Forensic Tools Evaluation in Kali Linux

Er. Anupreet Kaur[1], Dr. Jaswinder Singh[2]

[1]*Department of Computer Science and Engineering, Punjabi University, Patiala*

Abstract -Computer Forensics is rapidly becoming more and more important due to growth of internet and different technologies related with it like Internet of Things (IoT) and Artificial Intelligence (AI). This paper evaluates the different forensic tools available in Kali Linux. Different tools uses different techniques and are used for various purposes. Kali Linux can be run in the forensic mode where it is mainly run for forensic related work. There are both open-source and proprietary forensic tools in the kali linux and selection of tools is mainly done on the basis of requirements. Forensic Tools results and configurations related with the tools are included in this paper.

Keywords – Binwalk, Kali, Bulk-Extractor, Forensics, Xplico

INTRODUCTION

Computer Forensics is an emerging field which is dynamic in nature and it keeps on changing as the device types and their software changed[1]. Digitization is ruling this technological world and individuals and companies are moving their data to the cloud. Technologies like IoT and AI are ruling the market and devices are communicating directly with the devices. Machine to Machine (M2M) communications are rising with IoT. Courts worldwide are seeing digital evidence as reliable these days. Every country has its own guidelines on digital data as evidence. In one of the famous cases related with Michael Jackson, his doctor was found responsible for the death of singer on the basis of digital evidence that was found of doctor's computer[17]. With the advancement in technologies and internet, we are seeing a big role of computer forensics in this computing era. There are five standard phases of forensics as given below:

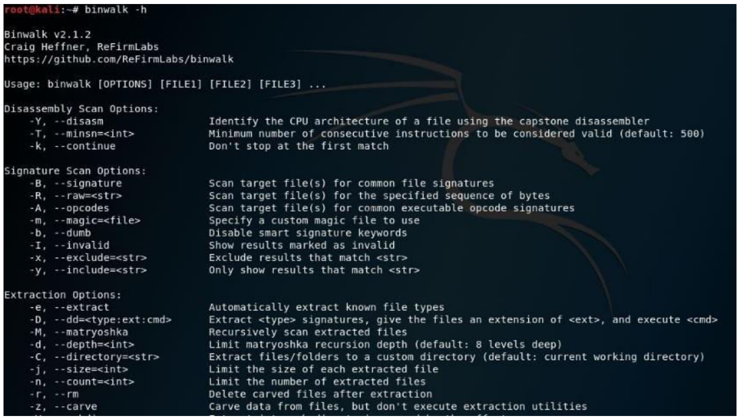
- Policy Development
- Assessment
- Acquisition
- Examination
- Reporting

FORENSIC TOOLS

Tool: **Binwalk**

Binwalk tool offers features to locate the binary image in documents and provide viable code options. Basically, it enables users to locate the code as well as the files in images. It uses libmagic library to perform actions.

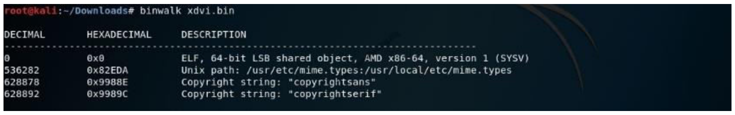
Step 1: For listing all the options of Binwalk



```
root@kali:~# binwalk -h

Binwalk v2.1.2
Craig Heffner, ReFirmLabs
https://github.com/ReFirmLabs/binwalk

Usage: binwalk [OPTIONS] [FILE1] [FILE2] [FILE3] ...

Disassembly Scan Options:
    -Y, --disasm                 Identify the CPU architecture of a file using the capstone disassembler
    -T, --minsn=<int>            Minimum number of consecutive instructions to be considered valid (default: 500)
    -k, --continue               Don't stop at the first match

Signature Scan Options:
    -B, --signature              Scan target file(s) for common file signatures
    -R, --raw=<str>              Scan target file(s) for the specified sequence of bytes
    -A, --opcodes                Scan target file(s) for common executable opcode signatures
    -m, --magic=<file>           Specify a custom magic file to use
    -b, --dumb                   Disable smart signature keywords
    -I, --invalid                Show results marked as invalid
    -x, --exclude=<str>          Exclude results that match <str>
    -y, --include=<str>          Only show results that match <str>

Extraction Options:
    -e, --extract                Automatically extract known file types
    -D, --dd=<type:ext:cmd>      Extract <type> signatures, give the files an extension of <ext>, and execute <cmd>
    -M, --matryoshka             Recursively scan extracted files
    -d, --depth=<int>            Limit matryoshka recursion depth (default: 8 levels deep)
    -C, --directory=<str>        Extract files/folders to a custom directory (default: current working directory)
    -j, --size=<int>             Limit the size of each extracted file
    -n, --count=<int>            Limit the number of extracted files
    -r, --rm                     Delete carved files after extraction
    -z, --carve                  Carve data from files, but don't execute extraction utilities
```

