc.exe       0x849b62f8      0x10c   Event   0x1f0003        -
1056    SLsvc.exe       0x849b63a8      0x110   Event   0x2     -
1056    SLsvc.exe       0x80d60ed0      0x114   Key     0xf003f USER\S-1-5-20
1056    SLsvc.exe       0x849b62f8      0x118   Event   0x2     -
1056    SLsvc.exe       0x849ae8a0      0x11c   EtwRegistration 0x804   -
1056    SLsvc.exe       0x849b67f0      0x120   File    0x100001        \Device\KsecDD
1056    SLsvc.exe       0x849b60d8      0x124   Event   0x1f0003        -
1056    SLsvc.exe       0x84938d80      0x128   Event   0x1f0003        -
1056    SLsvc.exe       0x849b6410      0x12c   Event   0x1f0003        -
1056    SLsvc.exe       0x849b63e0      0x130   Event   0x1f0003        -
1056    SLsvc.exe       0x849b0e08      0x134   Event   0x1f0003        -
1056    SLsvc.exe       0x849b0dd8      0x138   Event   0x1f0003        -
1056    SLsvc.exe       0x84c3b2d0      0x13c   Event   0x100003        -
1056    SLsvc.exe       0x849b5528      0x140   EtwRegistration 0x804   -
1056    SLsvc.exe       0x849afba8      0x144   Event   0x1f0003        -
1056    SLsvc.exe       0x8495e278      0x148   Mutant  0x1f0001
```

f) To find processes that might be hidden, we use the command `volatility3 -f a1memorydump.mem windows.psscan`. The list of any found hidden processes is shown below:



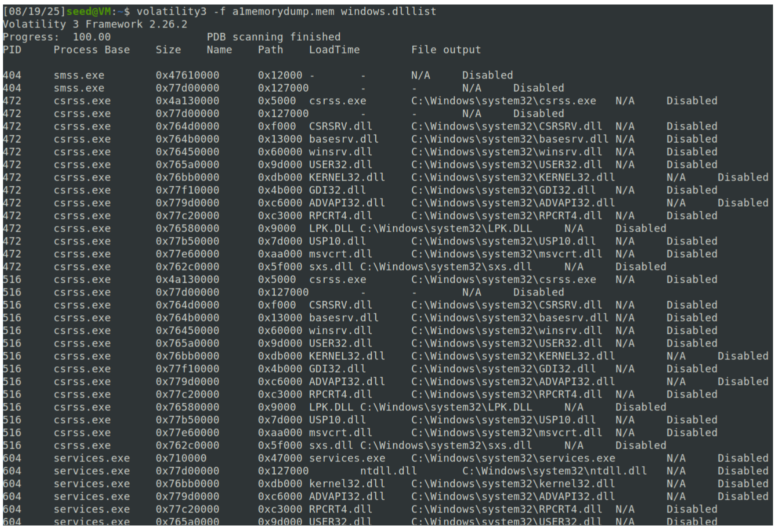
```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.psscan
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File output

788     604     svchost.exe     0xc4ad50        7       298     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
2616    2592    reader_sl.exe   0xc817b0        0       -       1       False   2014-01-08 02:18:18.000000 UTC  2014-01-08 02:19:20.000000 UTC  Disabled
4       0       System  0x2fb0910       100     541     N/A     False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
832     604     TPAutoConnSvc.e 0x4ff9b68       9       136     0       False   2014-01-08 02:17:52.000000 UTC  N/A     Disabled
832     604     TPAutoConnSvc.e 0x7f68b68       9       136     0       False   2014-01-08 02:17:52.000000 UTC  N/A     Disabled
832     604     TPAutoConnSvc.e 0x886bb68       9       136     0       False   2014-01-08 02:17:52.000000 UTC  N/A     Disabled
2720    2444    Oobe.exe        0x1e011298      0       -       1       False   2014-01-08 02:18:22.000000 UTC  2014-01-08 02:55:43.000000 UTC  Disabled
2352    1000    taskeng.exe     0x1e013020      10      250     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2368    552     userinit.exe    0x1e0148e8      0       -       1       False   2014-01-08 02:18:17.000000 UTC  2014-01-08 02:18:43.000000 UTC  Disabled
2392    1160    dwm.exe 0x1e01d020      3       76      1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2496    2368    explorer.exe    0x1e02c020      24      689     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2480    832     TPAutoConnect.e 0x1e02c898      2       103     1       False   2014-01-08 02:18:17.000000 UTC  N/A     Disabled
2616    2592    reader_sl.exe   0x1e0537b0      0       -       1       False   2014-01-08 02:18:18.000000 UTC  2014-01-08 02:19:20.000000 UTC  Disabled
