LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

Experiment No.

A.1 Aim:
To Perform Forensic Analysis of deleted files.
A.2 Prerequisite:
Data recovery, Digital Forensic, kali Linux

A.3 Outcome:
After successful completion of this experiment students will be able to
1. Appreciate foremost as a forensic tool to recover the data.
2. Explorer kali Linux as penetration testing Operating system.

A.4 Theory:

Virtual Machine: With a virtual machine, the sandbox is isolated from the underlying physical hardware but has access to the installed operating system. Virtualized environment. Usually, a sandbox is on a virtual machine so that it has no access to physical resources but can access virtualized hardware.

Kali Linux: Kali Linux is a Debian -derived Linux distribution designed for digital forensics and penetration testing. It is maintained and funded by Offensive Security.

Forensic: Digital forensics is a branch of forensic science encompassing the recovery, investigation, examination, and analysis of material found in digital devices, often in relation to mobile devices and computer crime.

Foremost is a digital forensic application that is used to recover lost or deleted files. Foremost can recover the files for hard disk, memory card, pen drive, and another mode of memory devices easily. It can also work on the image files that are being generated by any other Application. It is a free command-line tool that is pre-installed in Kali Linux. This tool comes pre-installed in Kali Linux. Foremost is an especially useful software that is used to recover the deleted files, if some files are deleted accidentally or in any case files are deleted. You can recover the deleted files from foremost only if the data in the device is not overridden, which means after deleting the files no more data is added to the storage device because in that case data may be overridden and the chances of recovery also get reduced and data must get corrupted.

Installing the Foremost Tool:
Use the following command to install this tool in any Debian based Linux Operating System or in any other Operating System using the APT package manager.

sudo apt install foremost

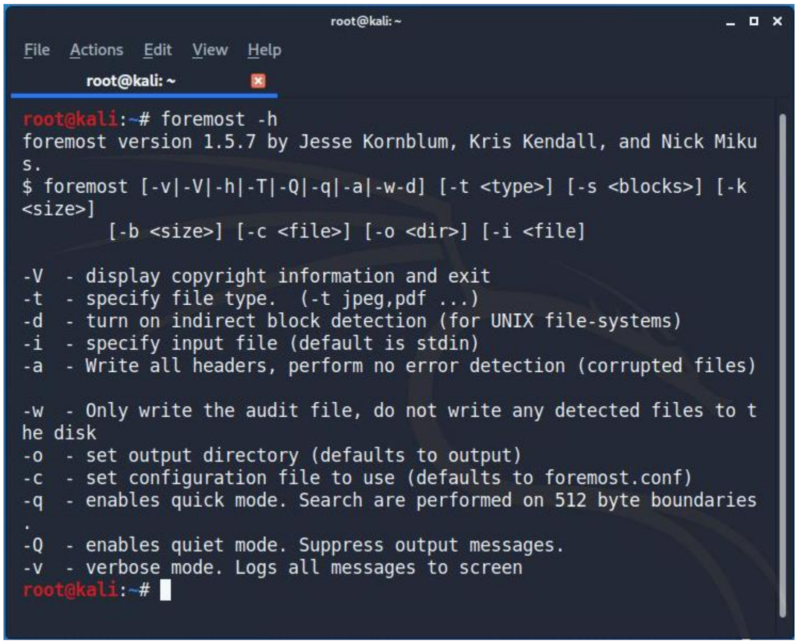
Use the following command to install this tool using dnf package manager

sudo dnf install foremost

Use the following command to install this tool using Pacman package manager or in Arch Linux.

sudo pacman -S foremost

Syntax:
foremost [options]

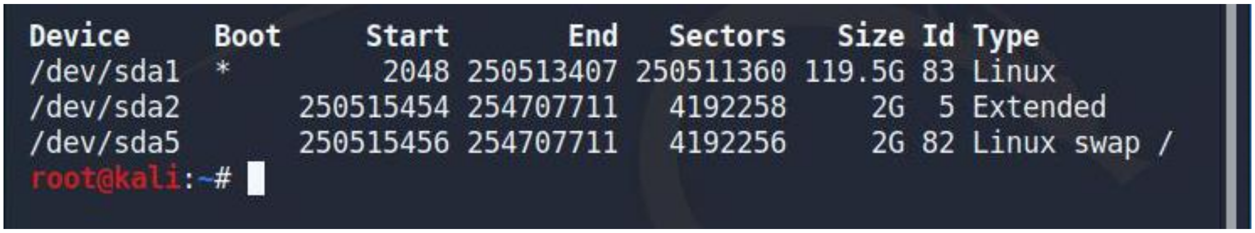


```
root@kali:~# foremost -h
foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Miku
s.
$ foremost [-v|-V|-h|-T|-Q|-q|-a|-w-d] [-t <type>] [-s <blocks>] [-k
<size>]
        [-b <size>] [-c <file>] [-o <dir>] [-i <file]

-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 t
he 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 to screen
root@kali:~# 
```

