

Week 2

Title:

Recover Deleted Evidence and Uncover Hidden Metadata

Saad Naveed

Cybersecurity & Digital Forensics

DF Internee

Company: Cyborts

Date: 8th July 2025

Submission: 15th July 2025

Objective

Perform forensic analysis on a previously captured disk image to identify, recover, and document deleted, hidden, or suspicious files — including embedded metadata and timestamps.

Step 1: Mount Forensic Image in Autopsy

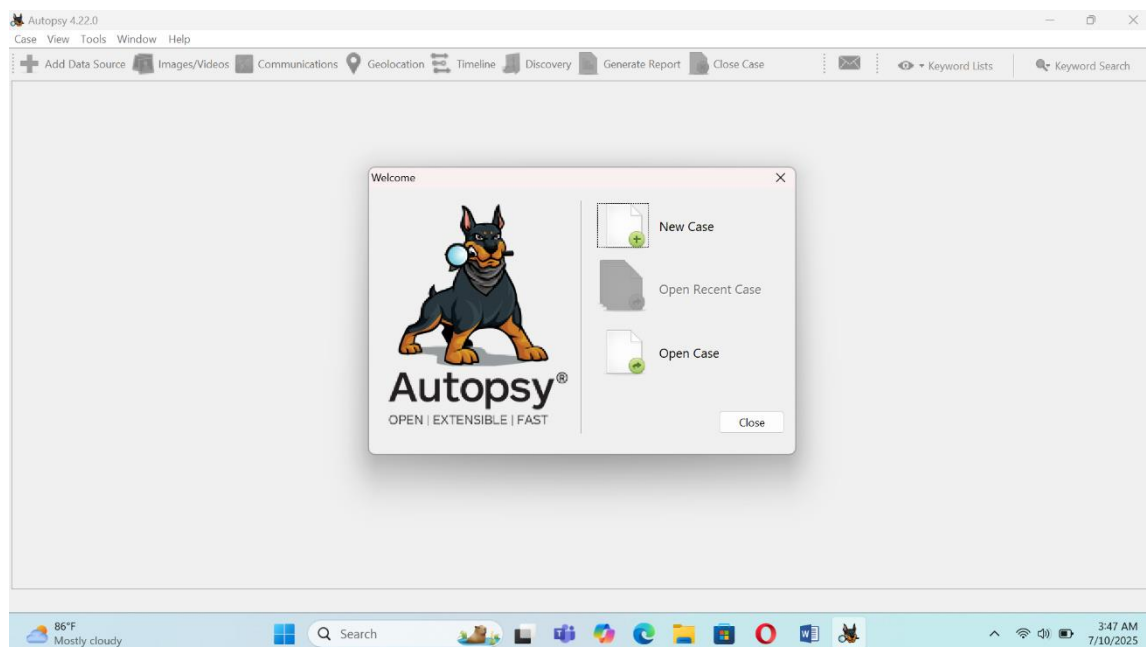
- **Open Autopsy**

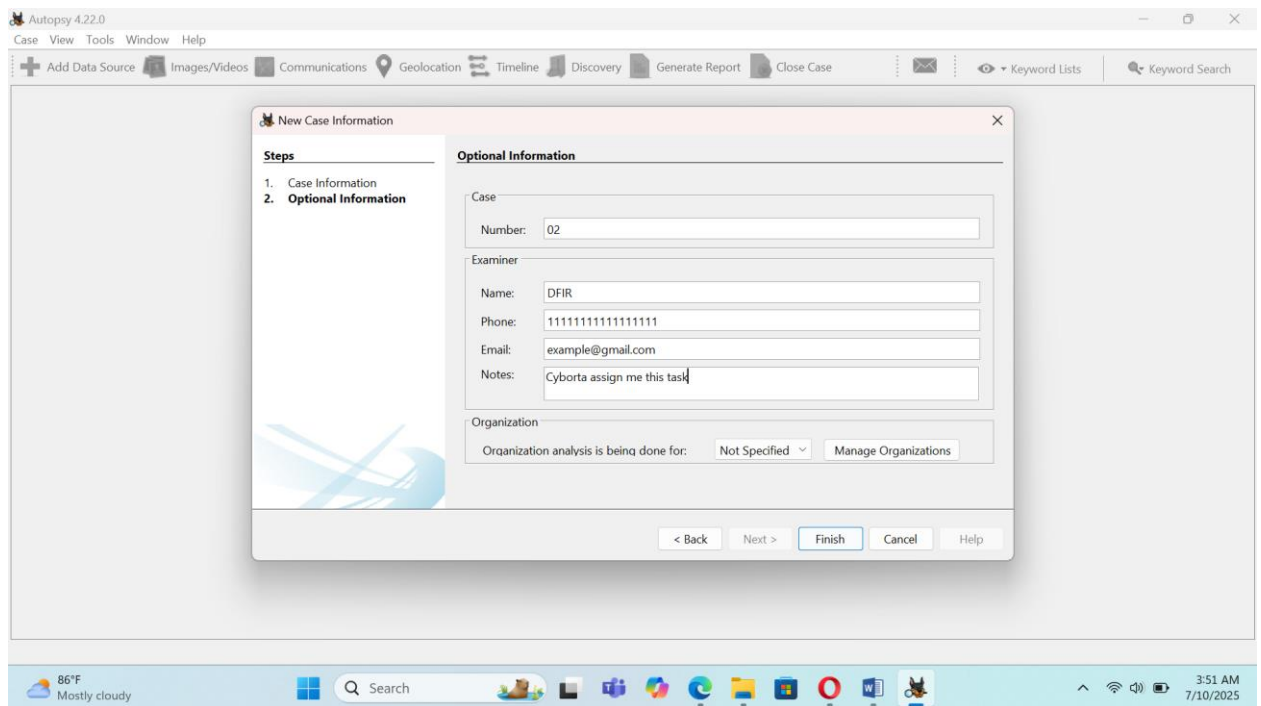
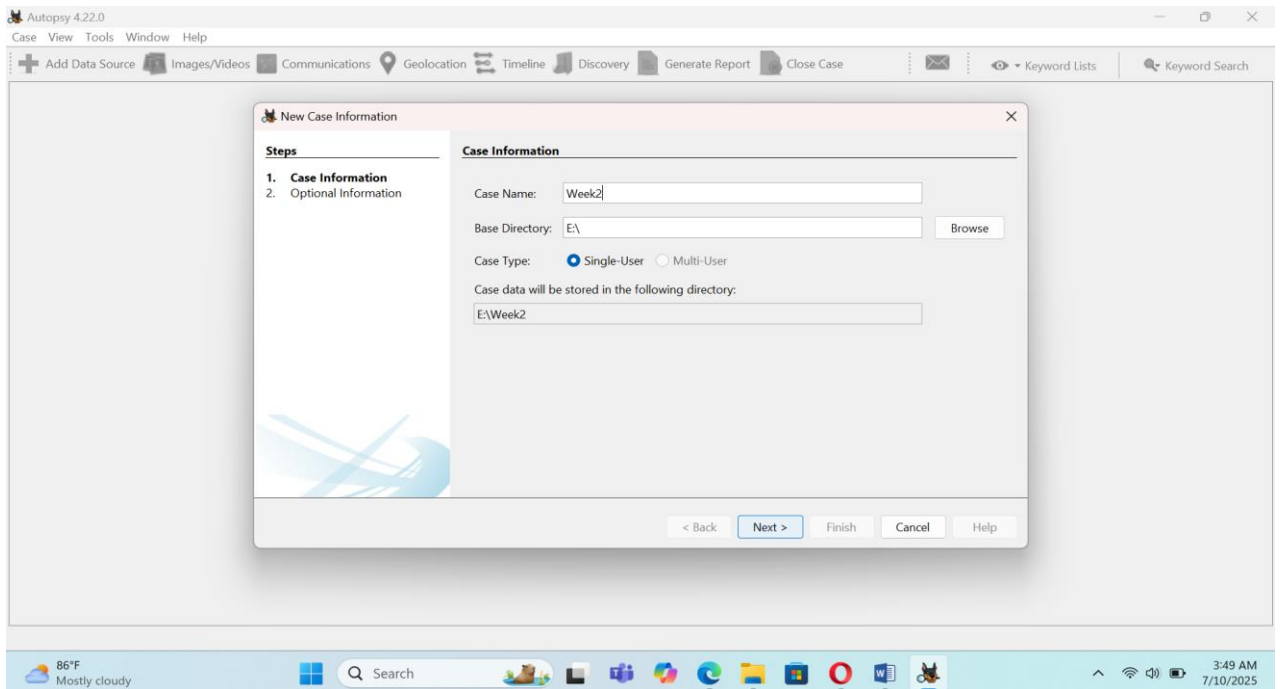
Create a new case:

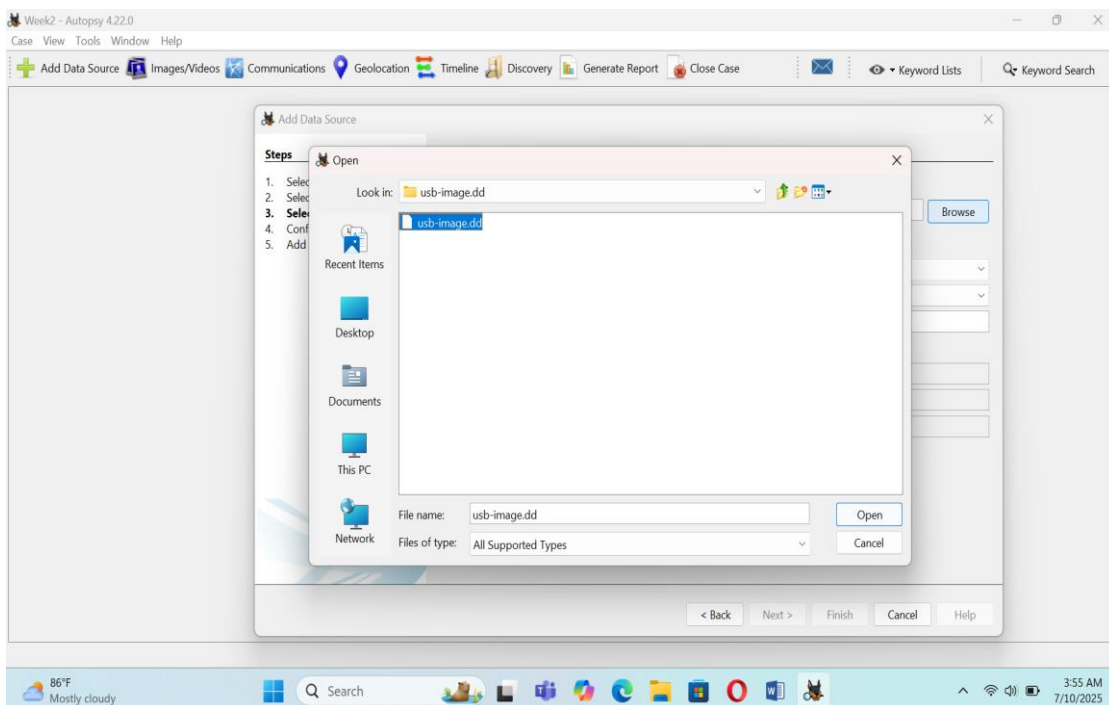
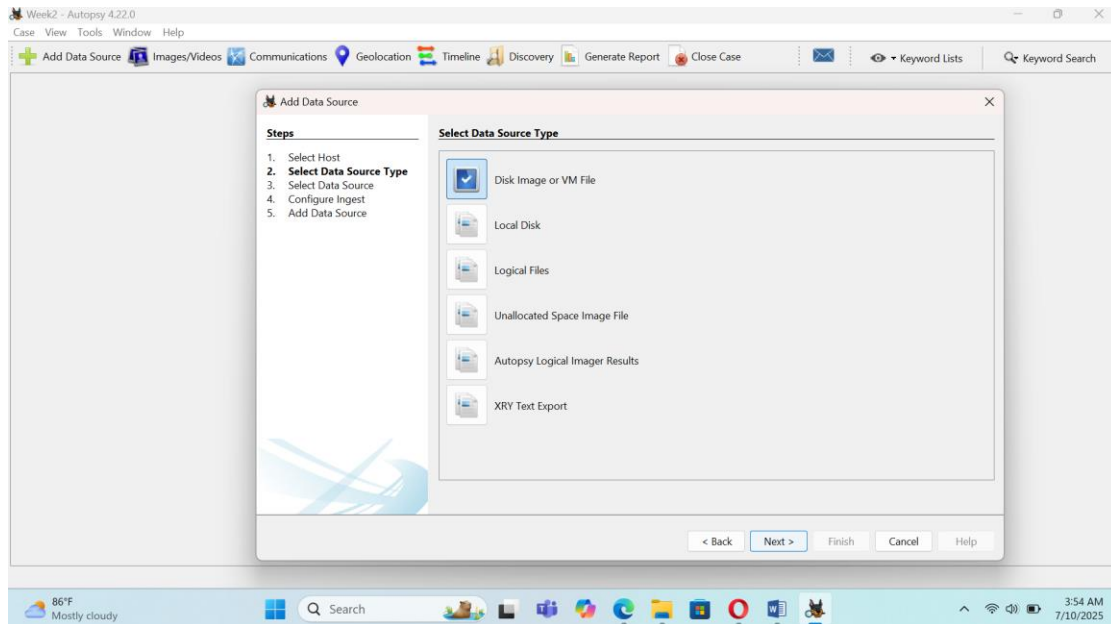
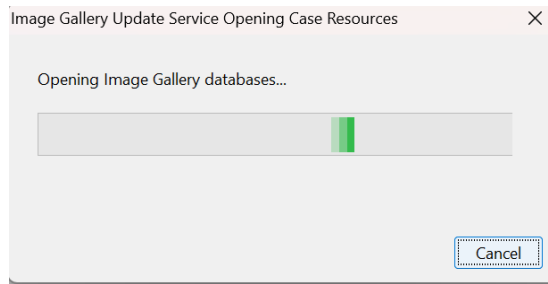
- Case Name: Week2
- Base Directory: I choose folder