Figure 4.1 – Binwalk Options

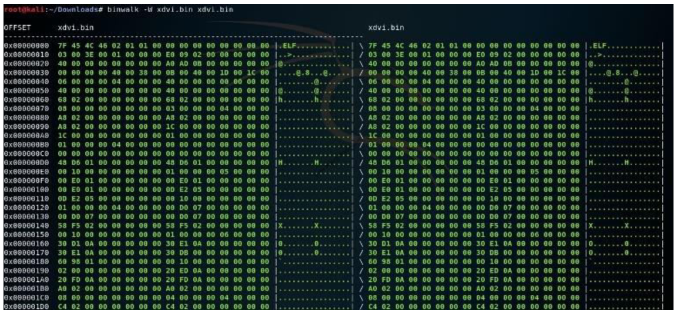
Step 2: For scanning the firmware for files and systems



```
root@kali:~/Downloads# binwalk xdvi.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             ELF, 64-bit LSB shared object, AMD x86-64, version 1 (SYSV)
536282        0x82EDA         Unix path: /usr/etc/mime.types:/usr/local/etc/mime.types
628878        0x9988E         Copyright string: "copyrightsans"
628892        0x9989C         Copyright string: "copyrightserif"
```

Figure 4.2 – Scanning firmware, files or other options

In order to check the difference between multiple files, we can use command in following figure:

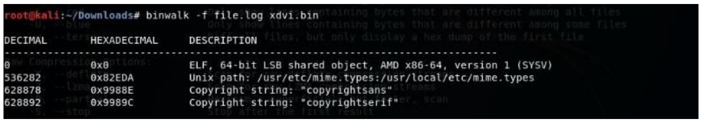


```
root@kali:~/Downloads# binwalk -W xdvi.bin xdvi.bin

OFFSET      xdvi.bin                                                                  xdvi.bin
-----------------------------------------------------------------------------------------------------------------------------------------------
0x00000000  7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF............| \ 7F 45 4C 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF............|
0x00000010  03 00 3E 00 01 00 00 00 E0 09 02 00 00 00 00 00 |..>.............| / 03 00 3E 00 01 00 00 00 E0 09 02 00 00 00 00 00 |..>.............|
0x00000020  40 00 00 00 00 00 00 00 A0 AD 0B 00 00 00 00 00 |@...............| \ 40 00 00 00 00 00 00 00 A0 AD 0B 00 00 00 00 00 |@...............|
0x00000030  00 00 00 00 40 00 38 00 0B 00 40 00 1D 00 1C 00 |....@.8...@.....| / 00 00 00 00 40 00 38 00 0B 00 40 00 1D 00 1C 00 |....@.8...@.....|
0x00000040  06 00 00 00 04 00 00 00 40 00 00 00 00 00 00 00 |........@.......| \ 06 00 00 00 04 00 00 00 40 00 00 00 00 00 00 00 |........@.......|
0x00000050  40 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |@.......@.......| / 40 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |@.......@.......|
0x00000060  68 02 00 00 00 00 00 00 68 02 00 00 00 00 00 00 |h.......h.......| \ 68 02 00 00 00 00 00 00 68 02 00 00 00 00 00 00 |h.......h.......|
0x00000070  08 00 00 00 00 00 00 00 03 00 00 00 04 00 00 00 |................| / 08 00 00 00 00 00 00 00 03 00 00 00 04 00 00 00 |................|
0x00000080  A8 02 00 00 00 00 00 00 A8 02 00 00 00 00 00 00 |................| \ A8 02 00 00 00 00 00 00 A8 02 00 00 00 00 00 00 |................|
0x00000090  A8 02 00 00 00 00 00 00 1C 00 00 00 00 00 00 00 |................| / A8 02 00 00 00 00 00 00 1C 00 00 00 00 00 00 00 |................|
0x000000A0  1C 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 |................| \ 1C 00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 |................|
0x000000B0  01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 |................| / 01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 |................|
0x000000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................| \ 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
0x000000D0  48 D6 01 00 00 00 00 00 48 D6 01 00 00 00 00 00 |H.......H.......| / 48 D6 01 00 00 00 00 00 48 D6 01 00 00 00 00 00 |H.......H.......|
0x000000E0  00 10 00 00 00 00 00 00 01 00 00 00 05 00 00 00 |................| \ 00 10 00 00 00 00 00 00 01 00 00 00 05 00 00 00 |................|
0x000000F0  00 E0 01 00 00 00 00 00 00 E0 01 00 00 00 00 00 |................| / 00 E0 01 00 00 00 00 00 00 E0 01 00 00 00 00 00 |................|
0x00000100  00 E0 01 00 00 00 00 00 0D E2 05 00 00 00 00 00 |................| \ 00 E0 01 00 00 00 00 00 0D E2 05 00 00 00 00 00 |................|
0x00000110  0D E2 05 00 00 00 00 00 00 10 00 00 00 00 00 00 |................| / 0D E2 05 00 00 00 00 00 00 10 00 00 00 00 00 00 |................|
0x00000120  01 00 00 00 04 00 00 00 00 D0 07 00 00 00 00 00 |................| \ 01 00 00 00 04 00 00 00 00 D0 07 00 00 00 00 00 |................|
0x00000130  00 D0 07 00 00 00 00 00 00 D0 07 00 00 00 00 00 |................| / 00 D0 07 00 00 00 00 00 00 D0 07 00 00 00 00 00 |................|
0x00000140  58 F5 02 00 00 00 00 00 58 F5 02 00 00 00 00 00 |X.......X.......| \ 58 F5 02 00 00 00 00 00 58 F5 02 00 00 00 00 00 |X.......X.......|
0x00000150  00 10 00 00 00 00 00 00 01 00 00 00 06 00 00 00 |................| / 00 10 00 00 00 00 00 00 01 00 00 00 06 00 00 00 |................|
0x00000160  30 D1 0A 00 00 00 00 00 30 E1 0A 00 00 00 00 00 |0.......0.......| \ 30 D1 0A 00 00 00 00 00 30 E1 0A 00 00 00 00 00 |0.......0.......|
0x00000170  30 E1 0A 00 00 00 00 00 30 DB 00 00 00 00 00 00 |0.......0.......| / 30 E1 0A 00 00 00 00 00 30 DB 00 00 00 00 00 00 |0.......0.......|
0x00000180  60 98 01 00 00 00 00 00 00 10 00 00 00 00 00 00 |`...............| \ 60 98 01 00 00 00 00 00 00 10 00 00 00 00 00 00 |`...............|
0x00000190  02 00 00 00 06 00 00 00 20 ED 0A 00 00 00 00 00 |................| / 02 00 00 00 06 00 00 00 20 ED 0A 00 00 00 00 00 |................|
0x000001A0  20 FD 0A 00 00 00 00 00 20 FD 0A 00 00 00 00 00 |................| \ 20 FD 0A 00 00 00 00 00 20 FD 0A 00 00 00 00 00 |................|
0x000001B0  A0 02 00 00 00 00 00 00 A0 02 00 00 00 00 00 00 |................| / A0 02 00 00 00 00 00 00 A0 02 00 00 00 00 00 00 |................|
0x000001C0  08 00 00 00 00 00 00 00 04 00 00 00 04 00 00 00 |................| \ 08 00 00 00 00 00 00 00 04 00 00 00 04 00 00 00 |................|
0x000001D0  C4 02 00 00 00 00 00 00 C4 02 00 00 00 00 00 00 |................| / C4 02 00 00 00 00 00 00 C4 02 00 00 00 00 00 00 |................|
```

Figure 4.3 – Comparing two different files

Step 3: For extracting the file types of firmware image, we use -e or -extract and for recursive file scanning, use –M.