Here you can check the options available and their functions. Let us now see how to recover deleted files using foremost:

Recovering from USB/Hard Disk:

- Connect the External memory storage with the system.
- First, you need to know the path of your external memory device, for that use the command fdisk -l



```
Device     Boot     Start       End   Sectors   Size Id Type
/dev/sda1  *         2048 250513407 250511360 119.5G 83 Linux
/dev/sda2       250515454 254707711   4192258     2G  5 Extended
/dev/sda5       250515456 254707711   4192256     2G 82 Linux swap /
root@kali:~# 
```

- After copying the device path, now we must recover the files from that device.

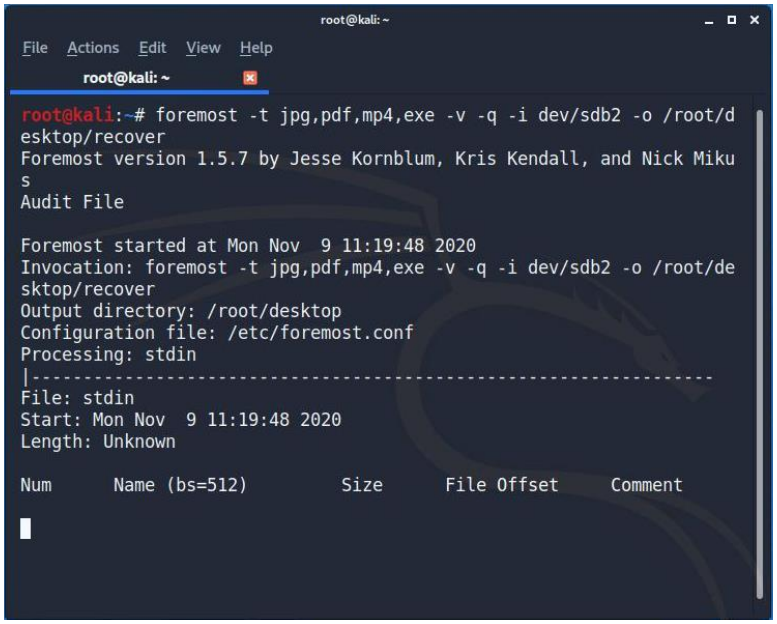
Use the options available by the “foremost -h” command.

For example :

foremost -t jpg,pdf,mp4,exe -v -q -i /dev/sdb2 -o /root/desktop/recover

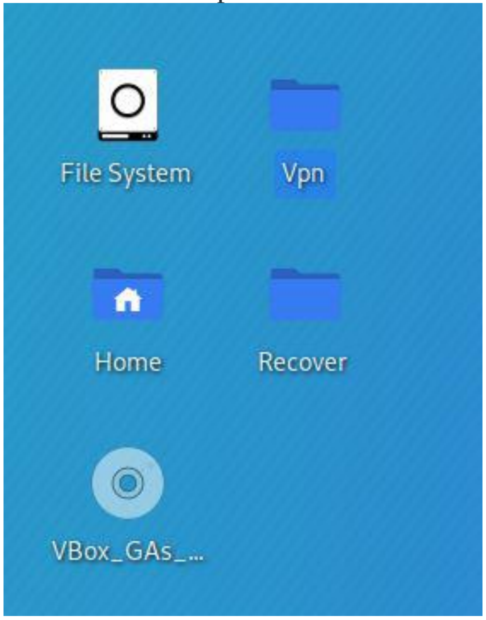
Here uses this command to recover the data from the device.