2580    2496    vmtoolsd.exe    0x1e05e020      6       9004    1       False   2014-01-08 02:18:18.000000 UTC  N/A     Disabled
2592    2496    AdobeARM.exe    0x1e05f020      6       282     1       False   2014-01-08 02:18:18.000000 UTC  N/A     Disabled
3920    2496    notepad.exe     0x1e064a50      1       51      1       False   2014-01-08 03:19:07.000000 UTC  N/A     Disabled
3224    604     svchost.exe     0x1e073b60      9       228     0       False   2014-01-08 02:19:53.000000 UTC  N/A     Disabled
3336    788     iashost.exe     0x1e0e3020      2       97      0       False   2014-01-08 02:19:53.000000 UTC  N/A     Disabled
1800    2496    FTK Imager.exe  0x1e0fd958      5       251     1       False   2014-01-08 03:19:32.000000 UTC  N/A     Disabled
1640    604     vmtoolsd.exe    0x1e20b020      7       273     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1696    604     svchost.exe     0x1e221d90      4       44      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
2096    1000    taskeng.exe     0x1e3a1c30      5       137     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
1924    604     dllhost.exe     0x1e3a8d90      13      240     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
1572    604     msdtc.exe       0x1e3b7ca8      11      167     0       False   2014-01-08 02:17:53.000000 UTC  N/A     Disabled
3680    1000    wuauclt.exe     0x1e3d8020      2       139     1       False   2014-01-08 02:20:55.000000 UTC  N/A     Disabled
788     604     svchost.exe     0x1e466d50      6       298     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
2708    2496    notepad.exe     0x1e4a1340      1       45      1       False   2014-01-08 17:33:08.000000 UTC  N/A     Disabled
1888    2496    iexplore.exe    0x1e4ab618      14      641     1       False   2014-01-08 03:20:24.000000 UTC  N/A     Disabled
884     604     svchost.exe     0x1e4fa118      15      274     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
976     604     svchost.exe     0x1e514d90      6       152     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1000    604     svchost.exe     0x1e51bd90      45      2072    0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1056    604     SLsvc.exe       0x1e52a6d0      4       96      0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1088    604     svchost.exe     0x1e537d90      17      567     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1160    604     svchost.exe     0x1e541d90      20      265     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1188    604     svchost.exe     0x1e545c30      22      596     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
1308    604     svchost.exe     0x1e56e9f0      17      265     0       False   2014-01-08 02:17:43.000000 UTC  N/A     Disabled