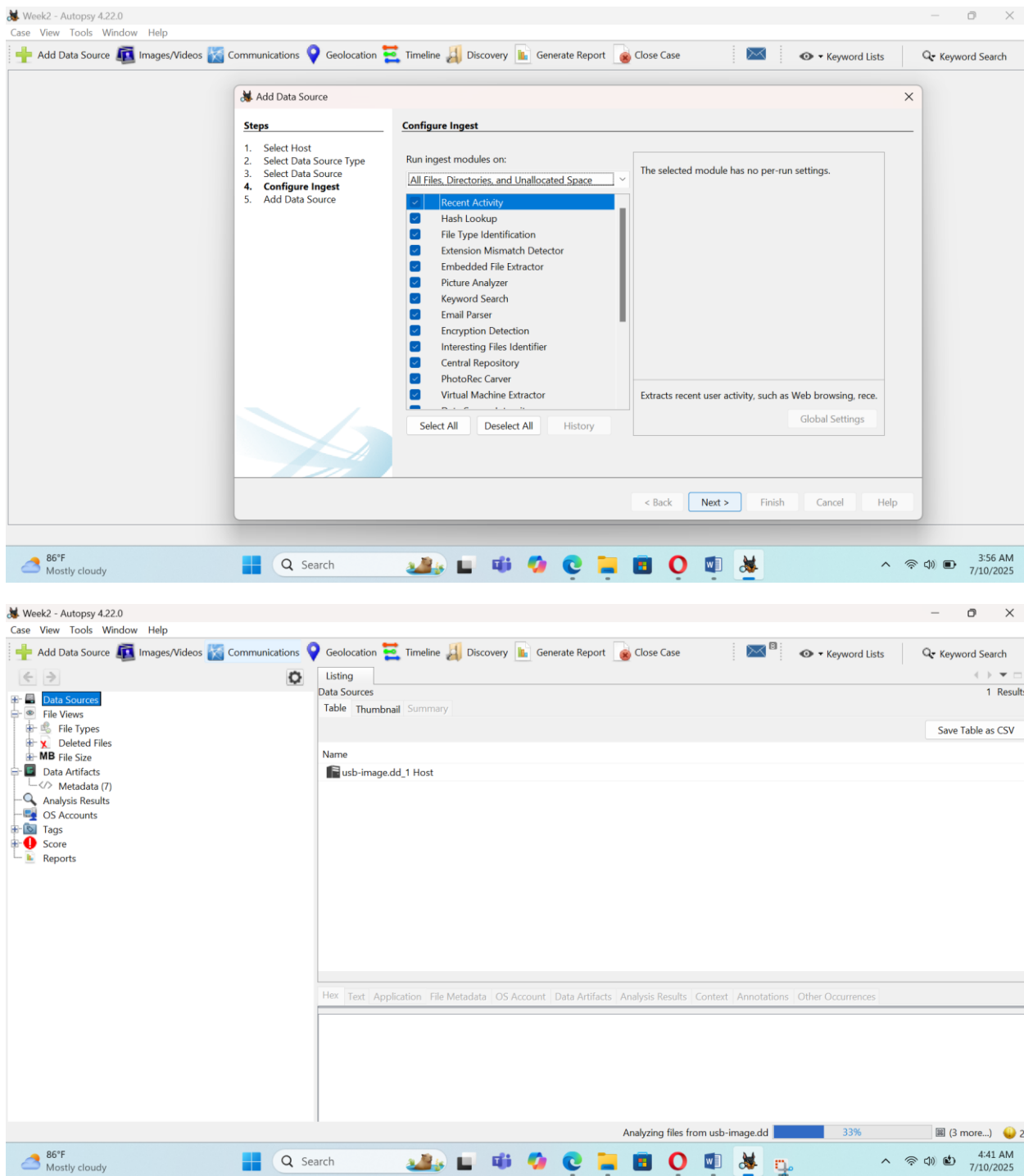
Add Data Source:

- I Select “Disk Image ”
- Browse and choose my Week 1 image (usb_image.dd)
- Check **Ingest Modules** (File Type Identification, Recent Activity, etc.)
- Keep image in **read-only** (Autopsy does this by default)