- **-t**: It is the type of files we want to recover. Here I want to recover jpg, pdf,mp4, and exe files.
- **-q**: It is a quick scan for the device
- **-i**: It means the input as in this case external memory.
- **-o**: It is the output folder, where to save the recovered files.



```
root@kali:~# foremost -t jpg,pdf,mp4,exe -v -q -i dev/sdb2 -o /root/d
esktop/recover
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, and Nick Miku
s
Audit File

Foremost started at Mon Nov  9 11:19:48 2020
Invocation: foremost -t jpg,pdf,mp4,exe -v -q -i dev/sdb2 -o /root/de
sktop/recover
Output directory: /root/desktop
Configuration file: /etc/foremost.conf
Processing: stdin
|------------------------------------------------------------------
File: stdin
Start: Mon Nov  9 11:19:48 2020
Length: Unknown

Num      Name (bs=512)         Size      File Offset     Comment

```

Hereafter running this command, all the files will be saved in the folder name as mentioned. Here you can see the folder recover on desktop and all the files will be stored here.



PART B

(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. A022	Name: Kartik Padave
Class: B.Tech	Batch: 1
Date of Experiment:	Date of Submission:
Grade:	

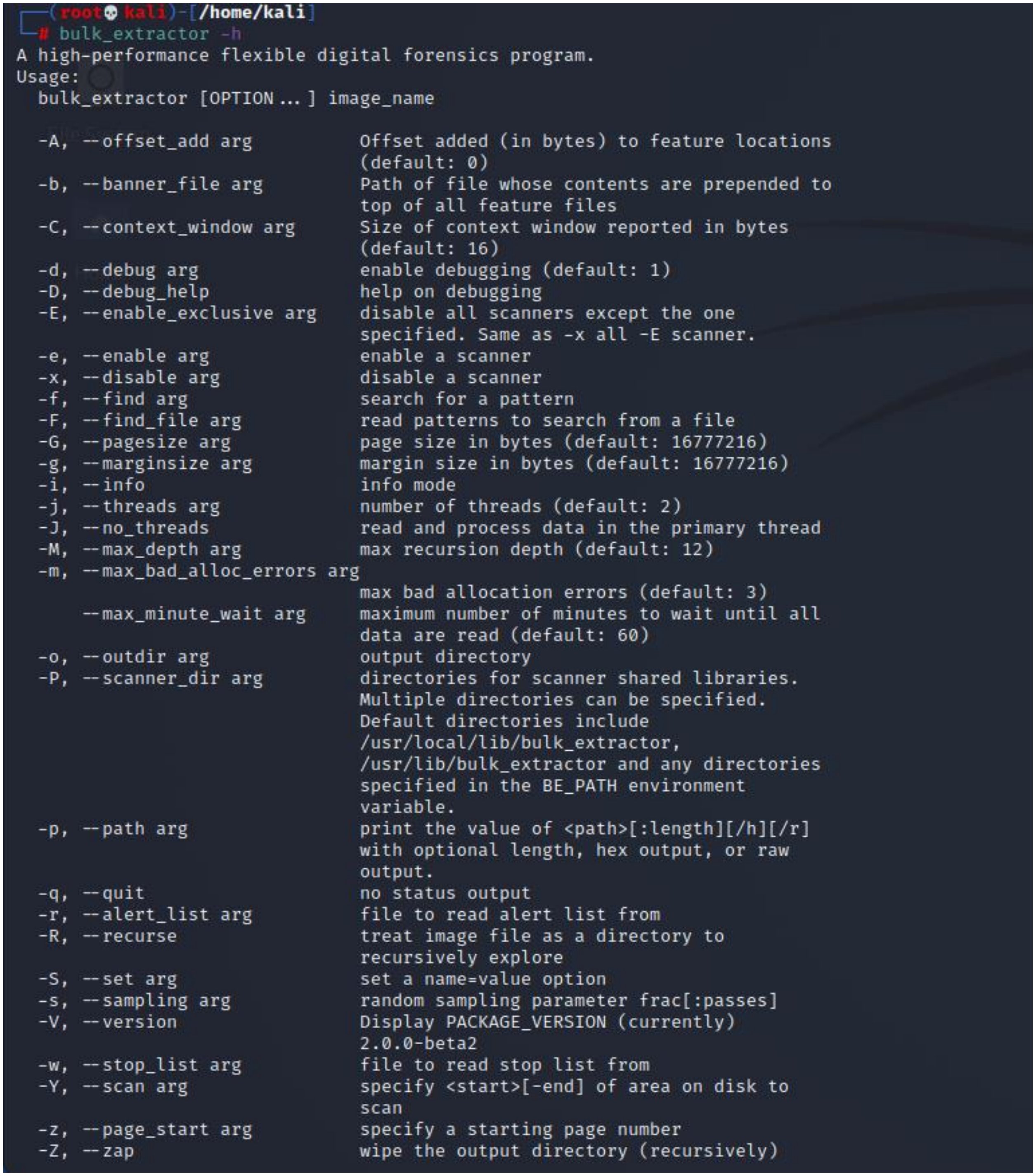
B.1 Software installation issues faced:

B.2 Input and Output:

(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output at the right time of time. Students will be graded accordingly.)