1424    604     spoolsv.exe     0x1e5c18a8      17      291     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1460    604     armsvc.exe      0x1e5d7610      2       56      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1480    604     dns.exe 0x1e5dcd90      10      164     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1508    604     ftpbasicsvr.exe 0x1e5e1cc0      2       52      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1576    604     svchost.exe     0x1e5f5888      5       124     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1620    604     snmp.exe        0x1e5f7380      4       147     0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
1604    604     svchost.exe     0x1e5faad8      3       73      0       False   2014-01-08 02:17:49.000000 UTC  N/A     Disabled
516     508     csrss.exe       0x1e850770      10      305     1       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
524     460     wininit.exe     0x1e853770      3       98      0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
552     508     winlogon.exe    0x1e865770      3       116     1       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
616     524     lsass.exe       0x1e8bf770      13      610     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
624     524     lsm.exe 0x1e8c2680      10      208     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
404     4       smss.exe        0x1e94c118      4       28      N/A     False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
472     460     csrss.exe       0x1e961968      11      466     0       False   2014-01-08 02:17:35.000000 UTC  N/A     Disabled
848     604     svchost.exe     0x1e9f37a8      8       280     0       False   2014-01-08 02:17:42.000000 UTC  N/A     Disabled
604     524     services.exe    0x1f632170      6       250     0       False   2014-01-08 02:17:36.000000 UTC  N/A     Disabled
832     604     TPAutoConnSvc.e 0x1fc66b68      9       136     0       False   2014-01-08 02:17:52.000000 UTC  N/A     Disabled
```

g) To find the list of registry hives loaded in memory, we can use the command `volatility3 -f a1memorydump.mem windows.registry.hivelist`. The output of this command is shown below:



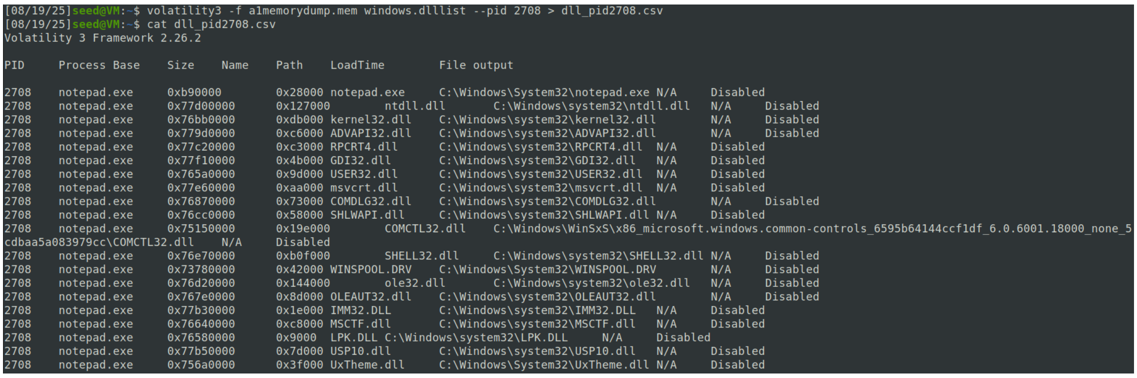
```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.registry.hivelist
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
Offset  FileFullPath    File output

0x812eb6b0      \Device\HarddiskVolume1\Windows\ServiceProfiles\NetworkService\NTUSER.DAT       Disabled
0x81321008      \Device\HarddiskVolume1\Windows\ServiceProfiles\LocalService\NTUSER.DAT Disabled
0x86211008              Disabled
0x86226008      \REGISTRY\MACHINE\SYSTEM        Disabled
0x86248008      \REGISTRY\MACHINE\HARDWARE      Disabled
0x89c2f148      \Device\HarddiskVolume1\Windows\System32\config\DEFAULT Disabled
0x89c33450      \Device\HarddiskVolume1\Windows\System32\config\SAM     Disabled
0x89c36008      \Device\HarddiskVolume1\Windows\System32\config\SECURITY        Disabled
0x89c47008      \Device\HarddiskVolume1\Windows\System32\config\COMPONENTS      Disabled
0x89c47a20      \Device\HarddiskVolume1\Windows\System32\config\SOFTWARE        Disabled
0x89cd1a20      \Device\HarddiskVolume1\Boot\BCD        Disabled
0x9465f6a8      \Device\HarddiskVolume1\Users\Administrator\NTUSER.DAT  Disabled
0x946ae008      \Device\HarddiskVolume1\Users\Administrator\AppData\Local\Microsoft\Windows\UsrClass.dat        Disabled
