

Analytical Forensic Investigation with Data Carving Tools

Dr. Priya P. Sajan*

*Senior Project Engineer at Centre For Development Of Advanced Computing (C-DAC), Thiruvananthapuram, Kerala, India

Neha S. Rokade**

Dinesh M. More**

Mahesh K. Adsure**

Mayur R. Thakare**

Purushottam P. Thutte**

**PG-Diploma in Cyber Security and Forensics, Centre For Development Of Advanced Computing (C-DAC), Thiruvananthapuram, Kerala, India

Abstract:- Data carving and file recovery are techniques for recovering lost or deleted files and data from storage media such as hard discs, flash drives, and memory cards. These methods can be used to recover data that have been deleted by accident, lost due to a hardware failure, or are otherwise unreachable. In digital investigation and computer forensics, data carving is a critical issue. As a result, research into enhancing data carving techniques is required to enable digital investigators to recover critical data and evidence from damaged or corrupted data resources. The purpose of a Foremost file recovery and data carving study would be to examine and assess the tool's capabilities and performance in recovering lost or deleted files and data from a range of storage systems. Testing may entail testing the tool on various storage devices, utilising various file kinds and recovery scenarios, and comparing the tool's performance to that of competing file recovery and data carving tools.

Keywords:- Digital Investigation, Computer Forensics, File Recovery, Data Carving, Foremost Tool.

I. INTRODUCTION

The technique of collecting evidence from both the digital and physical environments in a crime by preserving the data in its original form for use in court is known as digital forensics. In the case of digital crime, the investigator should collect and preserve all evidence or other material related to the crime found in digital media such as (computers, cameras, and networks...etc.) in order to conduct an investigation based on it, as well as to establish a timeline and determine the sequence of events. It should be remembered that certain digital information may be unavailable or purposely removed. As a result, the investigator must look through both existing and pre-existing data (such as data that has been erased or crashed) in order to recover it using some method.

Metadata, sometimes known as "data about data," assists the operating system in identifying data. Metadata contains technical information such as the data's creation and modification dates, as well as the file type. This information makes it much easier to find and index files. Instead of standard metadata produced by or connected with filesystems, file carving pulls data and files from unallocated space utilising specified criteria such as the file structure and file headers.

Even if the file extension has been modified or removed entirely, file headers retain information that may be used to identify the file type and dissect the file by analysing header and footer information. Data carving is a time-consuming procedure that should be completed utilising automated technologies. It also helps if the investigator knows what file types they are looking for in order to focus better and save time. But, this is forensics, and we know that patience and time are essential. Some typical file types, as shown in hexadecimal notation inside the file headers, are as follows:

- *Joint Photographic Experts Group (JPEG): FF D8 FF E0*
- *Portable Document Format (PDF): 25 50 44 46*

Hashing (ideally SHA-256) of all carved data, recovered files and media should be undertaken in accordance with best practices and effective case management. This stage is critical for investigators and follows worldwide best practices.

II. FORENSIC INVESTIGATION WITH DATA CARVING TOOLS

A. Foremost Tool:

➤ Forensic Test Images Used in Foremost:

In this project, we will be using a digital forensic tool-testing image to test data carving tools. Specifically, we will be utilizing the 11-carve-fat.dd example image of the FAT32 file system, which can be found online.

➤ File Recovery and Data Carving Using Foremost:

Foremost is a straightforward and efficient **command line interface (CLI)** utility for recovering files by scanning their headers and footers. We may begin by selecting **Apps | 11 - Forensics | foremost**.

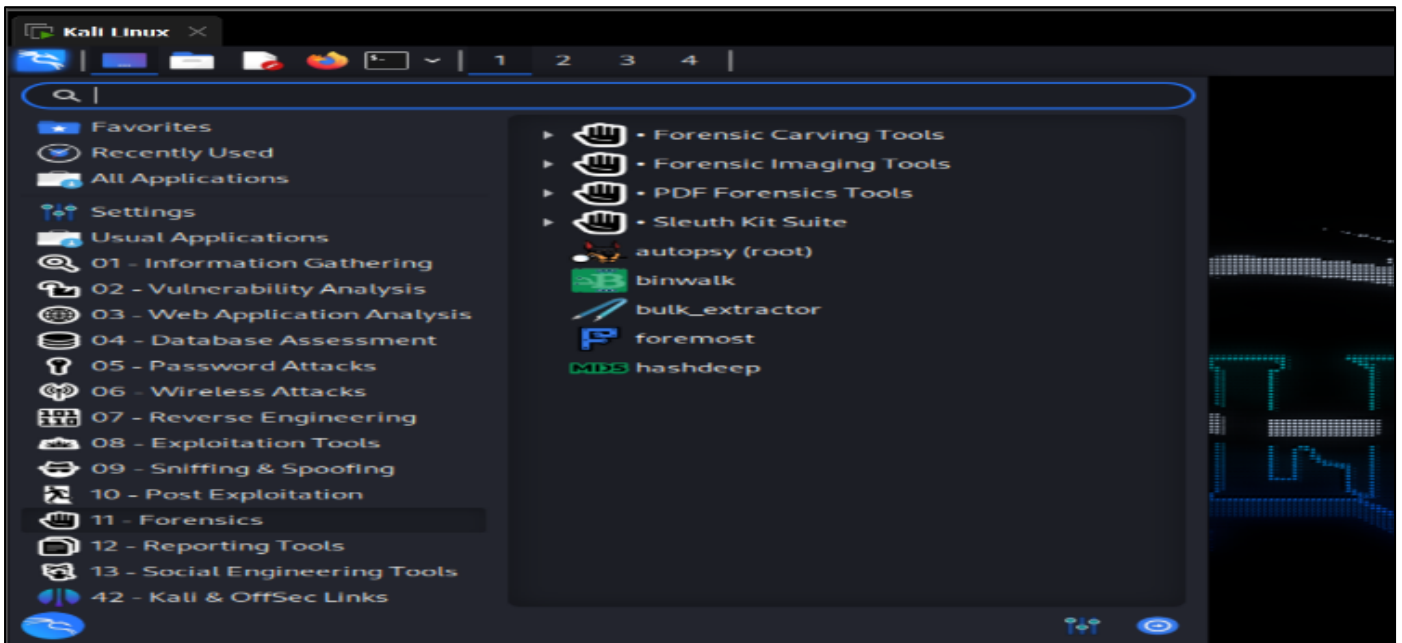


Fig 1 Foremost in Kali Linux

- If foremost is not installed on your version of Kali Linux, install it by typing:

- `Sudo Apt-Get Install Foremost.`

Once the interface has been successfully started, a terminal will open showing the version of the program, the creator, and some of the many switches to use:

```

(root@kali)-[~/home/kali/Desktop/cdac]
└─# foremost -h
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w|-d] [-t <type>] [-s <blocks>] [-k <size>]
  [-b <size>] [-c <file>] [-o <dir>] [-i <file>]

-V - display copyright information and exit
-t - specify file type. (-t jpeg,pdf ...)
-d - turn on indirect block detection (for UNIX file-systems)
-i - specify input file (default is stdin)
-a - Write all headers, perform no error detection (corrupted files)
-w - Only write the audit file, do not write any detected files to the disk
-o - set output directory (defaults to output)
-c - set configuration file to use (defaults to foremost.conf)
-q - enables quick mode. Search are performed on 512 byte boundaries.
-Q - enables quiet mode. Suppress output messages.
-v - verbose mode. Logs all messages
  
```

Fig 2 Foremost Help Options

- Go through the front-end system driver manual to better understand the front end and the switches used. Enter the following command to do this:
- *Man Foremost*
The output shows the user manual for everything and the supported file formats:

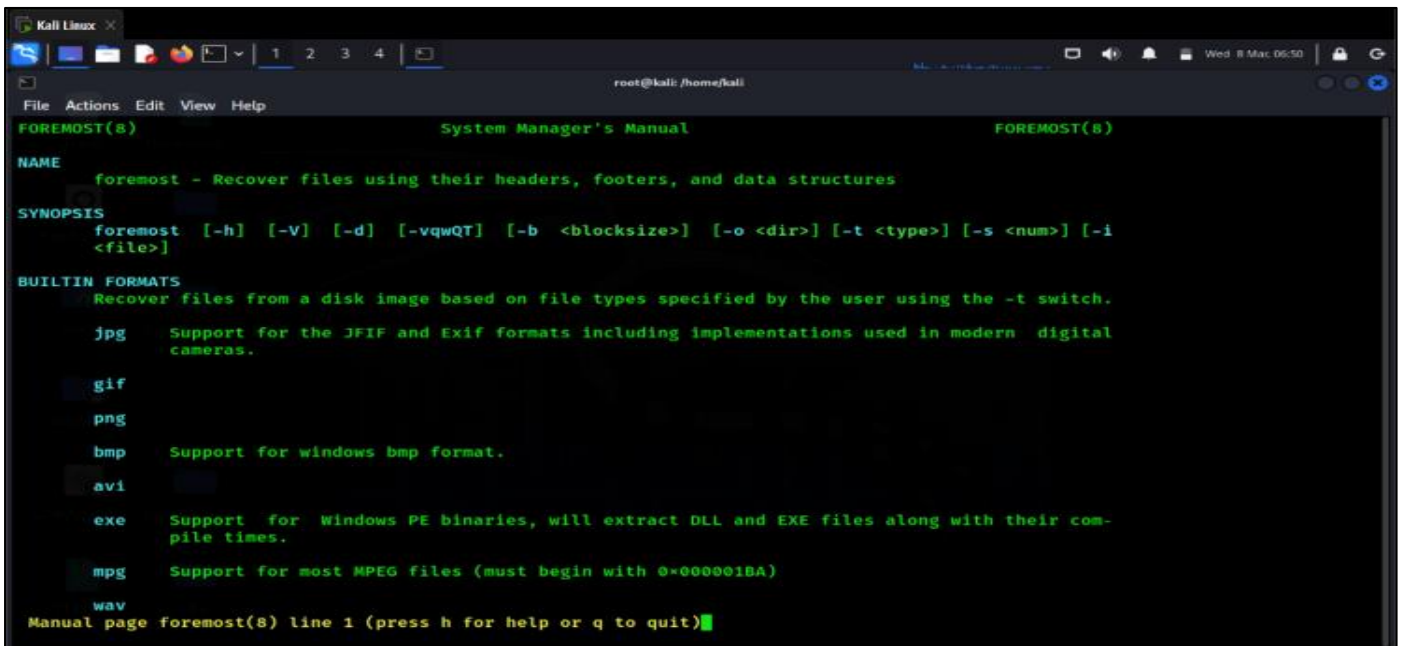


Fig 3 Supported File Types

- The Syntax for Using Foremost is as Follows:
- *Foremost -I (Forensic Image) -O (Output Folder) -Options*
In this example, we specify the 11-carve-fat.dd file in the specified folder as the input file (-i) and an empty folder named foremostrecovery as the output file (-o). Additionally, other switches can be specified as required. It should be mentioned that all file locations for imaging and etching data (although referred to as desktop in these exercises) should be unique to the case and may even be stored on forensically reliable media, in accordance with proper case management.
- To Begin Carving the 11-Carve-Fat.Dd Image with Foremost, we Type the Following Command in the Terminal:
- *Foremost -I 11-Carve-Fat.Dd -O Foremostrecovery*
The following image shows the command in the Terminal:

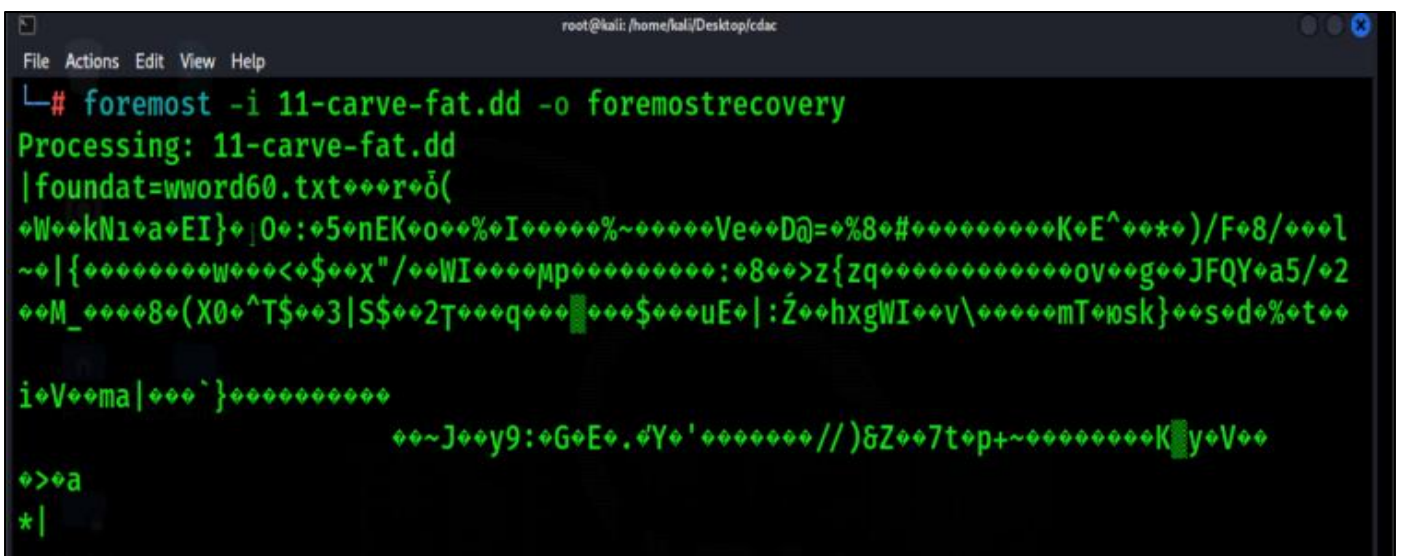


Fig 4 Foremost Carving Process

Although the characters discovered appear hazy during processing, the findings will be properly sorted and summarized in the output folder provided. If the selected output folder is not empty, you will experience issues. When the procedures are finished, we can browse to our output folder to examine the results.

➤ *Viewing The Foremost Results:*

After the Foremost tool has completed the carving process, the next step would be to access the output folder where the recovered files have been stored. This folder is named "foremostrecovery" and can be found in the same directory where the Foremost tool was executed:

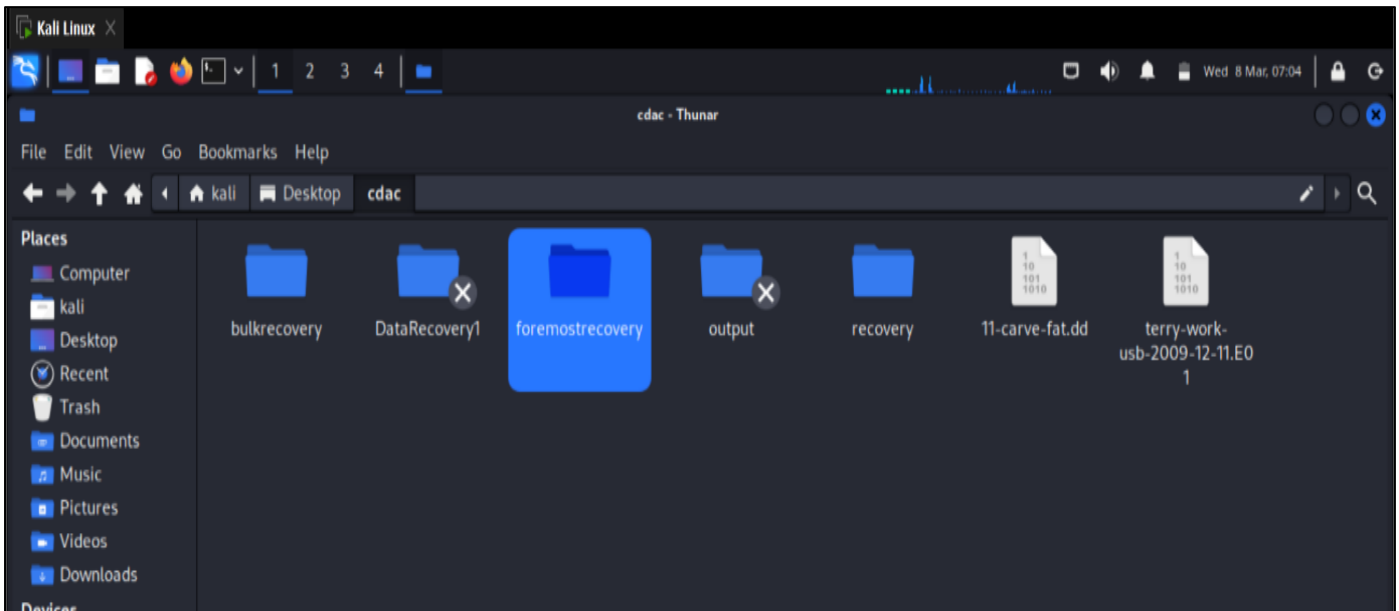


Fig 5 Foremost Output Directory

Upon opening the output directory "foremostrecovery", you will find that the recovered files are categorized according to their file types. Additionally, there will be an audit.txt file within the directory which contains a detailed report of the findings from the carving process:

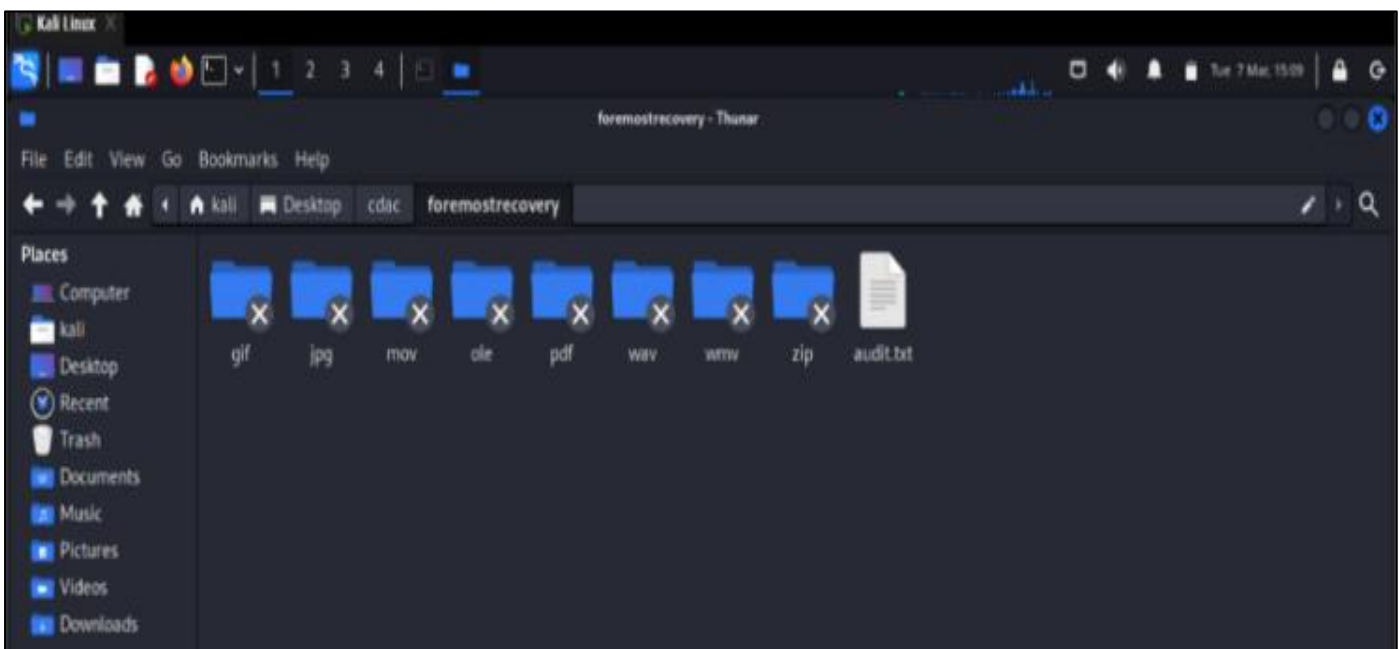


Fig 6 Carved File Types

The *audit.txt* file located within the "foremostrecovery" directory provides a list view of the items recovered by Foremost, along with their corresponding sizes and file offset locations. This information can be used to identify and further analyze the recovered files:

```

Kali Linux x
~/Desktop/cdac/foremostrecovery/audit.txt [Read Only] - Mousepad
File Edit Search View Document Help
1 Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Mikus
2 Audit File
3
4 Foremost started at Tue Mar 7 10:06:44 2023
5 Invocation: foremost -i 11-carve-fat.dd -o foremostrecovery
6 Output directory: /home/kali/Desktop/cdac/foremostrecovery
7 Configuration file: /etc/foremost.conf
8
9 File: 11-carve-fat.dd
10 Start: Tue Mar 7 10:06:44 2023
11 Length: 61 MB (64979456 bytes)
12
13 Num      Name (bs=512)      Size      File Offset      Comment
14
15 0:      00019717.jpg      29 KB     10095104
16 1:      00019777.jpg      433 KB    10125824
17 2:      00020645.jpg      96 KB     10570240
18 3:      00020841.gif      5 KB      10670592      (88 x 31)
19 4:      00000321.wmv      7 MB      104352
20 5:      00021929.wmv      1012 KB   11227648
21 6:      00020853.mov      537 KB    10676736
22 7:      00016021.wav      311 KB    8202752
23 8:      00000281.ole      20 KB     143872
24 9:      00016693.ole      24 KB     8546816
25 10:     00023957.ole      6 MB      12265984
26 11:     00023981.zip      77 KB     12278272
27 12:     00016741.pdf      1 MB      8571392
28 13:     00019477.pdf      119 KB    9972224      (PDF is Linearized)
29 Finish: Tue Mar 7 10:06:48 2023
30
31 14 FILES EXTRACTED
32
    
```

Fig 7 Carved Results as Displayed by Audit.Txt File

As you scroll down the audit.txt file, you will come across a summary of all the files discovered during the carving process.

In this particular case, the first three items listed in the audit.txt file are .jpg picture files, which can be found in the "jpg" sub-folder within the "foremostrecovery" output directory:

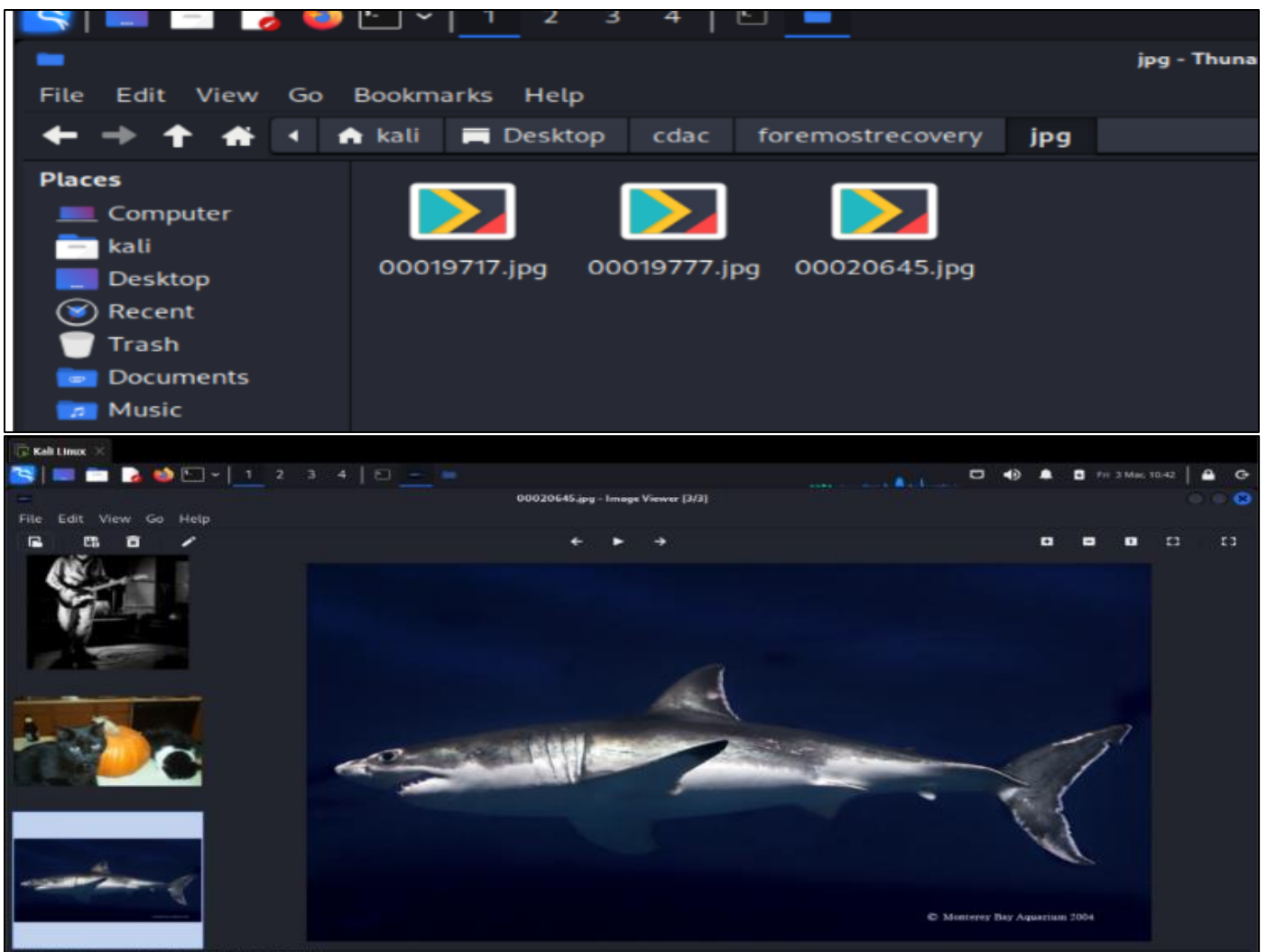


Fig 8 Snippet of Recovered Images

Foremost is a powerful tool for file and data recovery, and its effectiveness can vary depending on the size of the disk or image being used. File carving can be a time-consuming process, especially when dealing with larger images. However, if you know the specific file type you're looking for, you can save time by using the `-t` option to search for that file type only. Additionally, you can use the `-t` option followed by the file extension to specify distinct file types and speed up the search process. It is worth noting that Foremost supports a wide range of file formats, including but not limited to .jpg, .gif, .png, .bmp, .avi, .mpg, .wav, .mov, .pdf, .doc, .zip, and .mp4, as documented in the manual accessed via the "man foremost" command.