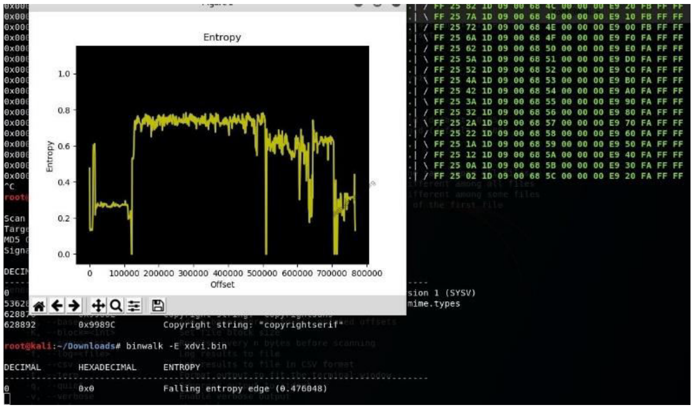
Step 4: To capture the log files,we can use the command as shown in figure below:



```
root@kali:~/Downloads# binwalk -f file.log xdvi.bin

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             ELF, 64-bit LSB shared object, AMD x86-64, version 1 (SYSV)
536282        0x82EDA         Unix path: /usr/etc/mime.types:/usr/local/etc/mime.types
628878        0x9988E         Copyright string: "copyrightsans"
628892        0x9989C         Copyright string: "copyrightserif"
```

Figure 4.4 – Capturing Logs in binwalk

The Entropy Evaluation is useful in searching certain sections of firmware image. It can be done using –E flag as shown in figure below:



```
.| / FF 25 82 1D 09 00 68 4C 00 00 00 E9 20 FB FF FF
.| \ FF 25 7A 1D 09 00 68 4D 00 00 00 E9 10 FB FF FF
.| / FF 25 72 1D 09 00 68 4E 00 00 00 E9 00 FB FF FF
.| \ FF 25 6A 1D 09 00 68 4F 00 00 00 E9 F0 FA FF FF
.| / FF 25 62 1D 09 00 68 50 00 00 00 E9 E0 FA FF FF
.| \ FF 25 5A 1D 09 00 68 51 00 00 00 E9 D0 FA FF FF
.| / FF 25 52 1D 09 00 68 52 00 00 00 E9 C0 FA FF FF
.| \ FF 25 4A 1D 09 00 68 53 00 00 00 E9 B0 FA FF FF
.| / FF 25 42 1D 09 00 68 54 00 00 00 E9 A0 FA FF FF
.| \ FF 25 3A 1D 09 00 68 55 00 00 00 E9 90 FA FF FF
.| / FF 25 32 1D 09 00 68 56 00 00 00 E9 80 FA FF FF
.| \ FF 25 2A 1D 09 00 68 57 00 00 00 E9 70 FA FF FF
.| / FF 25 22 1D 09 00 68 58 00 00 00 E9 60 FA FF FF
.| \ FF 25 1A 1D 09 00 68 59 00 00 00 E9 50 FA FF FF
.| / FF 25 12 1D 09 00 68 5A 00 00 00 E9 40 FA FF FF
.| \ FF 25 0A 1D 09 00 68 5B 00 00 00 E9 30 FA FF FF
.| / FF 25 02 1D 09 00 68 5C 00 00 00 E9 20 FA FF FF

628892        0x9989C         Copyright string: "copyrightserif"

root@kali:~/Downloads# binwalk -E xdvi.bin

DECIMAL       HEXADECIMAL     ENTROPY
--------------------------------------------------------------------------------
0             0x0             Falling entropy edge (0.476048)
```