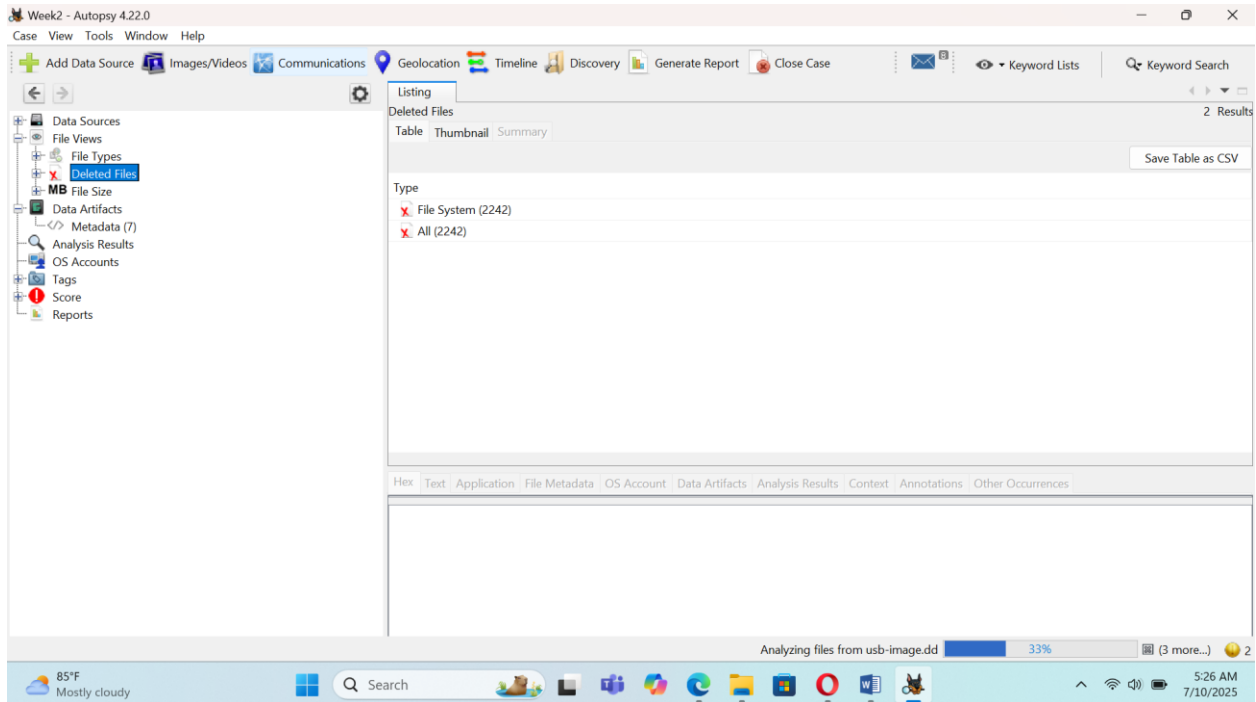
Step 2: Recover Deleted Files

- I select **“Deleted Files”** node in left panel

Autopsy shows recoverable deleted files

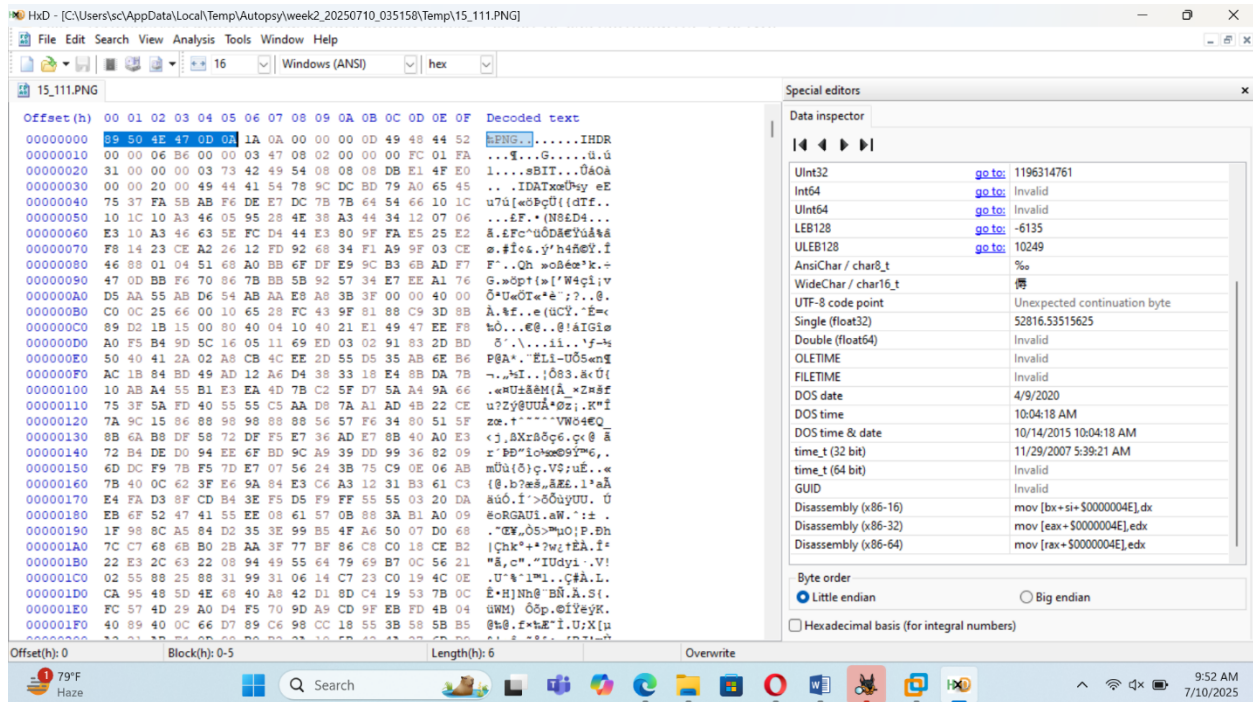
Recover **at least 5** deleted files:

- Right-click → Extract/Export

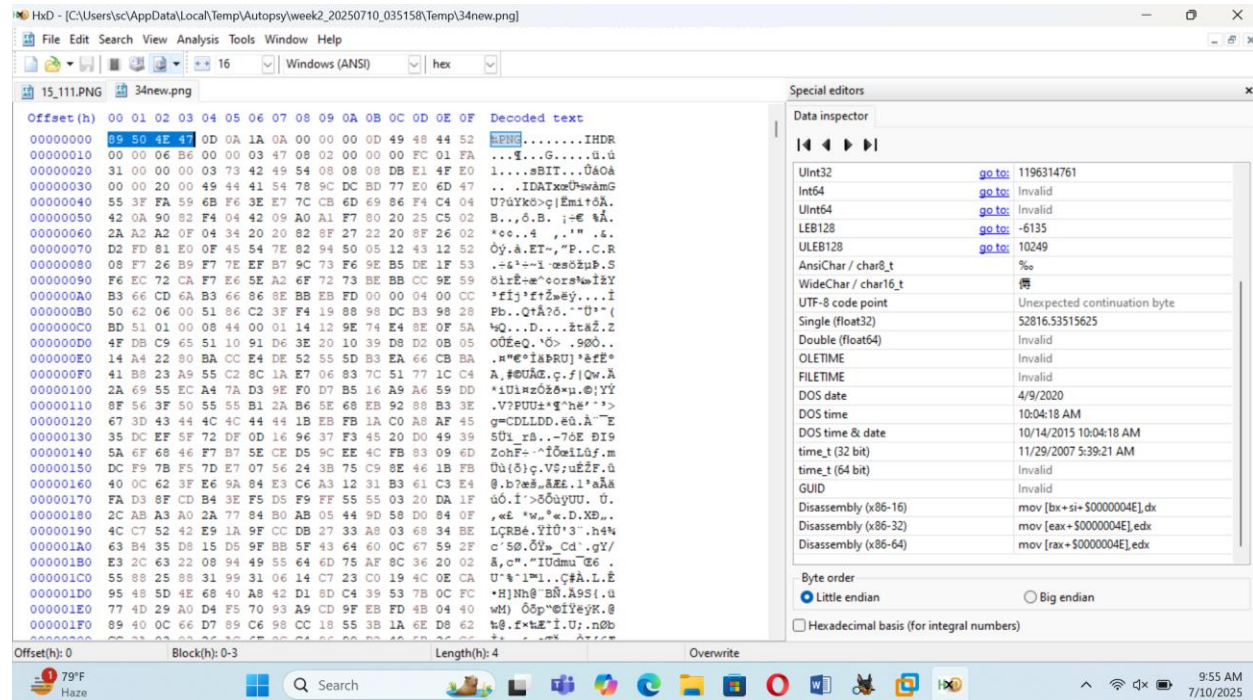


File No.	File Name	File Type	Path	Notes
1	new.png	.png	/img_usb-image.dd/.Trash-1000/files/new.png	Recovered
2	Title01.pdf	.pdf	/img_usb-image.dd/.Trash-1000/files/Title01.pdf	Recovered
3	111.png	.png	/img_usb-image.dd/.Trash-1000/files/111.png	Recovered
4	1.PNG	.PNG	/img_usb-image.dd/\$OrphanFiles/1.PNG	Recovered
5	3.PNG	.PNG	/img_usb-image.dd/\$OrphanFiles/3.PNG	Recovered

