

Digital Forensic Image Recovery using Autopsy, Metadata Analysis, and Hex-Based Extraction

Abstract

This investigation demonstrates a systematic digital forensic process for locating and recovering ten images embedded within a forensic disk image (forensic.dd). Using Autopsy, the initial analysis applied filtering techniques to reduce the dataset to relevant image file types. Thumbnails and metadata examination led to the discovery of multiple images, while compressed archives were further explored to extract additional files. For images concealed within document files, keyword searches and hexadecimal analysis were performed to identify JPEG signatures. The dd command was then used to extract the image data from the binary streams with byte-level precision. By applying a combination of forensic tools and investigative strategies, all ten target images were successfully located and recovered, illustrating the importance of reduction, metadata inspection, and low-level hex analysis in practical digital forensics.

Investigation Process

To efficiently narrow the dataset, Autopsy's file type filtering function was applied to isolate only image-related files (Figures 2.1–2.3). This approach significantly reduced irrelevant data and enabled faster identification of potential evidence. From this process, six items were revealed: file1.jpg, file2.jpg, file6.jpg, file7.jpg, file13.dll, and UpCase.

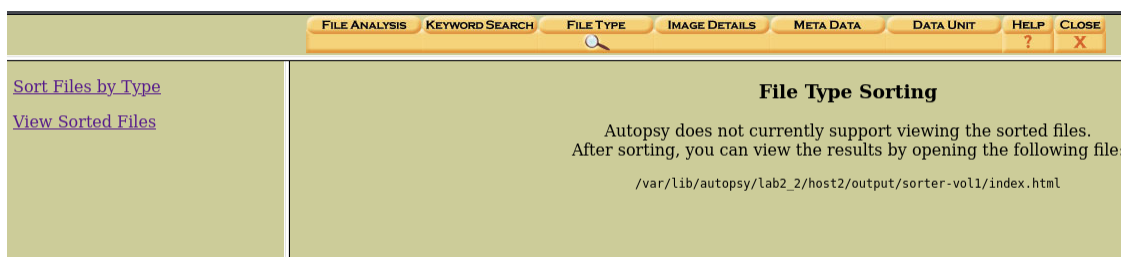


Figure 2.1

sorter output

Images

- /var/lib/autopsy/lab2_2/host2/images/forensic.dd

Files (45)

Files Skipped (14)

- Non-Files (14)
- Reallocated Name Files (0)
- 'ignore' category (0)

Extensions

- [Extension Mismatches](#) (5)

Categories (31)

- archive (0)
- audio (0)
- [compress](#) (3)
- crypto (0)
- [data](#) (15)
- [disk](#) (1)
- [documents](#) (1)
- exec (0)
- [images](#) (6)
- system (0)
- [text](#) (1)
- [unknown](#) (4)
- video (0)

Figure 2.2

images Category

C:/UpCase
Targa image data - Map 6 x 7 x 8 +4 +5
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 10-128-1

C:/alloc/file1.jpg
JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 698x752, components 3
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 29-128-3

C:/alloc/file2.dat
JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 437x365, components 3
--- Extension Mismatch! ---
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 28-128-3

C:/del1/file6.jpg
JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 563x527, components 3
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 32-128-3

C:/del2/file7.hmm
JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 698x752, components 3
--- Extension Mismatch! ---
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 31-128-3

C:/misc/file13.dll:here
JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 518x563, components 3
--- Extension Mismatch! ---
Image: /var/lib/autopsy/lab2_2/host2/images/forensic.dd Inode: 44-128-5

Figure 2.3

According to the findings in Autopsy, as shown in Figures 2.4-2.8, five of the required images were successfully located by expanding the directories and reviewing their thumbnails. The images found were pictures #1, #2, #3, #4, and #10.

Directory Seek

Enter the name of a directory that you want to view.
C:/

VIEW

File Name Search

Enter a Perl regular expression for the file names you want to find.