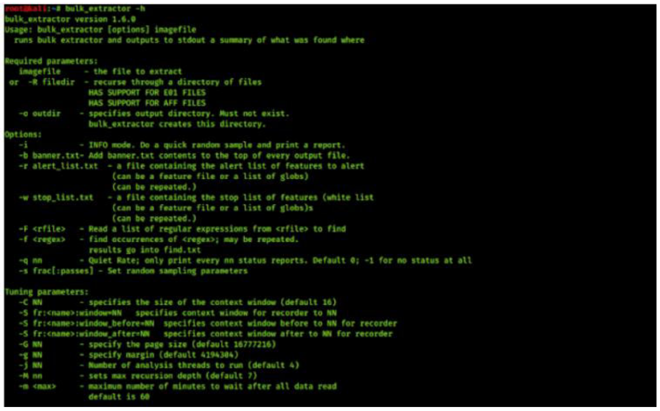
Figure 4.5 – Entropy Evaluation in Binwalk

Binwalk tool is considered as typical tool for the analysis of forensics. Also, it can be combined with other tools for better performance.

Tool: **Bulk-Extractor**

In the modern digital era, it becomes essential to find the sensitive data for digital evidences. For this Bulk-Extractor tool is quite useful. It fetches the data, such as email-address, credit card number, URL information, disk info, directory and so on. This tool comes with pre-installed packages in Kali Linux distribution.

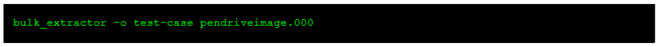
Step 1: To start the Bulk-extractor along with options available.



```
root@kali:~# bulk_extractor -h
bulk_extractor version 1.6.0
Usage: bulk_extractor [options] imagefile
  runs bulk extractor and outputs to stdout a summary of what was found where

Required parameters:
   imagefile     - the file to extract
 or  -R filedir  - recurse through a directory of files
                  HAS SUPPORT FOR E01 FILES
                  HAS SUPPORT FOR AFF FILES
   -o outdir    - specifies output directory. Must not exist.
                  bulk_extractor creates this directory.
Options:
   -i           - INFO mode. Do a quick random sample and print a report.
   -b banner.txt- Add banner.txt contents to the top of every output file.
   -r alert_list.txt  - a file containing the alert list of features to alert
                       (can be a feature file or a list of globs)
                       (can be repeated.)
   -w stop_list.txt   - a file containing the stop list of features (white list
                       (can be a feature file or a list of globs)s
                       (can be repeated.)
   -F <rfile>   - Read a list of regular expressions from <rfile> to find
   -f <regex>   - find occurrences of <regex>; may be repeated.
                  results go into find.txt
   -q nn        - Quiet Rate; only print every nn status reports. Default 0; -1 for no status at all
   -s frac[:passes] - Set random sampling parameters

Tuning parameters:
   -C NN        - specifies the size of the context window (default 16)
   -S fr:<name>:window=NN   specifies context window for recorder to NN
   -S fr:<name>:window_before=NN  specifies context window before to NN for recorder
   -S fr:<name>:window_after=NN   specifies context window after to NN for recorder
   -G NN        - specify the page size (default 16777216)
   -g NN        - specify margin (default 4194304)
   -j NN        - Number of analysis threads to run (default 4)
   -M nn        - sets max recursion depth (default 7)
   -m <max>     - maximum number of minutes to wait after all data read
                  default is 60
```

Figure 4.6 - Starting Bulk-Extractor

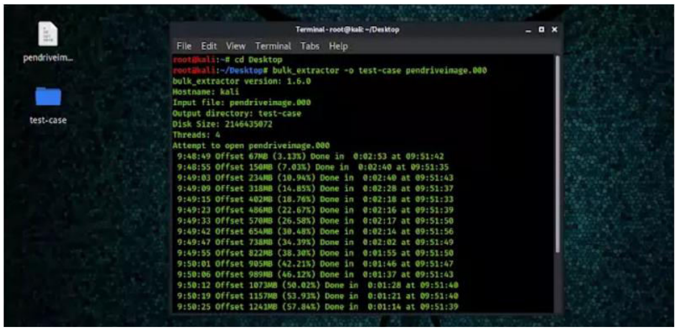
Step 2: The aforementioned options can be used to fetch the data from disk, drive or directories. Apart from this, -e flag can be used for scanning. There are certain scanners which remain disabled by default. In order to do default scanning, use this command



```
bulk_extractor -o test-case pendriveimage.000
```