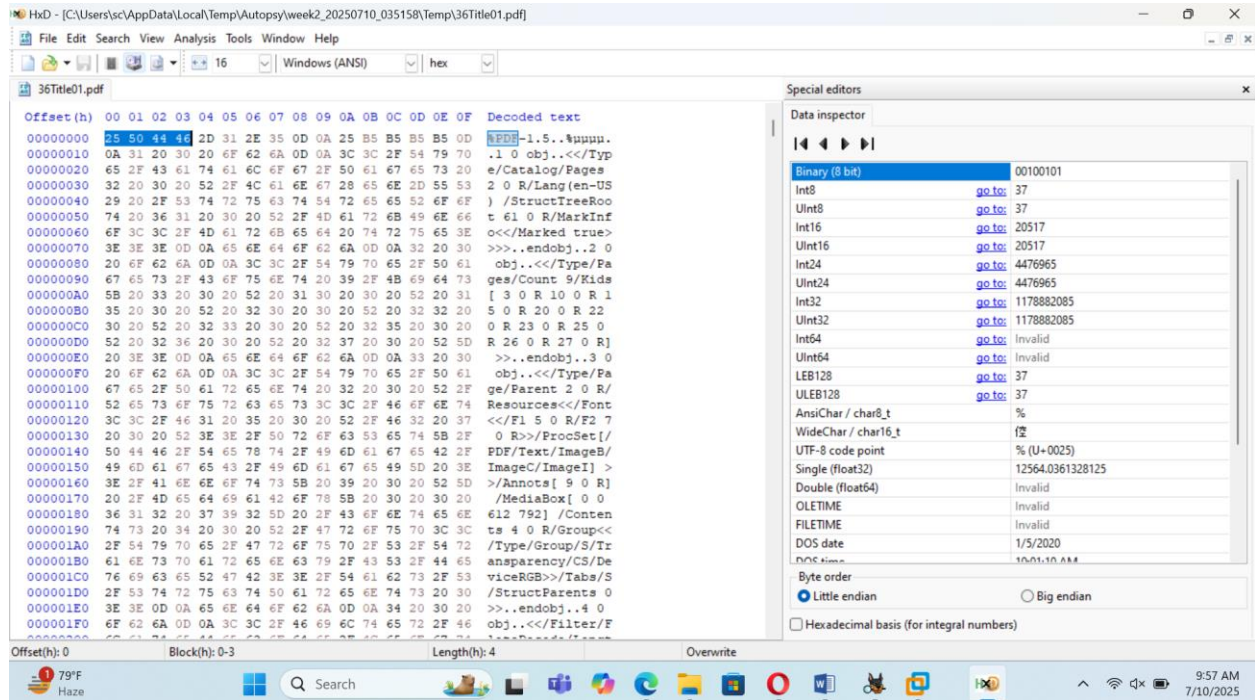
This file header is showing .png file.



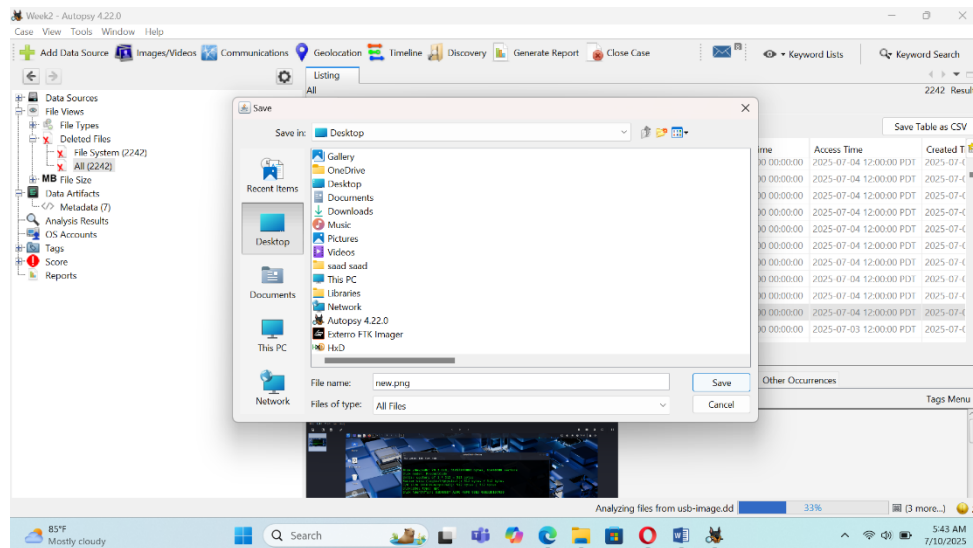
This file name new.png is and its header showing .png file.



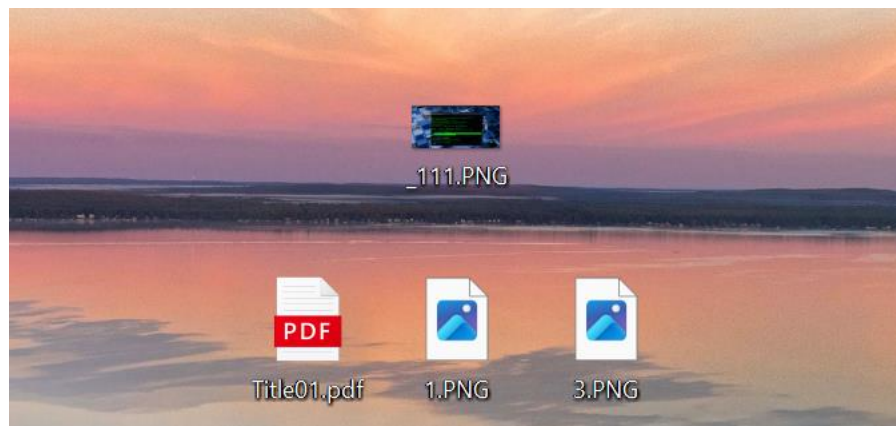
This file header showing its .pdf file.



This screenshot showing file successfully extract on desktop.



Showing extracted files.



Now Using Kali:

Open kali Linux and paste here extracted files.



Now using Exiftool and extracted file image10.jpg and this showing jpg metadata.

```
(syrian@kali)-[~/Desktop]
$ exiftool image10.jpg
ExifTool Version Number      : 13.10
File Name                    : image10.jpg
Directory                    : .
File Size                    : 44 kB
File Modification Date/Time   : 2025:07:10 20:17:31+05:30
File Access Date/Time        : 2025:07:10 20:17:55+05:30
File Inode Change Date/Time   : 2025:07:10 20:17:55+05:30
File Permissions              : -rw-----
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                  : 1.01
Resolution Unit               : inches
X Resolution                  : 96
Y Resolution                  : 96
Image Width                   : 727
Image Height                  : 537
Encoding Process              : Baseline DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)
Image Size                   : 727x537
Megapixels                   : 0.390

(syrian@kali)-[~/Desktop]
$
```

Now using another file PNG file and extract metadata.