SEARCH

ALL DELETED FILES

HIDE DIRECTORIES

C:/

+ /\$Extend

+ /alloc

+ /archive

+ /del1

+ /del2

+ /invalid

+ /misc

+ /misc/CLEB

+ /++

\$-1-5-21-1757981266-484763869-1060284298-1003

+ /System Volume Information

Current Directory: C:/ alloc/

ADD NOTE

GENERATE MD5 LIST OF FILES

DEL	Type	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE	UID	GID	META
	d / d	del	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:22:22 (EDT)	56	48	0	5-144-6
	d / d	del	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:06 (EDT)	256	0	0	27-144-1
	r / r	file1.jpg	2004-06-10 02:59:40 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	274260	0	0	29-128-3
	r / r	file2.dat	2004-06-10 02:46:52 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	2004-06-09 23:27:36 (EDT)	26081	0	0	28-128-3

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * View * Add Note

File Type: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 698x752, components 3

C:/alloc/file1.jpg

Thumbnail: [View Full Size Image](#)




Figure 2.4

ASCII (display - report) * Hex (display - report) * ASCII Strings (display - report) * Export * View * Add Note

File Type: JPEG image data, JFIF standard 1.01, aspect ratio, density 1x1, segment length 16, baseline, precision 8, 437x365, components 3

C:/alloc/file2.dat

Thumbnail: [View Full Size Image](#)