```

h) To find the list of Dynamic Link Libraries (DLLs) being used by the running processes, we use the command `volatility3 -f a1memorydump.mem windows.dlllist`. The full output is quite lengthy, so only a small part is shown in the screenshot below:



```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.dlllist
Volatility 3 Framework 2.26.2
Progress:  100.00               PDB scanning finished
PID     Process Base    Size    Name    Path    LoadTime        File output

404     smss.exe        0x47610000      0x12000 -       -       N/A     Disabled
404     smss.exe        0x77d00000      0x127000        -       -       N/A     Disabled
472     csrss.exe       0x4a130000      0x5000  csrss.exe       C:\Windows\system32\csrss.exe   N/A     Disabled
472     csrss.exe       0x77d00000      0x127000        -       -       N/A     Disabled
472     csrss.exe       0x764d0000      0xf000  CSRSRV.dll      C:\Windows\system32\CSRSRV.dll  N/A     Disabled
472     csrss.exe       0x764b0000      0x13000 basesrv.dll     C:\Windows\system32\basesrv.dll N/A     Disabled
472     csrss.exe       0x76450000      0x60000 winsrv.dll      C:\Windows\system32\winsrv.dll  N/A     Disabled
472     csrss.exe       0x765a0000      0x9d000 USER32.dll      C:\Windows\system32\USER32.dll  N/A     Disabled
472     csrss.exe       0x76bb0000      0xdb000 KERNEL32.dll    C:\Windows\system32\KERNEL32.dll        N/A     Disabled
472     csrss.exe       0x77f10000      0x4b000 GDI32.dll       C:\Windows\system32\GDI32.dll   N/A     Disabled
472     csrss.exe       0x779d0000      0xc6000 ADVAPI32.dll    C:\Windows\system32\ADVAPI32.dll        N/A     Disabled
472     csrss.exe       0x77c20000      0xc3000 RPCRT4.dll      C:\Windows\system32\RPCRT4.dll  N/A     Disabled
472     csrss.exe       0x76580000      0x9000  LPK.DLL C:\Windows\system32\LPK.DLL     N/A     Disabled
472     csrss.exe       0x77b50000      0x7d000 USP10.dll       C:\Windows\system32\USP10.dll   N/A     Disabled
472     csrss.exe       0x77e60000      0xaa000 msvcrt.dll      C:\Windows\system32\msvcrt.dll  N/A     Disabled
472     csrss.exe       0x762c0000      0x5f000 sxs.dll C:\Windows\system32\sxs.dll     N/A     Disabled
516     csrss.exe       0x4a130000      0x5000  csrss.exe       C:\Windows\system32\csrss.exe   N/A     Disabled
516     csrss.exe       0x77d00000      0x127000        -       -       N/A     Disabled
516     csrss.exe       0x764d0000      0xf000  CSRSRV.dll      C:\Windows\system32\CSRSRV.dll  N/A     Disabled
516     csrss.exe       0x764b0000      0x13000 basesrv.dll     C:\Windows\system32\basesrv.dll N/A     Disabled
516     csrss.exe       0x76450000      0x60000 winsrv.dll      C:\Windows\system32\winsrv.dll  N/A     Disabled
516     csrss.exe       0x765a0000      0x9d000 USER32.dll      C:\Windows\system32\USER32.dll  N/A     Disabled
516     csrss.exe       0x76bb0000      0xdb000 KERNEL32.dll    C:\Windows\system32\KERNEL32.dll        N/A     Disabled
516     csrss.exe       0x77f10000      0x4b000 GDI32.dll       C:\Windows\system32\GDI32.dll   N/A     Disabled