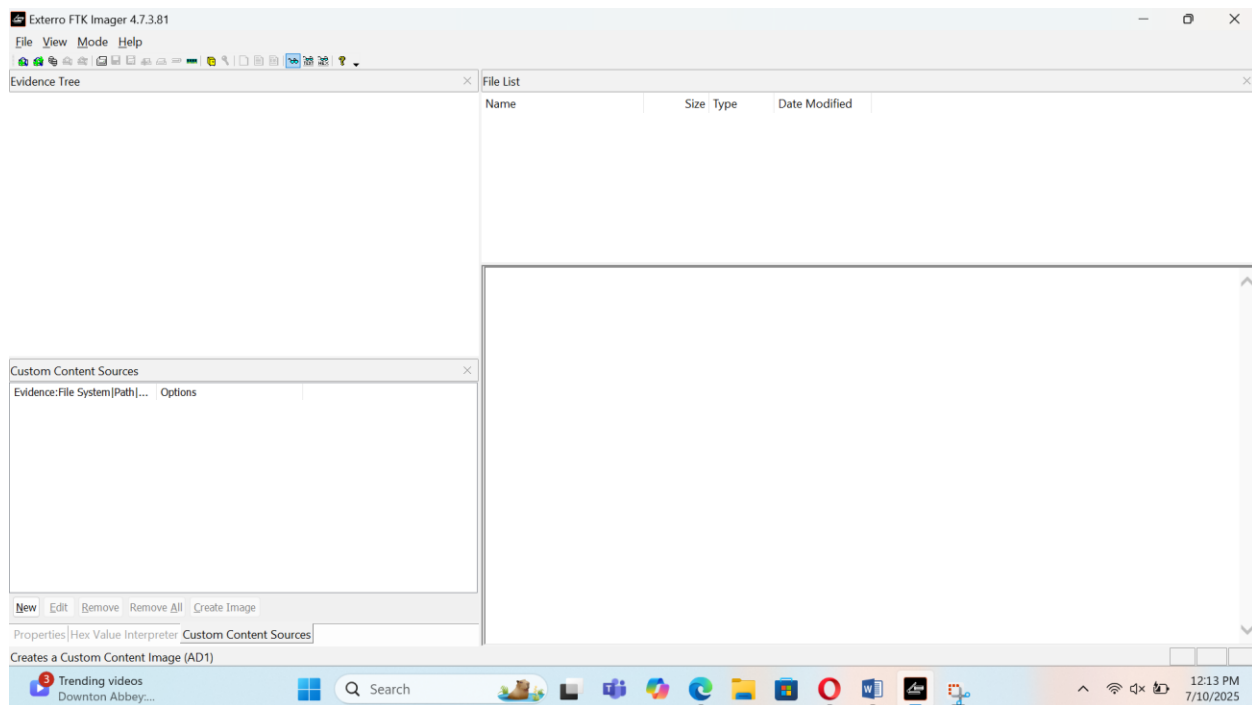
```
(syrian@kali)-[~/Desktop]
$ ls
1.PNG  3.PNG  image10.jpg  soclab.log  splunkforwarder-9.4.3-237ebbd22314-linux-amd64.deb

(syrian@kali)-[~/Desktop]
$ exiftool 1.PNG
ExifTool Version Number      : 13.10
File Name                    : 1.PNG
Directory                    : .
File Size                    : 745 kB
File Modification Date/Time   : 2025:07:10 18:21:52+05:30
File Access Date/Time        : 2025:07:10 20:15:08+05:30
File Inode Change Date/Time   : 2025:07:10 20:15:08+05:30
File Permissions              : -rw-----
Error                        : File format error

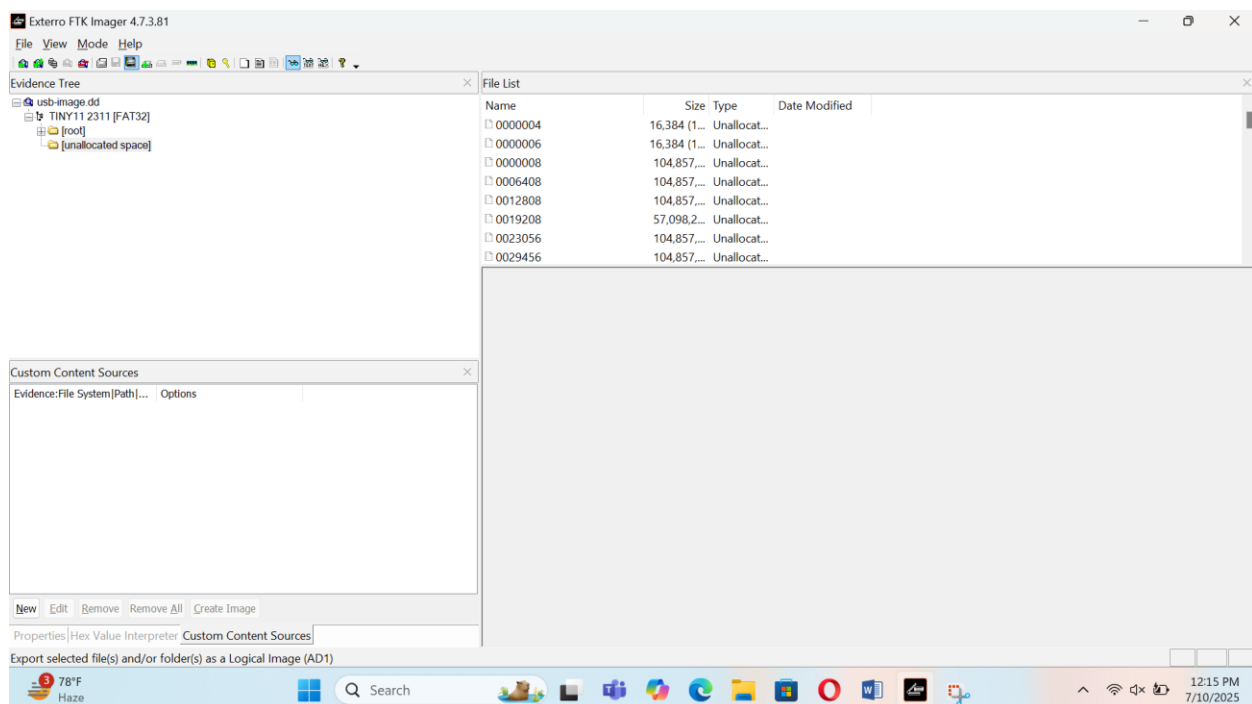
(syrian@kali)-[~/Desktop]
$
```

Scan unallocated space for deleted files.

Analyze unallocated space through FTK imager open FTK imager.



Now load image file in FTK imager and here is showing unallocated space and one file export on desktop.



Now identify unallocated file location through Autopsy and I uploaded here that file which I export on desktop.

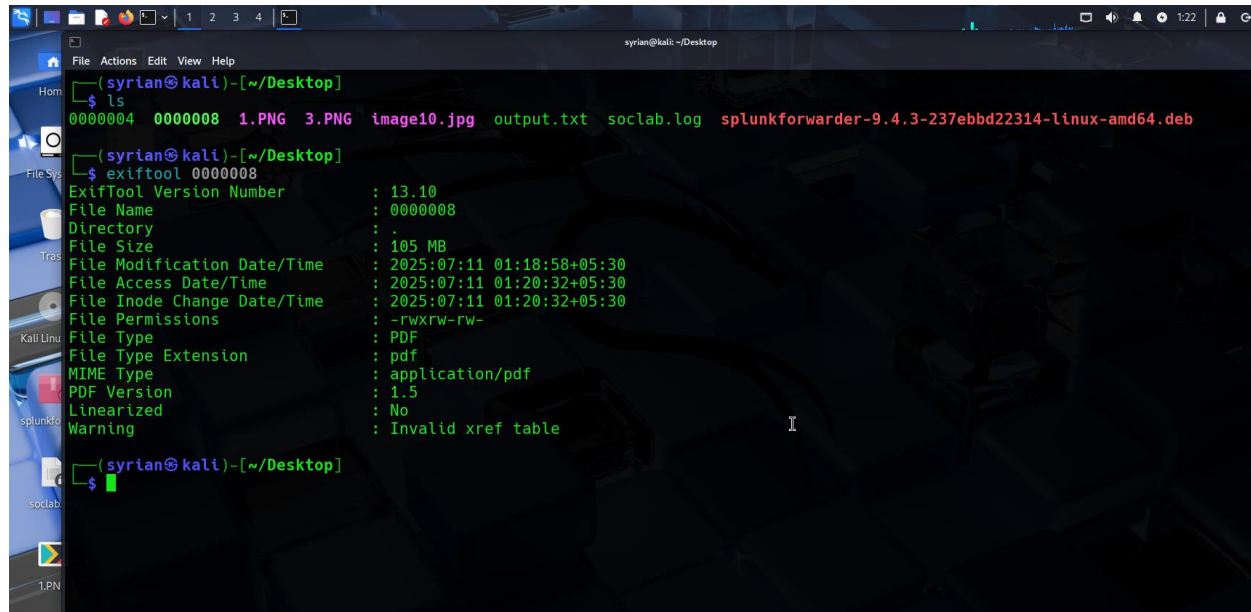
The screenshot shows the Autopsy 4.22.0 interface. The left sidebar displays the file tree structure, including 'Data Sources', 'File Views', and 'File Types'. The main pane shows the 'Listing' view for the path '/img_0000004/\$CarvedFiles/1'. A table lists the file 'f0000000.fat' with columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, and Flags(Dir). The file is marked as 'Unallocated'.