B. Scalpel Tool:

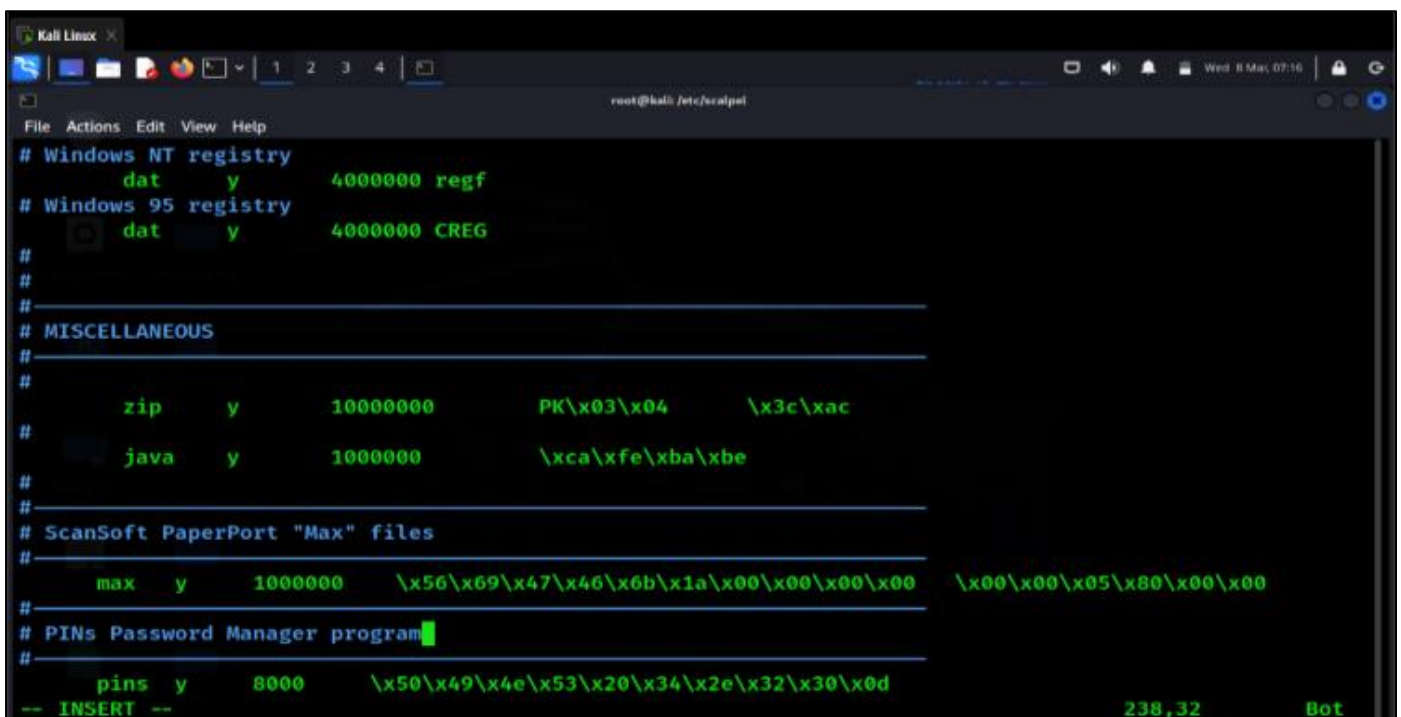
➤ Using Scalpel for Data Carving:

Scalpel was developed as an enhancement on an earlier version of first. Scalpel is designed to overcome the significant CPU and Memory use difficulties that occur when carving data using first.

➤ Specifying File Types in Scalpel:

With the Scalpel configuration file, the investigator must identify file types of interest, as opposed to first. This file is known as *scalpel.conf* which is present in the *etc/scalpel/* directory. To specify the file types, the investigator must remove the comments at the opening of the line containing the file type, as all supported file types are commented out at the start with a hashtag.

The Scalpel configuration file (*scalpel.conf*) is shown below with all file types commented away. Take Serious note that each line begins with a hashtag:



```

# Windows NT registry
  dat    y    4000000  regf
# Windows 95 registry
  dat    y    4000000  CREG
#
#
# MISCELLANEOUS
#
  zip    y    10000000  PK\x03\x04  \x3c\xac
#
  java   y    1000000    \xca\xfe\xba\xbe
#
# ScanSoft PaperPort "Max" files
#
  max    y    1000000    \x56\x69\x47\x46\x6b\x1a\x00\x00\x00\x00  \x00\x00\x05\x80\x00\x00
#
# PINs Password Manager program
#
  pins   y    8000    \x50\x49\x4e\x53\x20\x34\x2e\x32\x30\x0d
-- INSERT --
238,32 Bot

```

Fig 9 All File Types that were not Selected as Shown by the #.

To teach Scalpel to hunt for these specific file types, we removed the hashtags at the beginning of multiple lines. This also reduces the time required to search for all available file types. Scalpel is seen in the screenshot below looking for GIF and JPG files after the comments have been removed. Make sure you finish this step before specifying the image to be carved. If this is not done, an error notification is sent to the investigator as a reminder. Once we've done all of the required changes, we can go to the GUI menu and then just select Scalpel to start carving.

➤ Using Scalpel for File Carving:

When we've updated the *scalpel.conf* file to include file types and saved it, we can initiate Scalpel by clicking the Display Programs button in the sidebar of Linux and putting scalpel into the search box at the top of the page, as seen in the following image. To Initiate process, click on the scalpel box:

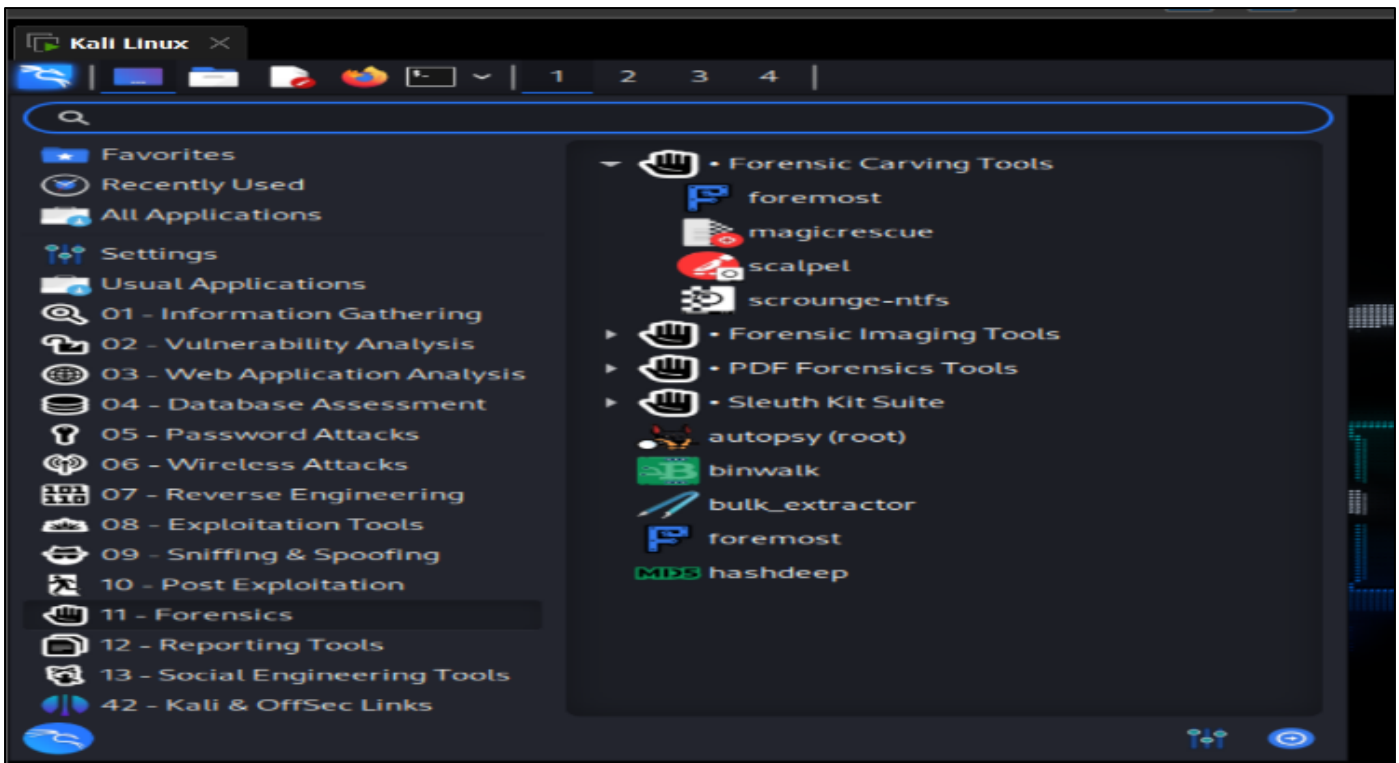


Fig 10 Scalpel Icon

When the programme is opened, a Terminal window displays, indicating the version number as (1.60), the developer (Golden G. Richard III), and the information that it is based on version 0.69. Scalpel's syntax and additional arguments are also shown, as with foremost tool:

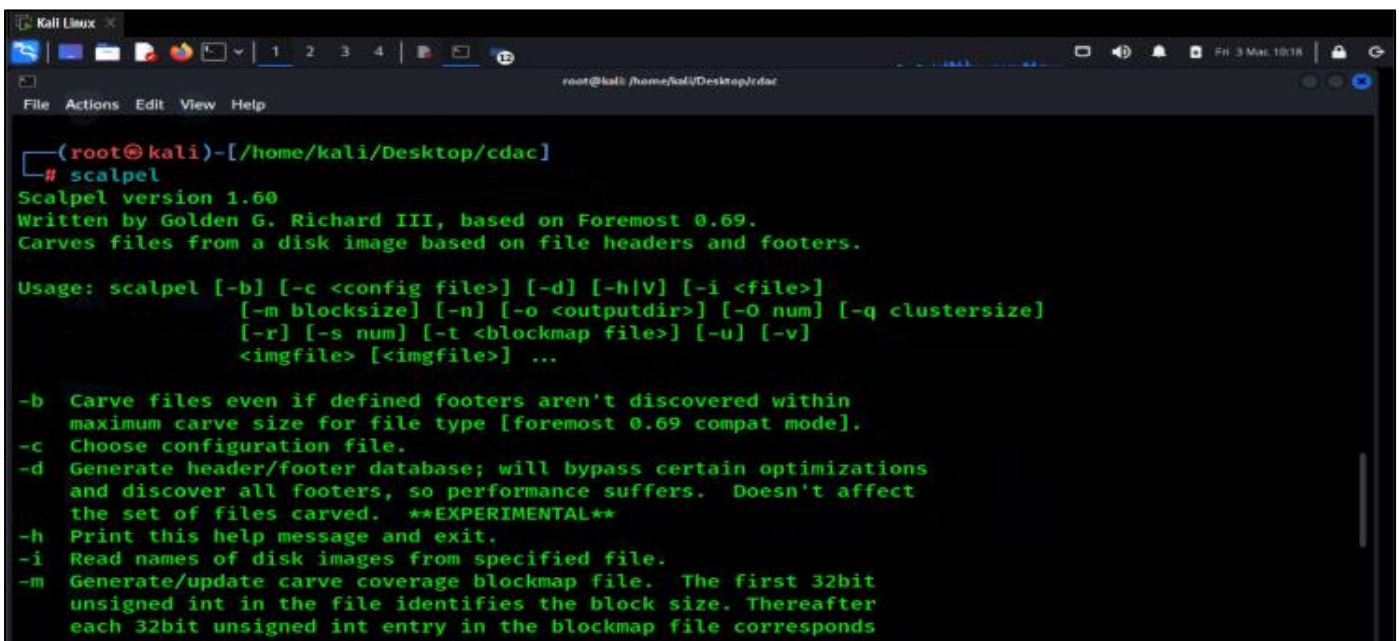


Fig 11 Scalpel Options

The same picture that was used for carving with foremost (*11-carve-fat.dd*) was used for this. The input file and output folder, as with foremost, must be given. Scalpel's various options and switches are listed with *scalpel -h*.

- *Scalpel Used the Command Line:*
- *Scalpel -O Recovery/ 11-Carve-Fat.Dd*
 Scalpel creates a carve list containing the file type, header and footer information, and the number of files sliced, as seen in the snapshot below:

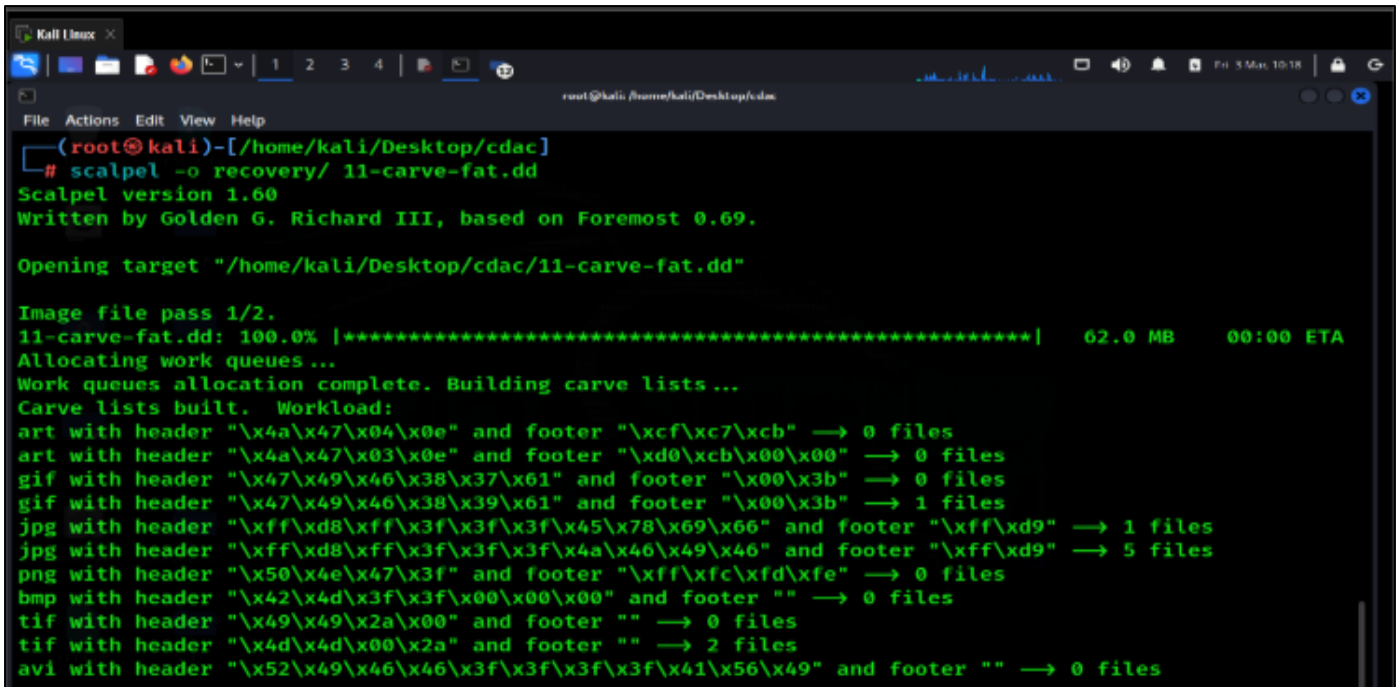


Fig 12 Scalpel Carving Process

As given in the preceding screenshot, Scalpel has now successfully completed all Data carving processes.

➤ *Viewing the Results of Scalpel:*

We may now view the carved files by navigating to the output folder, designated recovery. Scalpel output results are identical to the first, with both output directories including separate subfolders including carved files, as well as an audit.txt file holding information on the findings.