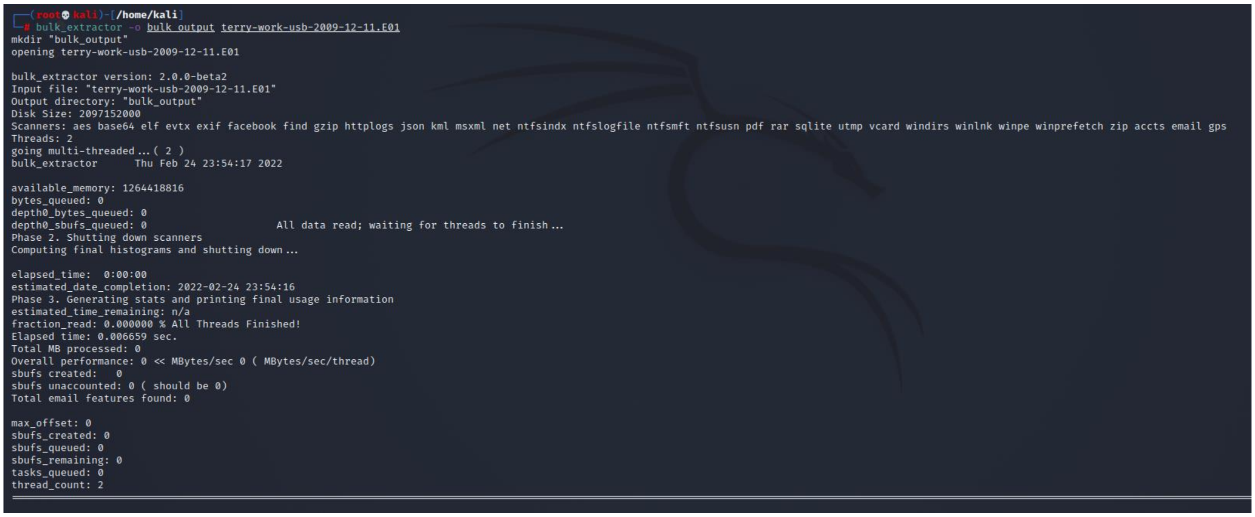
Input and Output

1. Bulk_extractor -h



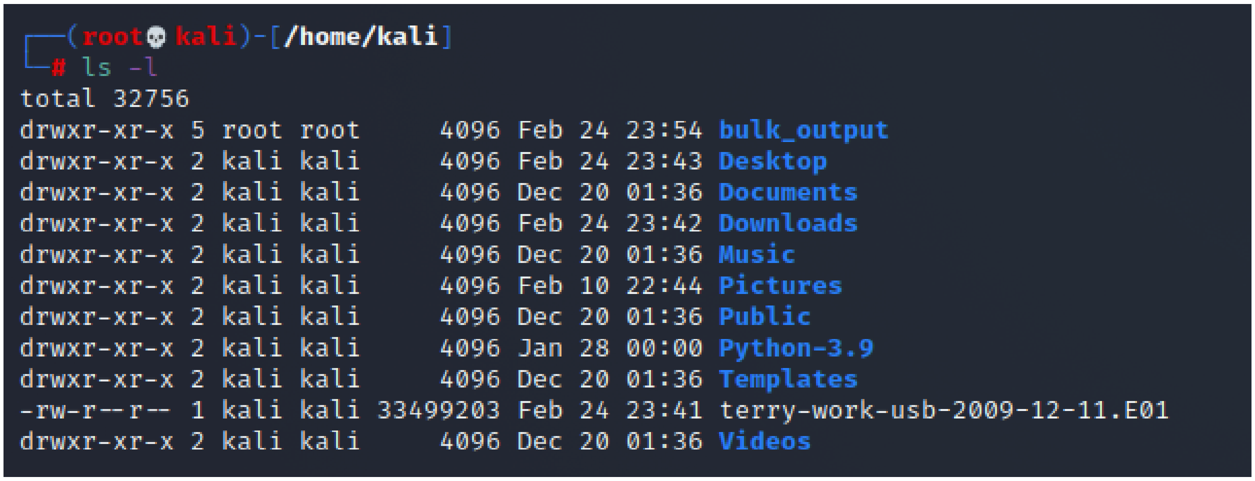
```
┌──(root💀kali)-[/home/kali]
└─# bulk_extractor -h
A high-performance flexible digital forensics program.
Usage:
  bulk_extractor [OPTION...] image_name

  -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
  -x, --disable arg             disable a scanner
  -f, --find arg                search for a pattern
  -F, --find_file arg           read patterns to search from a file
  -G, --pagesize arg            page size in bytes (default: 16777216)
  -g, --marginsize arg          margin size in bytes (default: 16777216)
  -i, --info                    info mode
  -j, --threads arg             number of threads (default: 2)
  -J, --no_threads              read and process data in the primary thread
  -M, --max_depth arg           max recursion depth (default: 12)
  -m, --max_bad_alloc_errors arg
                                max bad allocation errors (default: 3)
      --max_minute_wait arg     maximum number of minutes to wait until all
                                data are read (default: 60)
  -o, --outdir arg              output directory
  -P, --scanner_dir arg         directories for scanner shared libraries.
                                Multiple directories can be specified.
                                Default directories include
                                /usr/local/lib/bulk_extractor,
                                /usr/lib/bulk_extractor and any directories
                                specified in the BE_PATH environment
                                variable.
  -p, --path arg                print the value of <path>[:length][/h][/r]
                                with optional length, hex output, or raw
                                output.
  -q, --quit                    no status output
  -r, --alert_list arg          file to read alert list from
  -R, --recurse                 treat image file as a directory to
                                recursively explore
  -S, --set arg                 set a name=value option
  -s, --sampling arg            random sampling parameter frac[:passes]
  -V, --version                 Display PACKAGE_VERSION (currently)
                                2.0.0-beta2
  -w, --stop_list arg           file to read stop list from
  -Y, --scan arg                specify <start>[-end] of area on disk to
                                scan
  -z, --page_start arg          specify a starting page number
  -Z, --zap                     wipe the output directory (recursively)
```