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags
f0000000.fat			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	65536	Unallocated	Unallocated

The screenshot shows the Autopsy 4.22.0 interface. The left sidebar displays the file tree structure, including 'Data Sources', 'File Views', and 'File Types'. The main pane shows the 'Listing' view for the path '/img_usb-image.dd/\$OrphanFiles'. A table lists the file 'f0000000.fat' with columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, and Flags(Dir). The file is marked as 'Unallocated'.

Name	S	C	O	Modified Time	Change Time	Access Time	Created Time	Size	Flags(Dir)	Flags
\$H^\$8H^D\$				2014-06-05 12:00:00 PDT	0000-00-00 00:00:00	1984-01-03 11:00:00 PST	2014-06-06 12:00:00 PDT	16384		
0f55235.341				1980-04-04 03:00:08 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	1980-05-07 12:00:00 PDT	16384		
0H^D\$@L^0^S				2016-08-01 05:26:16 PDT	0000-00-00 00:00:00	1980-04-15 11:00:00 PST	1998-04-12 05:26:16 PDT	16384		
0H^D\$@L^0^S				2016-08-01 05:26:16 PDT	0000-00-00 00:00:00	1980-04-15 11:00:00 PST	1998-04-12 05:26:16 PDT	16384		
2^v^@^g.jh^				2024-01-16 11:03:50 PST	0000-00-00 00:00:00	1980-01-15 11:00:00 PST	1980-01-08 11:00:16 PST	16384		
6_CBC_SHA38				2013-10-04 12:00:00 PDT	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		
6_CBC_SHA38				2016-02-03 11:00:00 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		
6_GCM_SHA38				2018-02-19 11:00:00 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		
6_GCM_SHA38				2018-02-19 11:00:00 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		
8_CBC_SHA25				2018-02-19 11:00:00 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		
8_GCM_SHA25				2018-02-19 11:00:00 PST	0000-00-00 00:00:00	0000-00-00 00:00:00	0000-00-00 00:00:00	16384		

Found unallocated space file.

A terminal window on a Kali Linux desktop. The user is in the ~/Desktop directory. They run 'ls' showing files: 00000004, 00000008, 1.PNG, 3.PNG, image10.jpg, output.txt, soclab.log, and splunkforwarder-9.4.3-237ebbd22314-linux-amd64.deb. Then they run 'exiftool 00000008', which displays detailed EXIF data for a PDF file named 00000008, including version (13.10), size (105 MB), and timestamps. A warning about an invalid xref table is also shown.