516     csrss.exe       0x779d0000      0xc6000 ADVAPI32.dll    C:\Windows\system32\ADVAPI32.dll        N/A     Disabled
516     csrss.exe       0x77c20000      0xc3000 RPCRT4.dll      C:\Windows\system32\RPCRT4.dll  N/A     Disabled
516     csrss.exe       0x76580000      0x9000  LPK.DLL C:\Windows\system32\LPK.DLL     N/A     Disabled
516     csrss.exe       0x77b50000      0x7d000 USP10.dll       C:\Windows\system32\USP10.dll   N/A     Disabled
516     csrss.exe       0x77e60000      0xaa000 msvcrt.dll      C:\Windows\system32\msvcrt.dll  N/A     Disabled
516     csrss.exe       0x762c0000      0x5f000 sxs.dll C:\Windows\system32\sxs.dll     N/A     Disabled
604     services.exe    0x710000        0x47000 services.exe    C:\Windows\system32\services.exe        N/A     Disabled
604     services.exe    0x77d00000      0x127000        ntdll.dll       C:\Windows\system32\ntdll.dll   N/A     Disabled
604     services.exe    0x76bb0000      0xdb000 kernel32.dll    C:\Windows\system32\kernel32.dll        N/A     Disabled
604     services.exe    0x779d0000      0xc6000 ADVAPI32.dll    C:\Windows\system32\ADVAPI32.dll        N/A     Disabled
604     services.exe    0x77c20000      0xc3000 RPCRT4.dll      C:\Windows\system32\RPCRT4.dll  N/A     Disabled
604     services.exe    0x765a0000      0x9d000 USER32.dll      C:\Windows\system32\USER32.dll  N/A     Disabled
```

i) To get a list of DLLs used by just process 2708 and save it to a file, we can redirect the output to a CSV. The command is `volatility3 -f a1memorydump.mem windows.dlllist --pid 2708 > dll_pid2708.csv`. We can then view this file's contents with `cat dll_pid2708.csv`. The screenshot below shows these commands and their output:



```
[08/19/25]seed@VM:~$ volatility3 -f a1memorydump.mem windows.dlllist --pid 2708 > dll_pid2708.csv
[08/19/25]seed@VM:~$ cat dll_pid2708.csv
Volatility 3 Framework 2.26.2

PID     Process Base    Size    Name    Path    LoadTime        File output

2708    notepad.exe     0xb90000        0x28000 notepad.exe     C:\Windows\System32\notepad.exe N/A     Disabled
2708    notepad.exe     0x77d00000      0x127000        ntdll.dll       C:\Windows\system32\ntdll.dll   N/A     Disabled
2708    notepad.exe     0x76bb0000      0xdb000 kernel32.dll    C:\Windows\system32\kernel32.dll        N/A     Disabled
2708    notepad.exe     0x779d0000      0xc6000 ADVAPI32.dll    C:\Windows\system32\ADVAPI32.dll        N/A     Disabled
2708    notepad.exe     0x77c20000      0xc3000 RPCRT4.dll      C:\Windows\system32\RPCRT4.dll  N/A     Disabled
2708    notepad.exe     0x77f10000      0x4b000 GDI32.dll       C:\Windows\system32\GDI32.dll   N/A     Disabled
2708    notepad.exe     0x765a0000      0x9d000 USER32.dll      C:\Windows\system32\USER32.dll  N/A     Disabled
2708    notepad.exe     0x77e60000      0xaa000 msvcrt.dll      C:\Windows\system32\msvcrt.dll  N/A     Disabled
2708    notepad.exe     0x76870000      0x73000 COMDLG32.dll    C:\Windows\system32\COMDLG32.dll        N/A     Disabled
2708    notepad.exe     0x76cc0000      0x58000 SHLWAPI.dll     C:\Windows\system32\SHLWAPI.dll N/A     Disabled