Figure 2.5

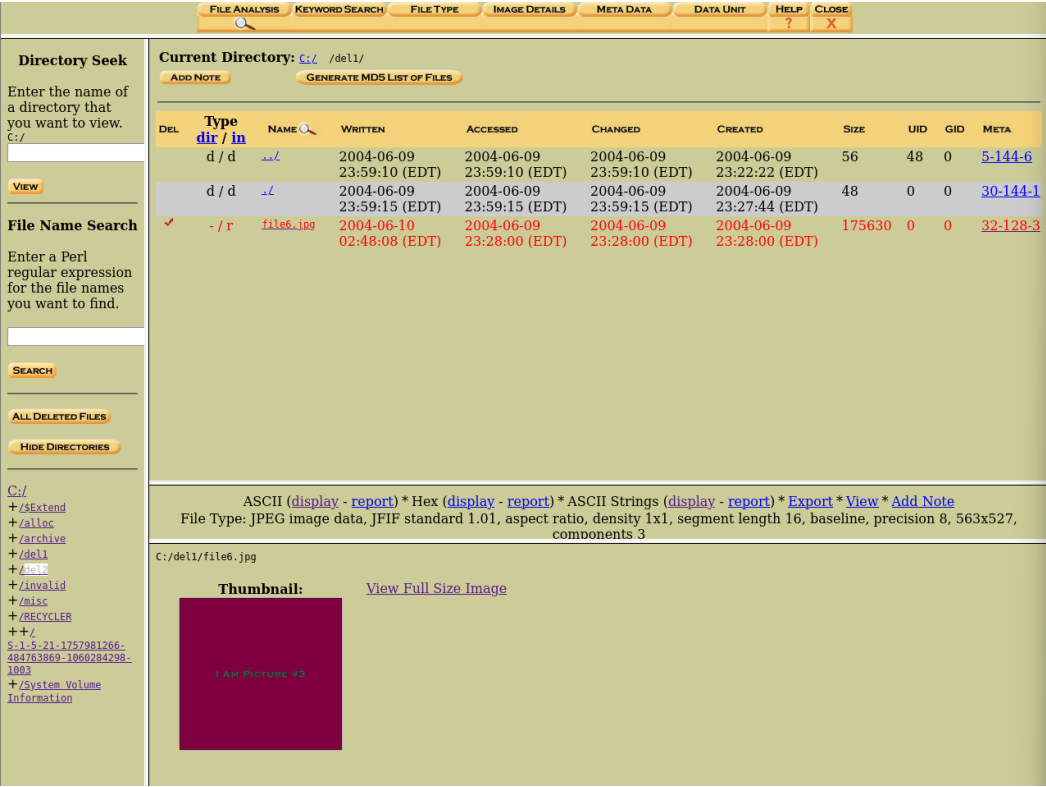


Figure 2.6



Figure 2.7

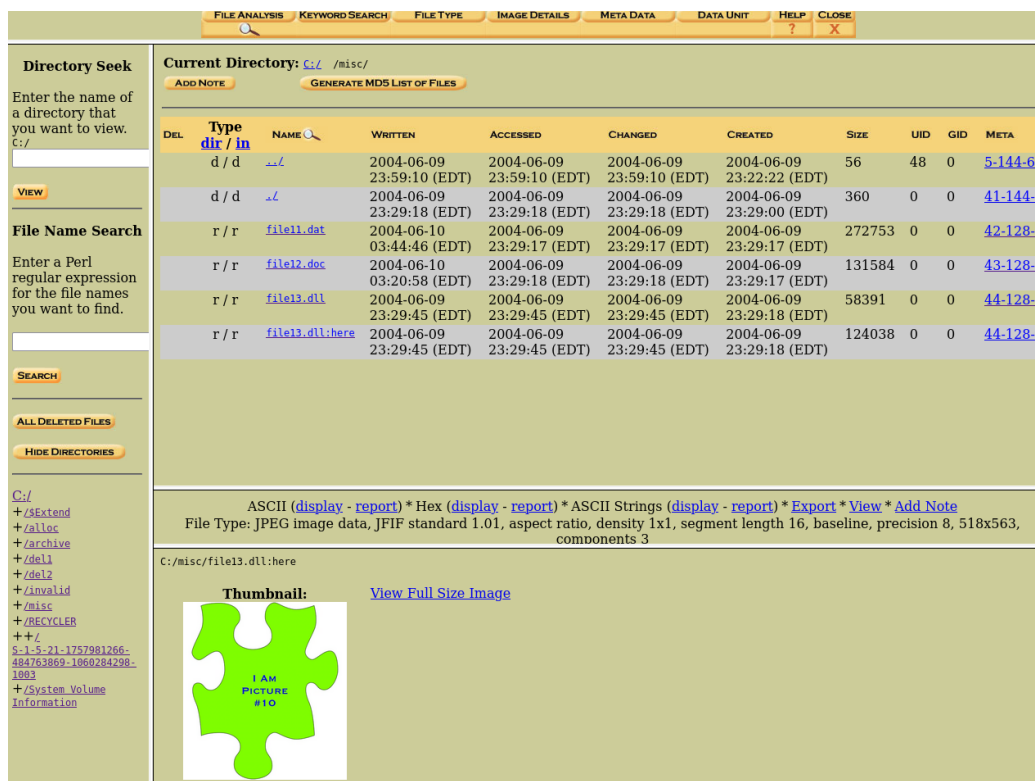


Figure 2.8

Further filtering by compressed file types uncovered file8.zip, file9.boo, and file10.tar (Figures 2.9–2.10). Inspection of the metadata for file8.zip revealed a reference to file8.jpg (Figure 2.11). Extraction of the archive confirmed the presence of Picture #5 (Figure 2.12).

Similarly, exporting and examining the remaining compressed files yielded Picture #6 (Figure 2.13) and Picture #7 (Figure 2.14). This demonstrated the importance of expanding beyond standard image categories when evidence may be embedded within archives.

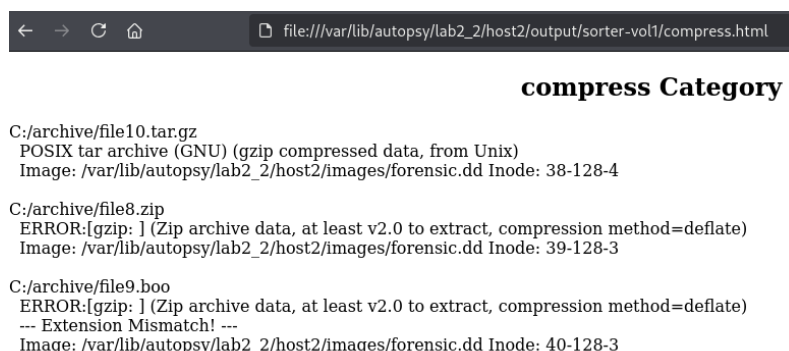


Figure 2.9

Current Directory: C:/archive/										
ADD NOTE GENERATE MD5 LIST OF FILES										
DEL	Type	NAME	WRITTEN	ACCESSED	CHANGED	CREATED	SIZE	UID	GID	META
d / d	dir / in	..	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:59:10 (EDT)	2004-06-09 23:22:22 (EDT)	56	48	0	5-144-6
d / d	dir / in	..	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:52 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:37 (EDT)	472	0	0	37-144-1
r / r	file	file10.tar.gz	2004-06-10 03:18:54 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:50 (EDT)	207272	0	0	38-128-4
r / r	file	file8.zip	2004-06-10 03:16:42 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:51 (EDT)	2004-06-09 23:28:51 (EDT)	335371	0	0	39-128-3
r / r	file	file9.boo	2004-06-10 03:17:46 (EDT)	2004-06-09 23:28:54 (EDT)	2004-06-09 23:28:54 (EDT)	2004-06-09 23:28:51 (EDT)	294124	0	0	40-128-3