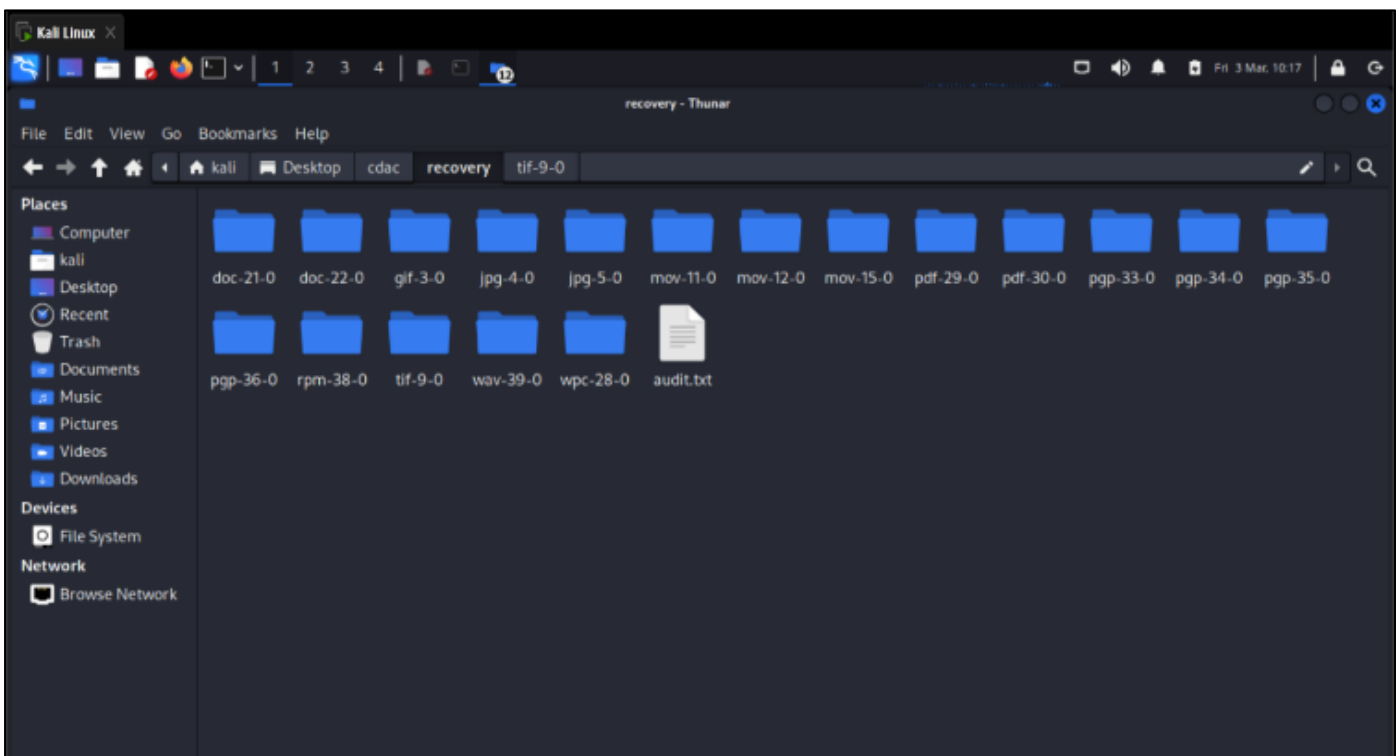


Fig 13 Scalpel Output Folder

Within the *mov-12-0* folder, we can clearly see two *.mov* files.

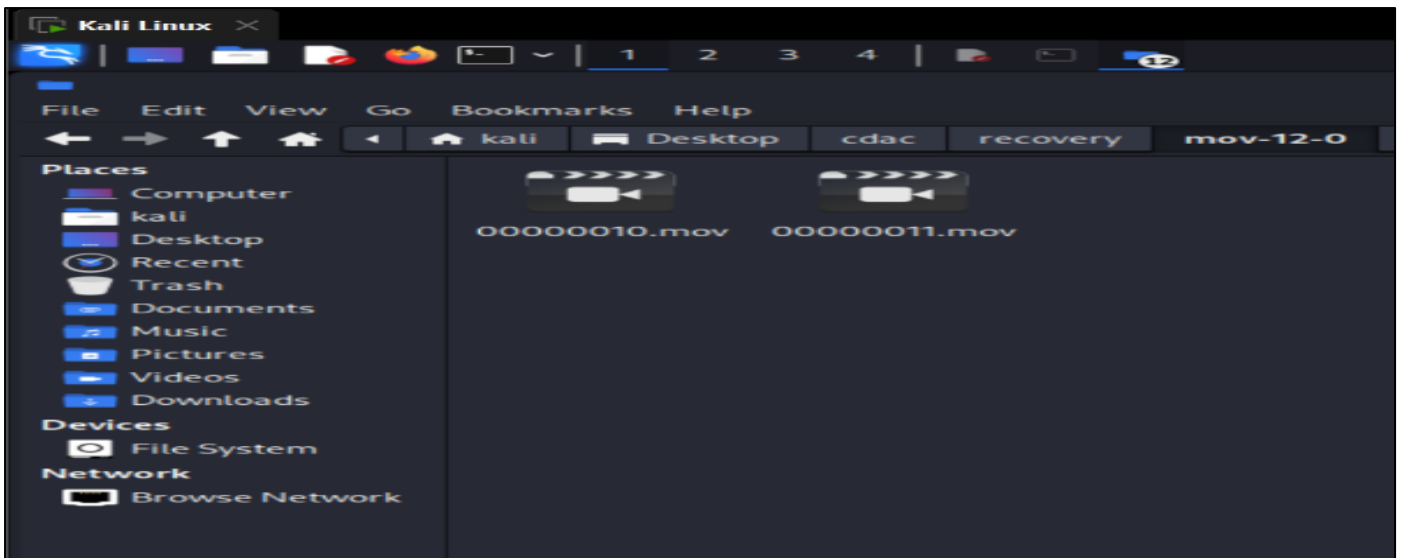


Fig 14 Carved .Mov Files

Within the *pdf-30-0* folder, we can now see two .pdf files.

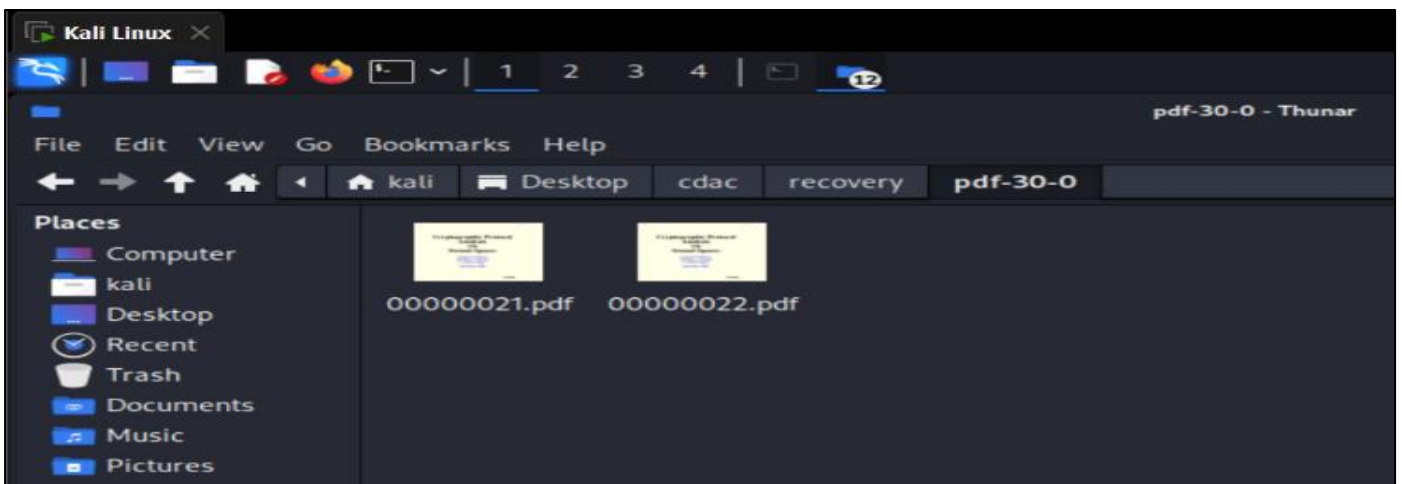


Fig 15 Carved .Pdf Files

The below screenshot shows a snippet of the *audit.txt* file, showing information of the carved files:

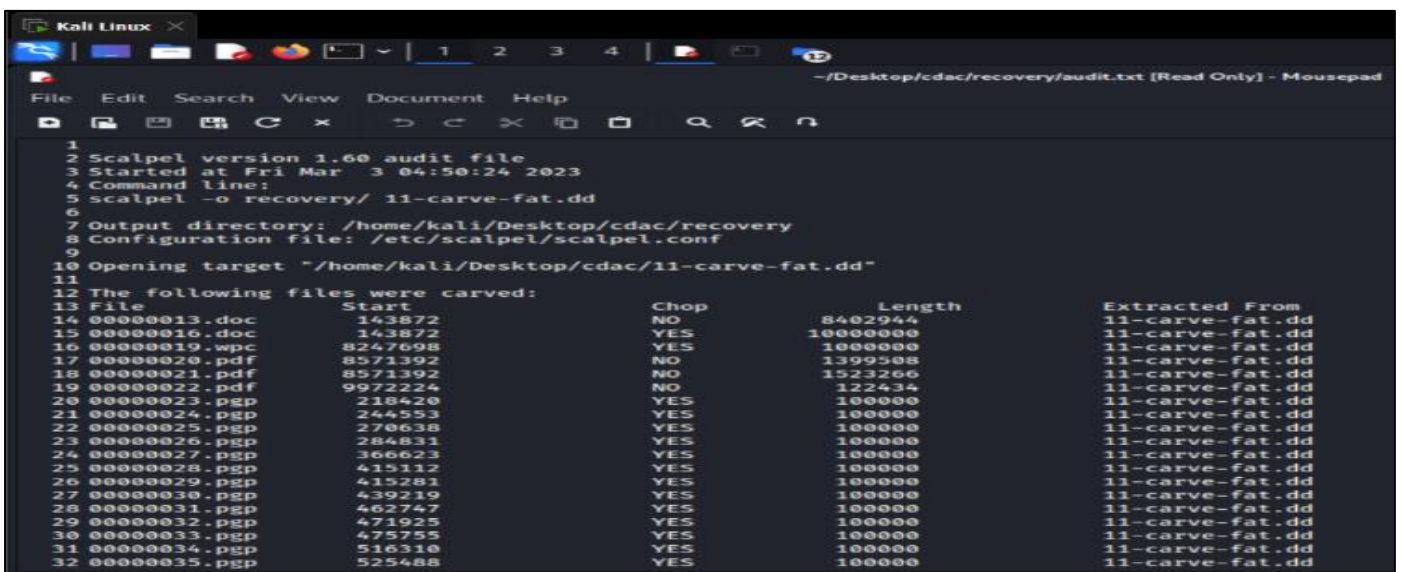


Fig 16 Scalpel Output Results Listed within the Audit.Txt File

Now here Scalpel gave us a precise list of all the files in the audit.txt file, as well as storing every file type in its own folder within the recovery folder. You can carve either the first or second file.

➤ *Comparing Foremost and Scalpel:*

Compare the carved files discovered by both foremost and Scalpel, regardless of the fact that Scalpel extracted more files than foremost. The filenames offered by both programmes, however, are not the original filenames, and there may be duplicates of carved files in certain instances because many files are fragmented and appear to be separate files. Now try exploring the files in both the primary and Scalpel output folders and performing your own comparative study to decide which tool was more productive.

C. *Bulk Extractor:*

Although foremost and Scalpel can extract images, music, video, and compressed files, bulk extractor recovers a wide range of additional data that may be quite useful in analyses. Although bulk extractor can recover and carve Image, video, and document files, it can also carve and extract the following data:

- *Credit Card Numbers*
- *Email ID*
- *Website Urls*
- *Webpage Information*
- *Social Media Accounts and Information*

➤ *Forensic Test Image Used in Bulk_Extractor:*

We will use a readily available evidence file called *terry-workusb-2009-12-11.E01*. This file permits the use of forensic evidence images in forensic research.

➤ *Using Bulk_Extractor:*

To receive a list of widely used arguments and options, start bulk extractor by entering command bulk extractor -h.

```
(root@kali)-[~/Desktop/cdac]
# bulk_extractor
imagefile not provided
bulk_extractor version 2.0.0: A high-performance flexible digital forensics program.
Usage:
bulk_extractor [OPTION... ] image_name

-A, --offset_add arg          Offset added (in bytes) to feature locations (default: 0)
-b, --banner_file arg        Path of file whose contents are prepended to top of all feature files
-C, --context_window arg     Size of context window reported in bytes (default: 16)
-d, --debug arg              enable debugging (default: 1)
-D, --debug_help             help on debugging
-E, --enable_exclusive arg   disable all scanners except the one specified. Same as -x all -E scanner.
-e, --enable arg             enable a scanner (can be repeated)
-x, --disable arg            disable a scanner (can be repeated)
-f, --find arg                search for a pattern (can be repeated)
-F, --find_file arg          read patterns to search from a file (can be repeated)
-G, --pagesize arg           page size in bytes (default: 16777216)
-g, --margin_size arg        margin size in bytes (default: 4194304)
-j, --threads arg            number of threads (default: 4)
-J, --no_threads             read and process data in the primary thread
```

Fig 17 Available Options in Bulk Extractor

Like foremost and Scalpel, Bulk extractor has a straightforward syntax that calls for an output folder (-o) and the forensic picture. As previously indicated, for this experiment, we will extract data from the *terry-workusb-2009-12-11.E01* image and store the results to a folder entitled bulk-output.

• *The syntax is as follows:*

• *Bulk_Extractor -O Bulkrecovery Terry-Work-Usb-2009-12-11.E01*

Data extraction from huge files might be time-consuming. Nevertheless, after running the above command, a status update is provided, as shown in the following screenshot:

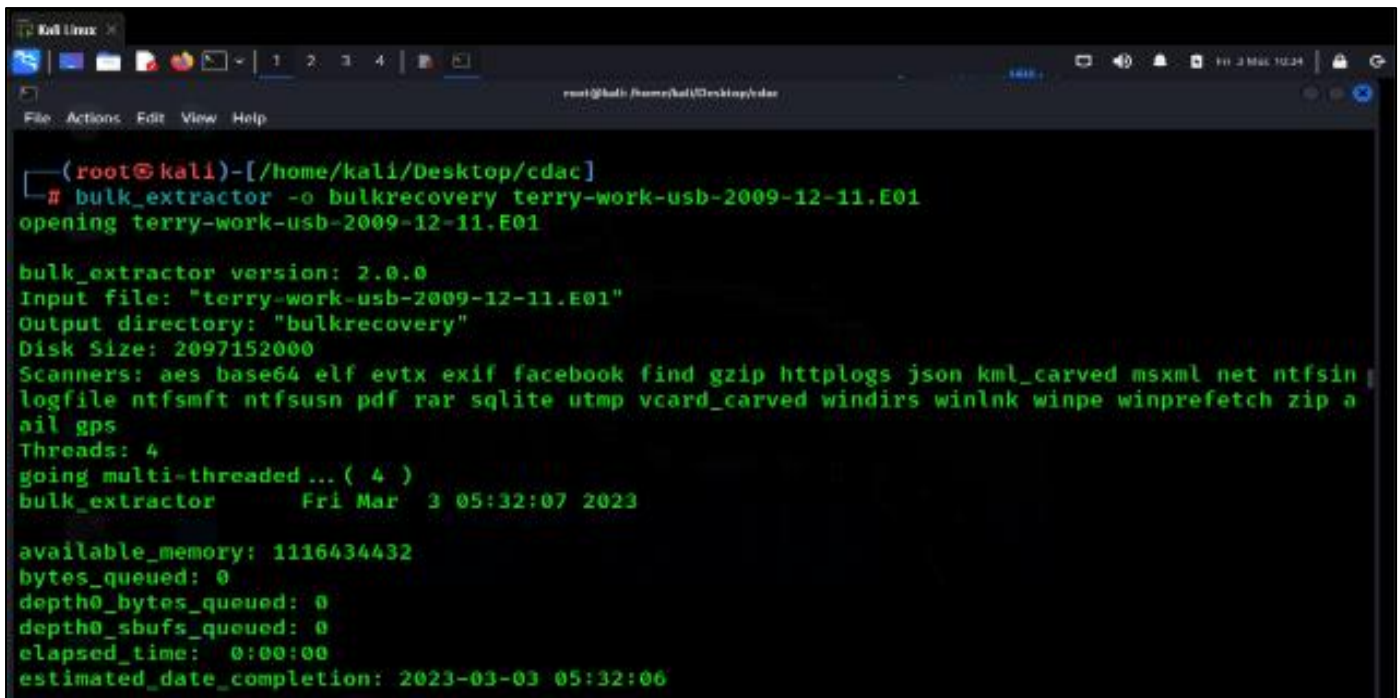


Fig 18 Bulk Extractor Process Completion

When all threads have completed, the bulk extractor displays an overview of the operation as well as some outcomes. Bulk extractor, as demonstrated in the accompanying image, displays the MD5 hash, the total number of MB processed, and even announces the detection of three email characteristics. In the next section, we'll look more closely at the findings.

➤ *Viewing the Results of Bulk_Extractor:*

We may browse a list of directories while examining the output and discoveries of the bulk extractor by executing `ls -l` in the Terminal. As we can see, the bulk extractor created the *bulkrecovery* folder. We can now examine the contents of our output folder by typing `ls -l bulkrecovery`.