2708    notepad.exe     0x75150000      0x19e000        COMCTL32.dll    C:\Windows\WinSxS\x86_microsoft.windows.common-controls_6595b64144ccf1df_6.0.6001.18000_none_5
cdbaa5a083979cc\COMCTL32.dll    N/A     Disabled
2708    notepad.exe     0x76e70000      0xb0f000        SHELL32.dll     C:\Windows\system32\SHELL32.dll N/A     Disabled
2708    notepad.exe     0x73780000      0x42000 WINSPOOL.DRV    C:\Windows\System32\WINSPOOL.DRV        N/A     Disabled
2708    notepad.exe     0x76d20000      0x144000        ole32.dll       C:\Windows\system32\ole32.dll   N/A     Disabled
2708    notepad.exe     0x767e0000      0x8d000 OLEAUT32.dll    C:\Windows\system32\OLEAUT32.dll        N/A     Disabled
2708    notepad.exe     0x77b30000      0x1e000 IMM32.DLL       C:\Windows\system32\IMM32.DLL   N/A     Disabled
2708    notepad.exe     0x76640000      0xc8000 MSCTF.dll       C:\Windows\system32\MSCTF.dll   N/A     Disabled
2708    notepad.exe     0x76580000      0x9000  LPK.DLL C:\Windows\system32\LPK.DLL     N/A     Disabled
2708    notepad.exe     0x77b50000      0x7d000 USP10.dll       C:\Windows\system32\USP10.dll   N/A     Disabled
2708    notepad.exe     0x756a0000      0x3f000 UxTheme.dll     C:\Windows\System32\UxTheme.dll N/A     Disabled
```

References

[1] CacheSleuth. *ROT (Caesar) Cipher*. Accessed: August 7, 2025. 2024. URL: `https://www.cachesleuth.com/rot.html`.

[2] David Conger. *Perform Linux memory forensics with this open source tool*. Accessed: August 7, 2025. Apr. 2021. URL: `https://opensource.com/article/21/4/linux-memory-forensics`.

[3] H-11 Digital Forensics. *Cellebrite Digital Collector*. Accessed: August 7, 2025. 2024. URL: `https://h11dfs.com/macquisition_cellebrite/`.

[4] Info-Savvy. *Data Acquisition and Duplication Tools: Software*. Accessed: August 7, 2025. 2024. URL: `https://info-savvy.com/data-acquisition-and-duplication-tools-software/`.

[5] Infosec Institute. *Kali Linux: Top 5 tools for digital forensics*. Accessed: August 7, 2025. Jan. 2025. URL: `https://www.infosecinstitute.com/resources/digital-forensics/kali-linux-top-5-tools-for-digital-forensics/`.

[6] Magnet Forensics. *Magnet RAM Capture*. Accessed: August 7, 2025. 2024. URL: `https://www.magnetforensics.com/resources/magnet-ram-capture/`.

[7] Bill Nelson, Amelia Phillips, and Christopher Steuart. *Guide to Computer Forensics and Investigations*. Fourth. Referenced via UTC document server. Cengage Learning, 2009.

[8] Ponder The Bits. *OSX (Mac) Memory Acquisition and Analysis Using OSXpmem and Volatility*. Accessed: August 7, 2025. Feb. 2017. URL: `https://ponderthebits.com/2017/02/osx-mac-memory-acquisition-and-analysis-using-osxpmem-and-volatility/`.

[9] James Stanley. *Image Steganography*. Accessed: August 7, 2025. 2016. URL: `https://incoherency.co.uk/image-steganography/`.

[10] stylesuxx. *Steganography Online*. Accessed: August 7, 2025. 2014. URL: `https://stylesuxx.github.io/steganography/`.

[11] SUMURI. *Open Source Tools & Mac Forensics*. Accessed: August 7, 2025. Aug. 2023. URL: `https://sumuri.com/open-source-tools-mac-forensics/`.

[12] The Rekall Team. *The Pmem Acquisition Tools*. Accessed: August 7, 2025. 2016. URL: `http://www.rekall-forensic.com/docs/Tools/pmem.html`.