Figure 2.10

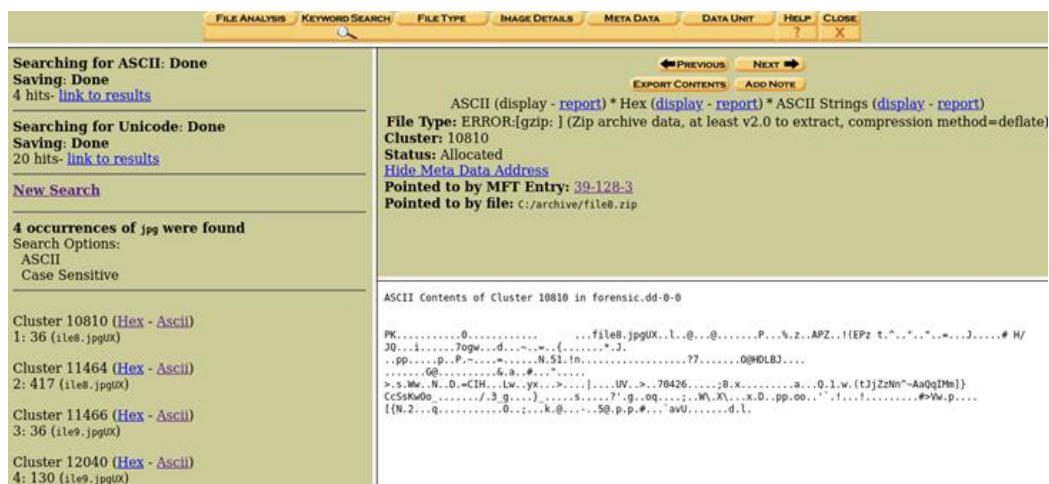


Figure 2.11

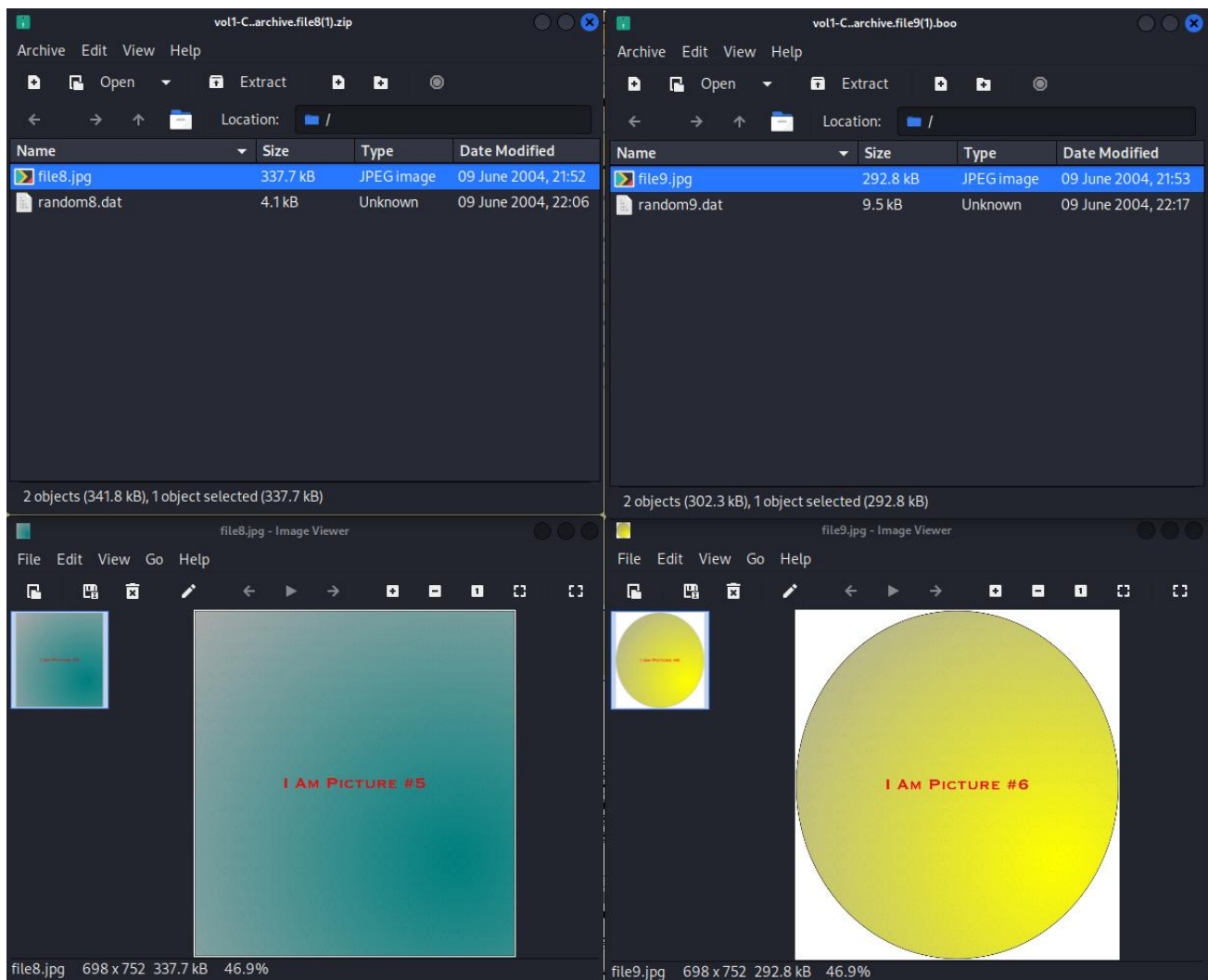


Figure 2.12

Figure 2.13

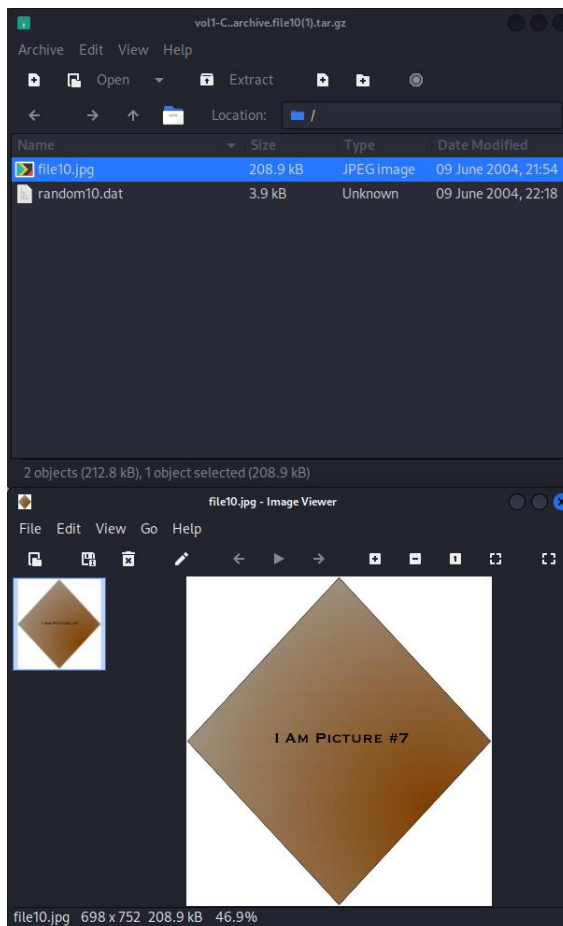


Figure 2.14

To locate the remaining images, a keyword search was performed within Autopsy. An initial search for the phrase “I AM PICTURE” returned no results. A refined search using the keyword “jpg” identified a reference within C:/misc/file.doc, which contained the string Pict.jpg (Figure 2.15).

The file was exported and opened with GHex, a hexadecimal editor. A search for the JPEG magic number (FF D8) successfully identified embedded image data (Figure 2.16). Using the dd command, the binary stream was extracted with precision: `dd if=file12.doc of=file12.jpg bs=1 skip=4937`

This process created file12.jpg, corresponding to Picture #9 (Figures 2.17–2.18).