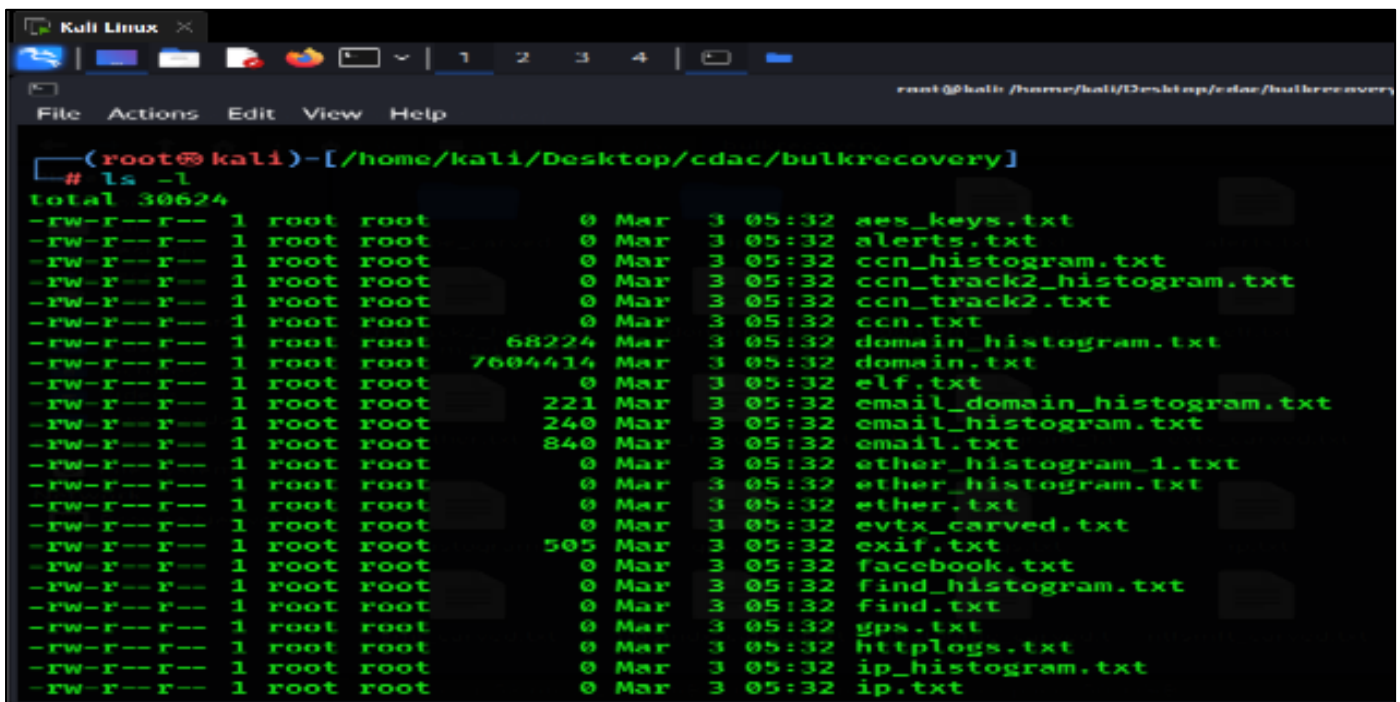


Fig 19 Carved Files

Keep in mind that not all of the text files mentioned below will contain data. Data will be present only for those with numbers larger than 0 to the left of the text filenames. If we go to the output folder, we can see all of the extracted data in the various text files:

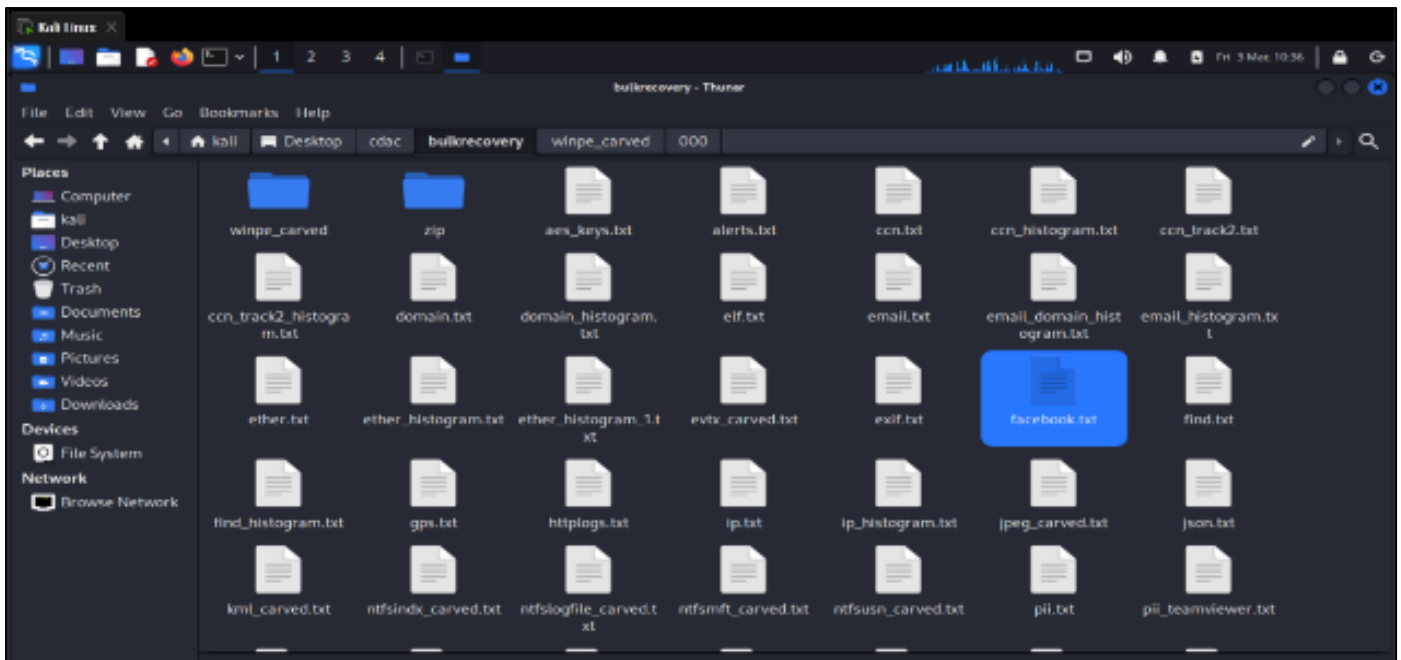


Fig 20 Carved .Txt Files

III. FUTURE SCOPE

To gain a better understanding of this subject, it would be beneficial to use a broader range of open-source forensic tools to compare and quantify their results across different disk image formats such as dd, split dd, and EWF. By analyzing the number of false positives generated by these tools, it would be possible to determine their effectiveness in general and with respect to specific file types. However, due to the complexity of this task, it is beyond the scope of the current paper to undertake this level of research. In the future, it may be possible to modify the framework to allow both Foremost and Scalpel to be used on Linux and Windows machines. This would necessitate a partial rewrite of the script to account for the different naming conventions on the two operating systems and would require comprehensive re-testing on all versions of Python and platforms.

IV. CONCLUSION

In this analysis, we learned about file recovery and data extraction using popular open-source tools in Kali Linux. We first performed file carving using the very impressive foremost, which searched an entire image for supported file types within the file's headers and footers. We then did the same using recover jpg and the newer Scalpel, but had to make a slight modification by selecting the file types we wished to carve. Both foremost and Scalpel presented us with an audit.txt file summarizing the carve list and its details, along with subfolders containing the actual evidence. bulk_extractor is a wonderful tool that carves data and also finds useful information, such as email addresses, visited URLs, Facebook URLs, credit card numbers, and a variety of other information. bulk_extractor is great for investigations requiring file recovery and carving, together with either foremost or Scalpel, or even both.

REFERENCES

- [1]. www.cyberforensics.in
- [2]. <https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/cftt-technical-0>
- [3]. Download 11-carve-fat.zip (Digital Forensic Tool Testing) (sourceforge.net)
- [4]. Digital Corpora: corpora/scenarios/2009-m57-patents/drives-redacted/
- [5]. Esra'a Alshammary, Ali Hadi, "Reviewing and Evaluating Existing File Carving Techniques for JPEG Files".IEEE, 2016.
- [6]. G. G. Richard and V. Roussev, "Scalpel: A frugal, high performance file carver". 5th Annual Digital Forensic Research Workshop, 2005.
- [7]. "A Survey on Multimedia File Carving", (IJCSSES) Vol.6, No.6, December 2015.
- [8]. Gareth Palmieri, Shahrzad Zargari, "Using Open Source Forensic Carving tools on split dd and EWF files.", Sheffield Hallam University Research Archive, 2017
- [9]. Nurhayati, Nurul Fikri, "The Analysis of File Carving Process Using Photorec and Foremost.", 2017
- [10]. S. L. Garfinkel, "Digital media triage with bulk data analysis and bulk-extractor," *Comput. Secur.*, vol. 32, pp. 56–72, 2013.
- [11]. Simson Garfinkel, "Carving contiguous and fragmented files with fast object validation". The Digital Forensic Research Conference, 2007.