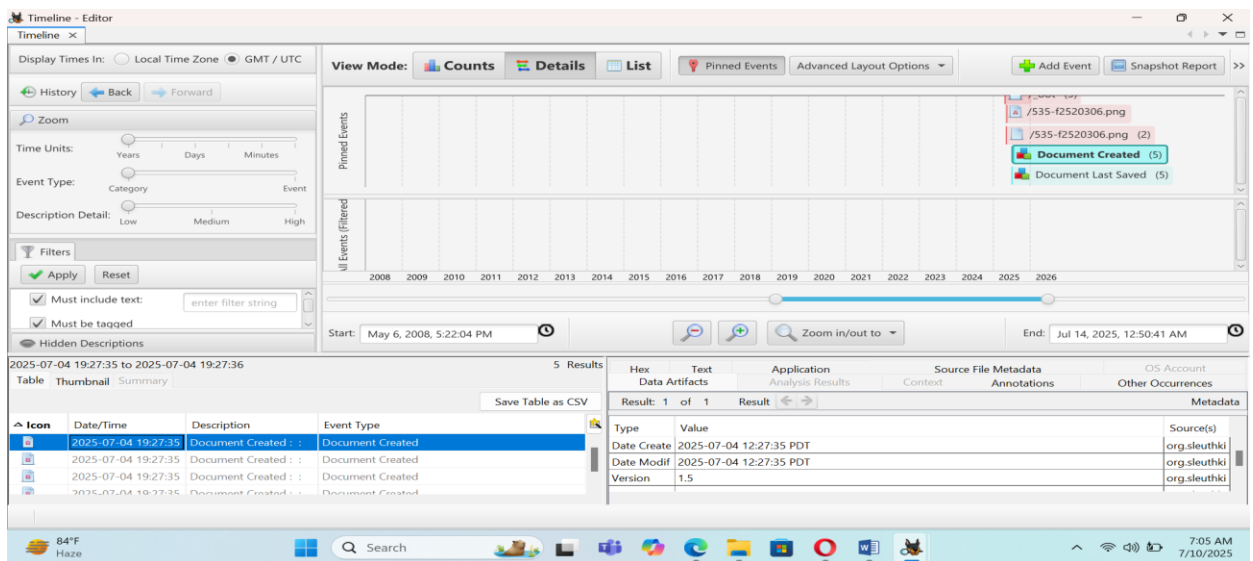
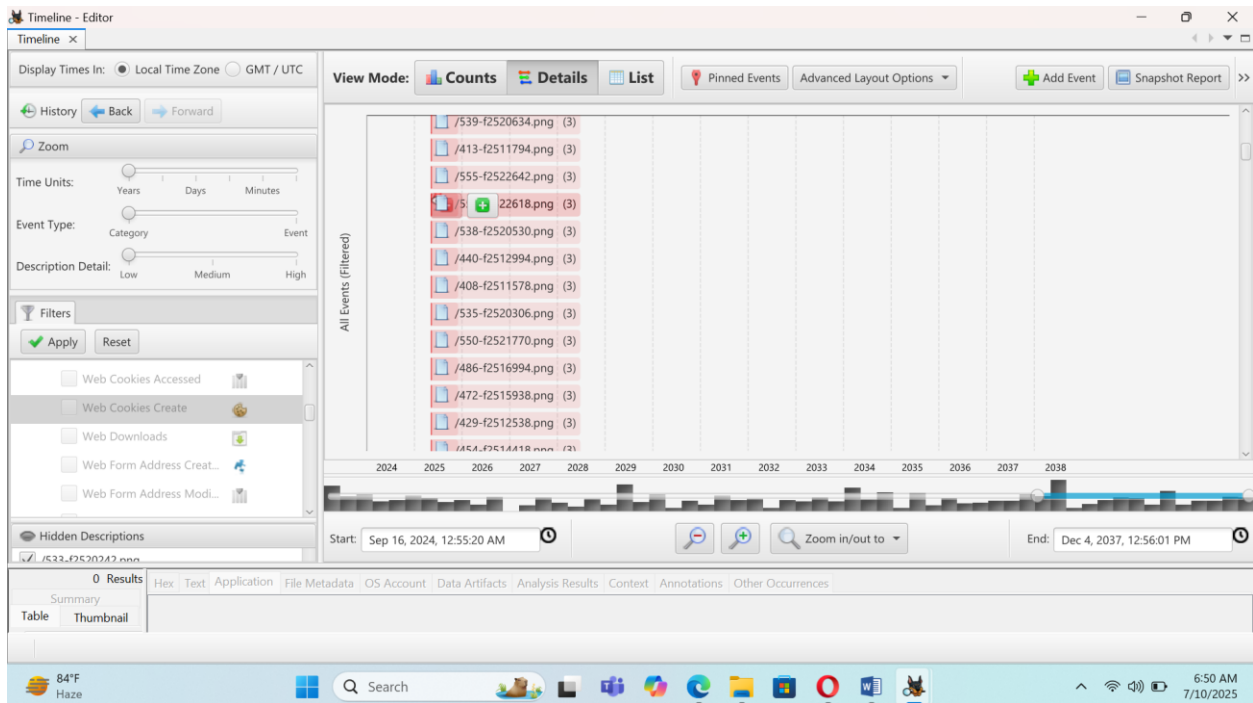
```
(syrian@kali)-[~/Desktop]
$ ls
00000004  00000008  1.PNG  3.PNG  image10.jpg  output.txt  soclab.log  splunkforwarder-9.4.3-237ebbd22314-linux-amd64.deb

(syrian@kali)-[~/Desktop]
$ exiftool 00000008
ExifTool Version Number      : 13.10
File Name                    : 00000008
Directory                   : .
File Size                    : 105 MB
File Modification Date/Time  : 2025:07:11 01:18:58+05:30
File Access Date/Time       : 2025:07:11 01:20:32+05:30
File Inode Change Date/Time  : 2025:07:11 01:20:32+05:30
File Permissions             : -rwxrwx-rw-
File Type                    : PDF
File Type Extension          : pdf
MIME Type                    : application/pdf
PDF Version                  : 1.5
Linearized                   : No
Warning                      : Invalid xref table

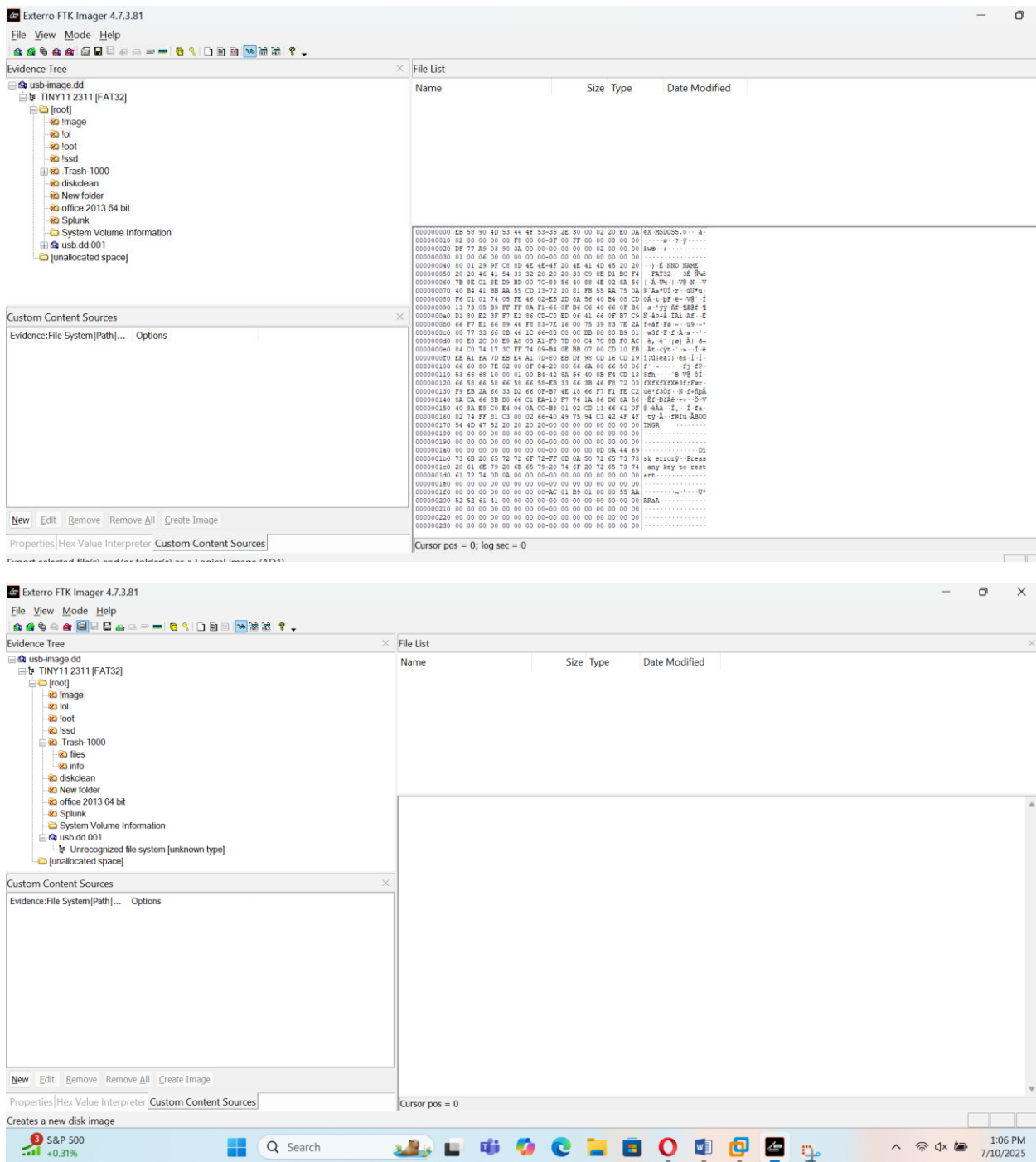
(syrian@kali)-[~/Desktop]
$
```

Step 3: Timeline Linking

- In Autopsy, select to “**Timeline**” module
- Filter by Time Range and File Types
- I track user actions:
- When files were created, opened, deleted



Step 6 Document Suspicious Findings:



During my analysis using FTK Imager, I carefully reviewed the file system, metadata, and file headers. No password-protected, encrypted, or renamed files were identified. All timestamps appeared consistent and there was no sign of tampering. Based on this examination, no suspicious findings were observed in the recovered files from this disk image.

Conclusion

In this task, I analyzed a disk image using **Autopsy** and **FTK Imager** to recover deleted and hidden files. I was able to successfully recover **five deleted files** like images and a PDF. These files were found in the system's Trash and hidden folders.

Then, I used **ExifTool in Kali Linux** to check the **metadata** of the recovered image files. This helped me see extra information like file creation time and other hidden details.

Next, I scanned the **unallocated space** (empty space where deleted data can still exist) and found another file. I exported it using FTK Imager and confirmed its location with Autopsy.

I also checked the **timeline** of file activities like when files were created, opened, or deleted.

In the end, I **didn't find any suspicious activity** no hidden, renamed, encrypted, or password-protected files. All the file details and timestamps were normal.

This task helped me improve my practical skills in digital forensics like file recovery, metadata checking, and timeline analysis.