2. bulk_extractor -o bulk_output terry-work-usb-2009-12-11.E01



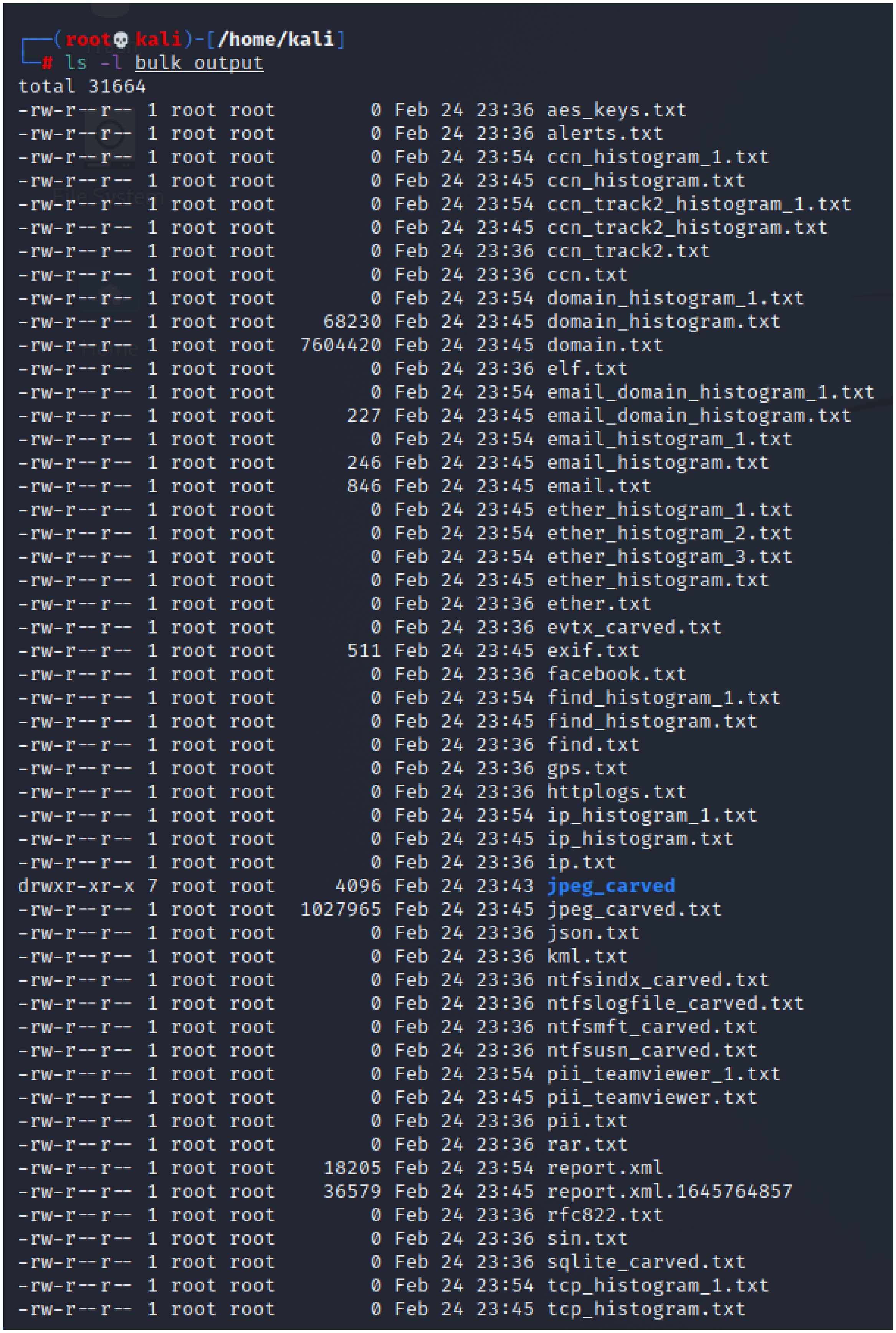
```
┌──(root💀 kali)-[/home/kali]
└─# bulk_extractor -o bulk_output terry-work-usb-2009-12-11.E01
mkdir "bulk_output"
opening terry-work-usb-2009-12-11.E01

bulk_extractor version: 2.0.0-beta2
Input file: "terry-work-usb-2009-12-11.E01"
Output directory: "bulk_output"
Disk Size: 2097152000
Scanners: aes base64 elf evtx exif facebook find gzip httplogs json kml msxml net ntfsindx ntfslogfile ntfsmft ntfsusn pdf rar sqlite utmp vcard windirs winlnk winpe winprefetch zip accts email gps
Threads: 2
going multi-threaded...( 2 )
bulk_extractor      Thu Feb 24 23:54:17 2022

available_memory: 1264418816
bytes_queued: 0
depth0_bytes_queued: 0
depth0_sbufs_queued: 0                    All data read; waiting for threads to finish...
Phase 2. Shutting down scanners
Computing final histograms and shutting down...

elapsed_time:  0:00:00
estimated_date_completion: 2022-02-24 23:54:16
Phase 3. Generating stats and printing final usage information
estimated_time_remaining: n/a
fraction_read: 0.000000 % All Threads Finished!
Elapsed time: 0.006659 sec.
Total MB processed: 0
Overall performance: 0 << MBytes/sec 0 ( MBytes/sec/thread)
sbufs created:   0
sbufs unaccounted: 0 ( should be 0)
Total email features found: 0

max_offset: 0
sbufs_created: 0
sbufs_queued: 0
sbufs_remaining: 0
tasks_queued: 0
thread_count: 2
```