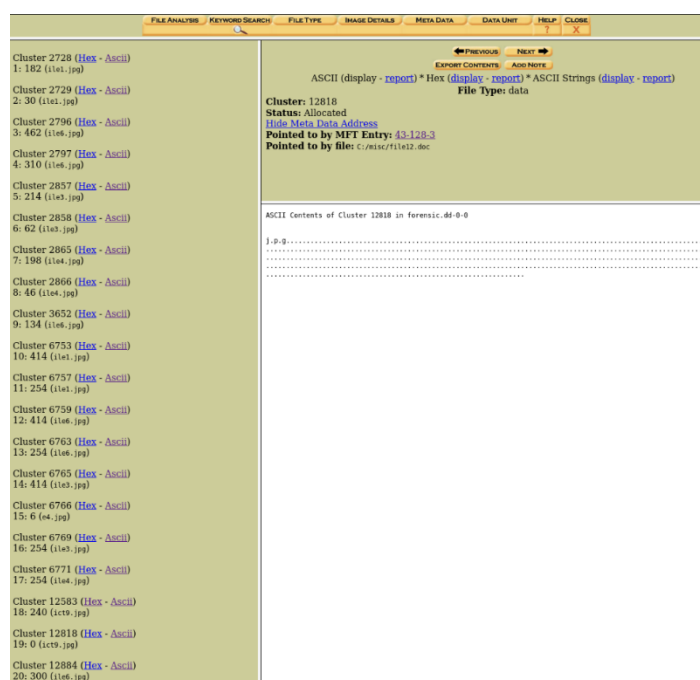


Figure 2.15

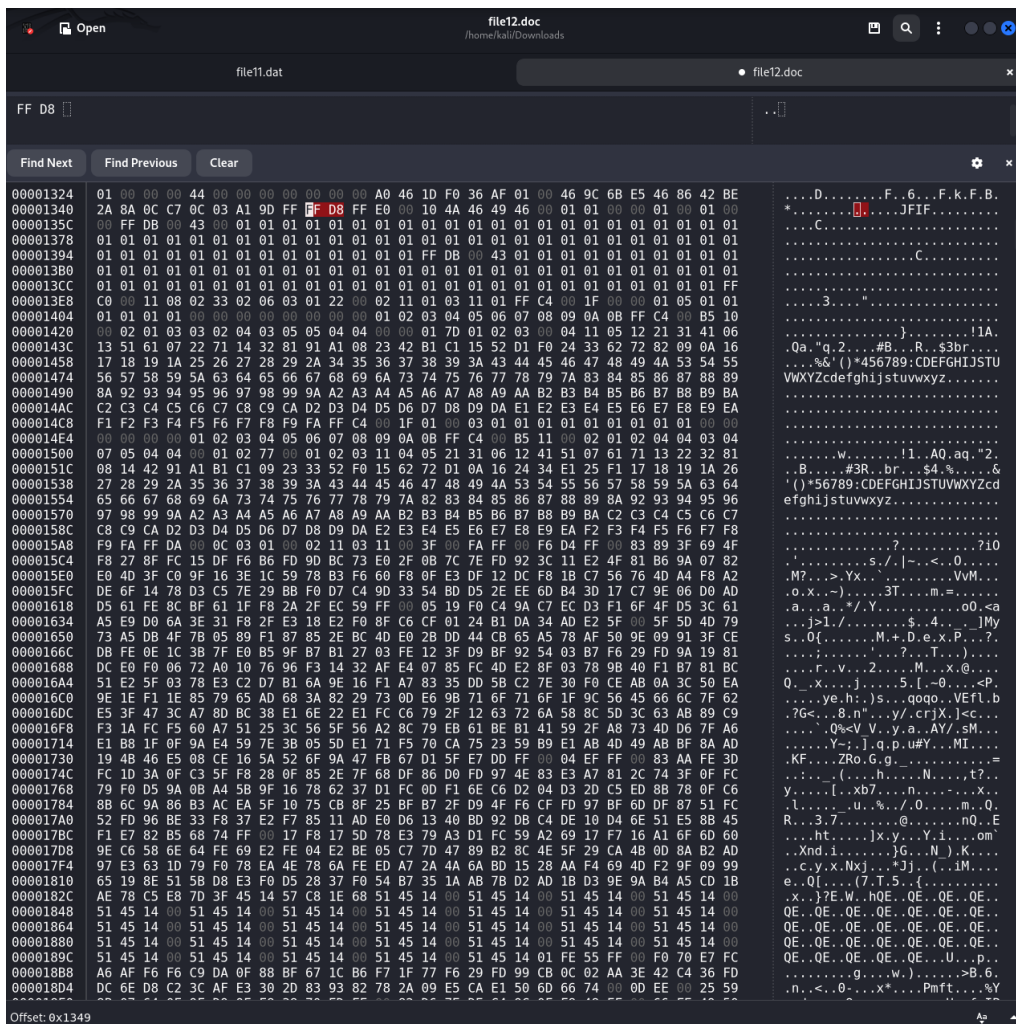


Figure 2.16

```
(kali㉿kali)-[~/Downloads]
$ dd if=file12.doc of=file12.jpg bs=1 skip=4937
126647+0 records in
126647+0 records out
126647 bytes (127 kB, 124 KiB) copied, 0.090189 s, 1.4 MB/s
```

Figure 2.17

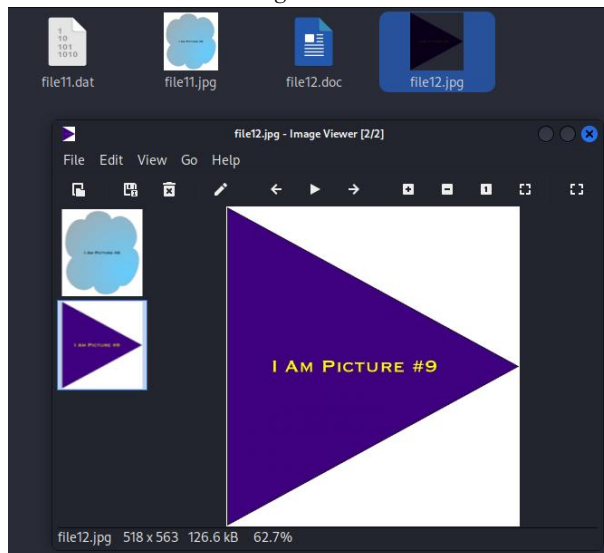


Figure 2.18

Applying the same method to file11, located in the same directory as file12, revealed a JFIF signature in GHex (Figure 2.19). Again, the dd command was used to extract the data (Figure 2.20), successfully recovering Picture #8 (Figure 2.21).