3. ls -l

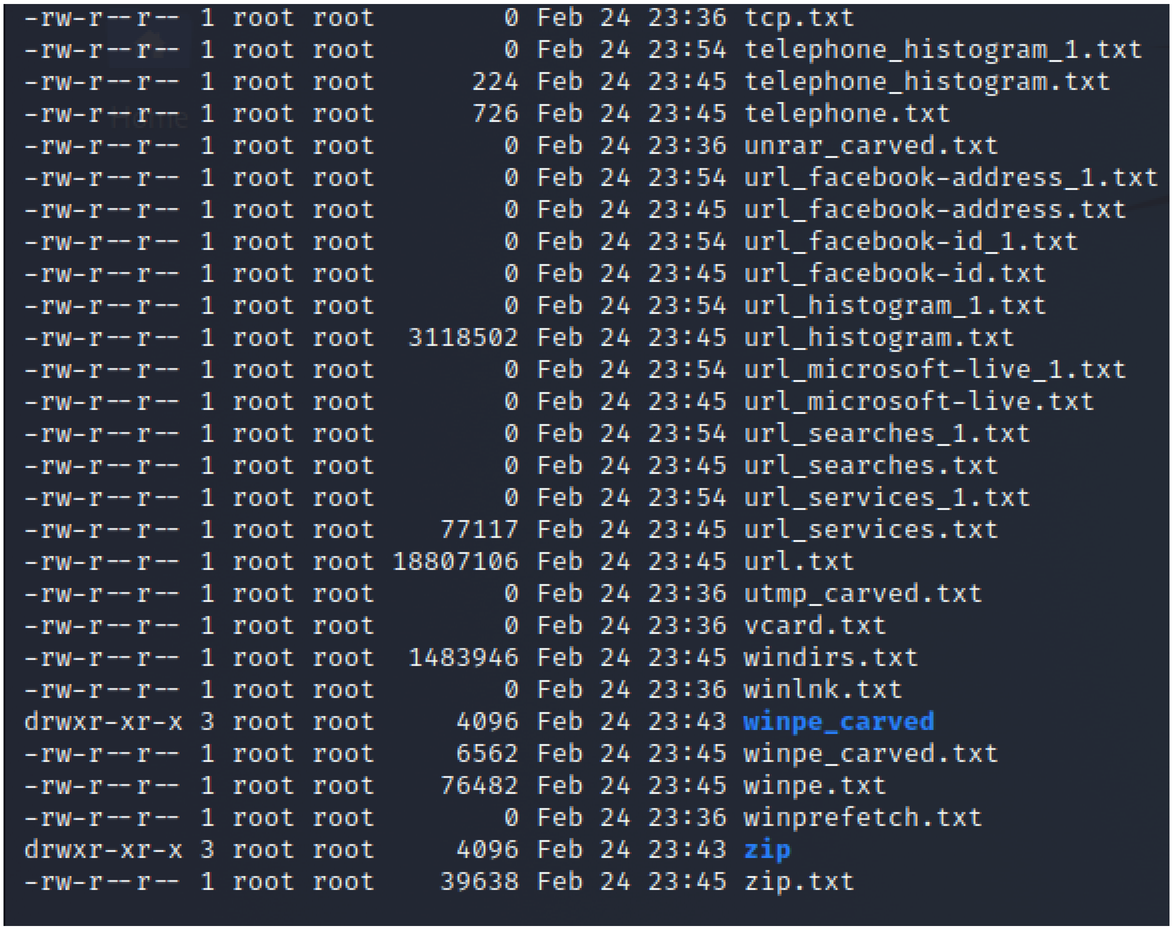


```
┌──(root💀 kali)-[/home/kali]
└─# ls -l
total 32756
drwxr-xr-x 5 root root     4096 Feb 24 23:54 bulk_output
drwxr-xr-x 2 kali kali     4096 Feb 24 23:43 Desktop
drwxr-xr-x 2 kali kali     4096 Dec 20 01:36 Documents
drwxr-xr-x 2 kali kali     4096 Feb 24 23:42 Downloads
drwxr-xr-x 2 kali kali     4096 Dec 20 01:36 Music
drwxr-xr-x 2 kali kali     4096 Feb 10 22:44 Pictures
drwxr-xr-x 2 kali kali     4096 Dec 20 01:36 Public
drwxr-xr-x 2 kali kali     4096 Jan 28 00:00 Python-3.9
drwxr-xr-x 2 kali kali     4096 Dec 20 01:36 Templates
-rw-r--r-- 1 kali kali 33499203 Feb 24 23:41 terry-work-usb-2009-12-11.E01
drwxr-xr-x 2 kali kali     4096 Dec 20 01:36 Videos
```