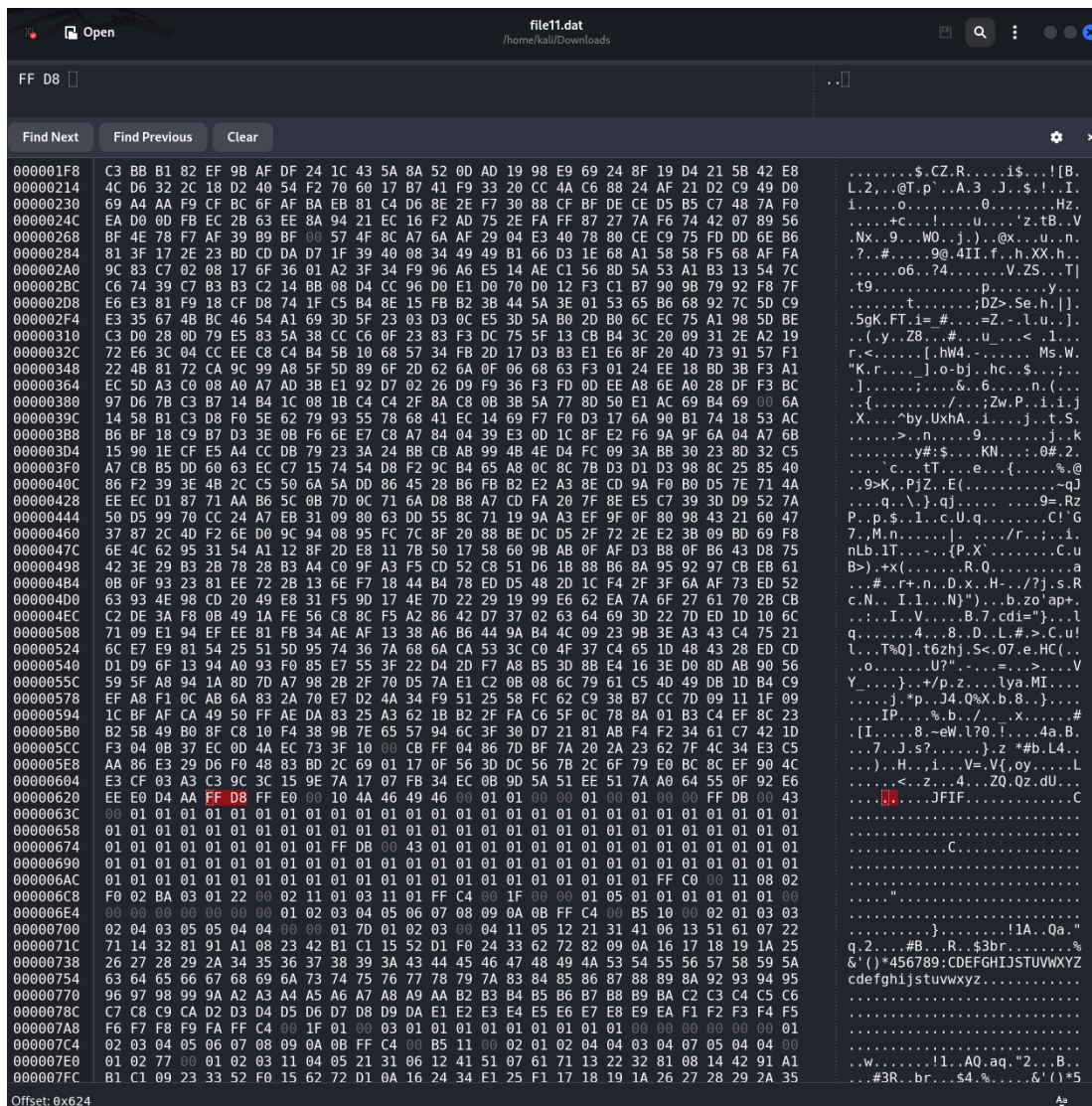


Figure 2.19

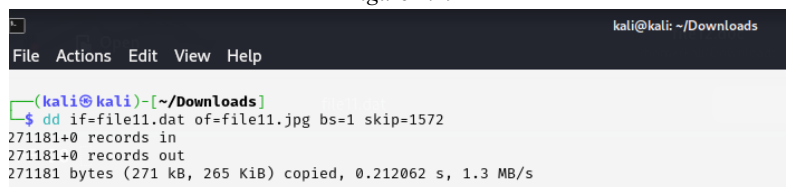


Figure 2.20

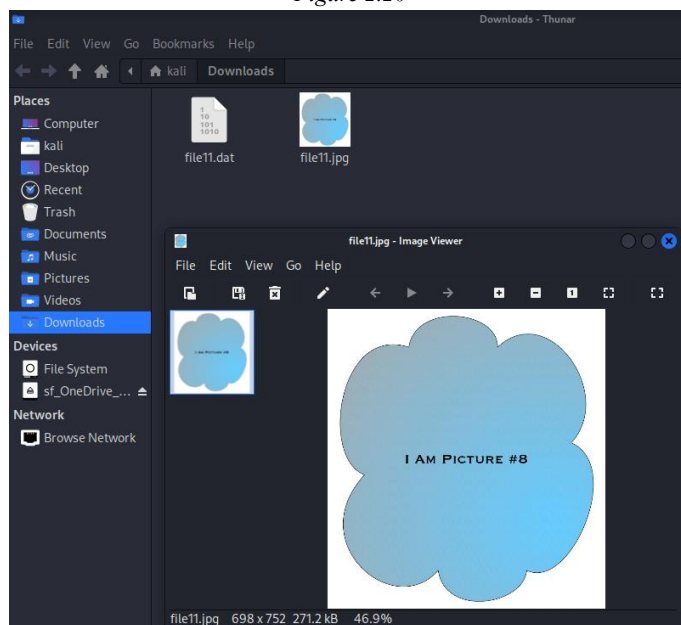


Figure 2.21

Conclusion

This investigation demonstrates a systematic digital forensic process for locating and recovering ten images embedded within a forensic disk image (forensic.dd). Using Autopsy, the initial analysis applied filtering techniques to reduce the dataset to relevant image file types. Thumbnails and metadata examination led to the discovery of multiple images, while compressed archives were further explored to extract additional files. For images concealed within document files, keyword searches and hexadecimal analysis were performed to identify JPEG signatures. The dd command was then used to extract the image data from the binary streams with byte-level precision. By applying a combination of forensic tools and investigative strategies, all ten target images were successfully located and recovered, illustrating the importance of reduction, metadata inspection, and low-level hex analysis in practical digital forensics.