4. ls -l bulk output



```
┌──(root💀kali)-[/home/kali]
└─# ls -l bulk_output
total 31664
-rw-r--r-- 1 root root       0 Feb 24 23:36 aes_keys.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 alerts.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 ccn_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 ccn_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 ccn_track2_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 ccn_track2_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ccn_track2.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ccn.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 domain_histogram_1.txt
-rw-r--r-- 1 root root   68230 Feb 24 23:45 domain_histogram.txt
-rw-r--r-- 1 root root 7604420 Feb 24 23:45 domain.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 elf.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 email_domain_histogram_1.txt
-rw-r--r-- 1 root root     227 Feb 24 23:45 email_domain_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 email_histogram_1.txt
-rw-r--r-- 1 root root     246 Feb 24 23:45 email_histogram.txt
-rw-r--r-- 1 root root     846 Feb 24 23:45 email.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 ether_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 ether_histogram_2.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 ether_histogram_3.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 ether_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ether.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 evtx_carved.txt
-rw-r--r-- 1 root root     511 Feb 24 23:45 exif.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 facebook.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 find_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 find_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 find.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 gps.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 httplogs.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 ip_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 ip_histogram.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ip.txt
drwxr-xr-x 7 root root    4096 Feb 24 23:43 jpeg_carved
-rw-r--r-- 1 root root 1027965 Feb 24 23:45 jpeg_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 json.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 kml.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ntfsindx_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ntfslogfile_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ntfsmft_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 ntfsusn_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 pii_teamviewer_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 pii_teamviewer.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 pii.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 rar.txt
-rw-r--r-- 1 root root   18205 Feb 24 23:54 report.xml
-rw-r--r-- 1 root root   36579 Feb 24 23:45 report.xml.1645764857
-rw-r--r-- 1 root root       0 Feb 24 23:36 rfc822.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 sin.txt
-rw-r--r-- 1 root root       0 Feb 24 23:36 sqlite_carved.txt
-rw-r--r-- 1 root root       0 Feb 24 23:54 tcp_histogram_1.txt
-rw-r--r-- 1 root root       0 Feb 24 23:45 tcp_histogram.txt
```



```
-rw-r--r-- 1 root root        0 Feb 24 23:36 tcp.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 telephone_histogram_1.txt
-rw-r--r-- 1 root root      224 Feb 24 23:45 telephone_histogram.txt
-rw-r--r-- 1 root root      726 Feb 24 23:45 telephone.txt
-rw-r--r-- 1 root root        0 Feb 24 23:36 unrar_carved.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_facebook-address_1.txt
-rw-r--r-- 1 root root        0 Feb 24 23:45 url_facebook-address.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_facebook-id_1.txt
-rw-r--r-- 1 root root        0 Feb 24 23:45 url_facebook-id.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_histogram_1.txt
-rw-r--r-- 1 root root  3118502 Feb 24 23:45 url_histogram.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_microsoft-live_1.txt
-rw-r--r-- 1 root root        0 Feb 24 23:45 url_microsoft-live.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_searches_1.txt
-rw-r--r-- 1 root root        0 Feb 24 23:45 url_searches.txt
-rw-r--r-- 1 root root        0 Feb 24 23:54 url_services_1.txt
-rw-r--r-- 1 root root    77117 Feb 24 23:45 url_services.txt
-rw-r--r-- 1 root root 18807106 Feb 24 23:45 url.txt
-rw-r--r-- 1 root root        0 Feb 24 23:36 utmp_carved.txt
-rw-r--r-- 1 root root        0 Feb 24 23:36 vcard.txt
-rw-r--r-- 1 root root  1483946 Feb 24 23:45 windirs.txt
-rw-r--r-- 1 root root        0 Feb 24 23:36 winlnk.txt
drwxr-xr-x 3 root root     4096 Feb 24 23:43 winpe_carved
-rw-r--r-- 1 root root     6562 Feb 24 23:45 winpe_carved.txt
-rw-r--r-- 1 root root    76482 Feb 24 23:45 winpe.txt
-rw-r--r-- 1 root root        0 Feb 24 23:36 winprefetch.txt
drwxr-xr-x 3 root root     4096 Feb 24 23:43 zip
-rw-r--r-- 1 root root    39638 Feb 24 23:45 zip.txt
```

B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

bulk_extractor is a wonderful tool that carves data and finds useful information, such as email addresses, visited URLs, Facebook URLs, credit card numbers, and a variety of other information.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Questions of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1: what are open source and proprietary forensic tools for multimedia recovery?

1. Autopsy
   Autopsy is a GUI-based open-source digital forensic program to analyze hard drives and smart phones effectively. Thousands of users use autopsy worldwide to investigate what happened in the computer.

2. Encrypted Disk Detector
   Encrypted Disk Detector can be helpful to check encrypted physical drives. It supports TrueCrypt, PGP, Bitlocker, Safeboot encrypted volumes.
3. Wireshark
   Wireshark is a network capture and analyzer tool to see what is happening in your network. Wireshark will be handy to investigate network related incident.4. Magnet RAM Capture. You can use Magnet RAM to capture the physical memory of a computer and analyze artifacts in memory. It supports Windows operating system.
4. Network Miner
   An interesting network forensic analyzer for Windows, Linux & MAC OS X to detect OS, hostname, sessions, and open ports through packet sniffing or by PCAP file. Network Miner provide extracted artifacts in an intuitive user interface.6. NMAP