Figure 4.7 - Default Scanning on Storage Device

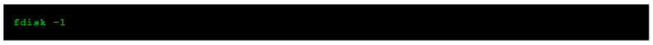
Step 3: The -o flag represents the name of the output directory.


pendriveim...
test-case
Terminal - root@kali: ~/Desktop
File Edit View Terminal Tabs Help

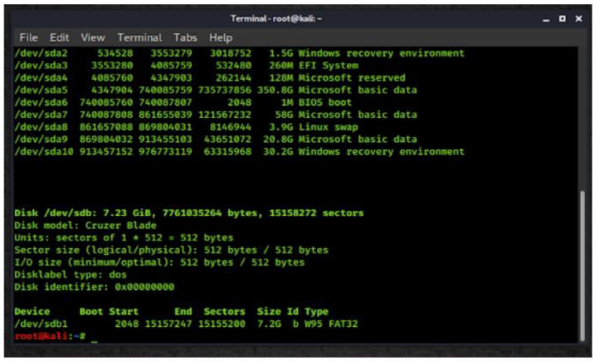
```
root@kali:~# cd Desktop
root@kali:~/Desktop# bulk_extractor -o test-case pendriveimage.000
bulk_extractor version: 1.6.0
Hostname: kali
Input file: pendriveimage.000
Output directory: test-case
Disk Size: 2146435072
Threads: 4
Attempt to open pendriveimage.000
 9:48:49 Offset 67MB (3.13%) Done in  0:02:53 at 09:51:42
 9:48:55 Offset 150MB (7.03%) Done in  0:02:40 at 09:51:35
 9:49:03 Offset 234MB (10.94%) Done in  0:02:40 at 09:51:43
 9:49:09 Offset 318MB (14.85%) Done in  0:02:28 at 09:51:37
 9:49:15 Offset 402MB (18.76%) Done in  0:02:18 at 09:51:33
 9:49:23 Offset 486MB (22.67%) Done in  0:02:16 at 09:51:39
 9:49:33 Offset 570MB (26.58%) Done in  0:02:17 at 09:51:50
 9:49:42 Offset 654MB (30.48%) Done in  0:02:14 at 09:51:56
 9:49:47 Offset 738MB (34.39%) Done in  0:02:02 at 09:51:49
 9:49:55 Offset 822MB (38.30%) Done in  0:01:55 at 09:51:50
 9:50:01 Offset 905MB (42.21%) Done in  0:01:46 at 09:51:47
 9:50:06 Offset 989MB (46.12%) Done in  0:01:37 at 09:51:43
 9:50:12 Offset 1073MB (50.02%) Done in  0:01:28 at 09:51:40
 9:50:19 Offset 1157MB (53.93%) Done in  0:01:21 at 09:51:40
 9:50:25 Offset 1241MB (57.84%) Done in  0:01:14 at 09:51:39
```

Figure 4.8 - Scanning PenDrive Image

Step 4: The whole scanning process may take a while as it depends on the enabled scanners as well as the size of the disk drive. For fast execution, unwanted scanners can be disabled. Further, scan the Pen Drive without creating any disk. For this, after plugging the drive into the system, we need to check the partitioning of the drive with the following command:



```
fdisk -l
```

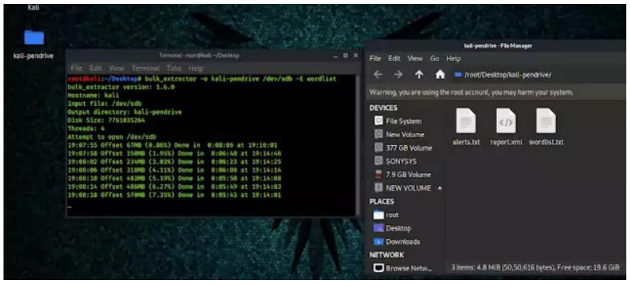


```
/dev/sda2     534528   3553279   3018752   1.5G Windows recovery environment
/dev/sda3    3553280   4085759    532480   260M EFI System
/dev/sda4    4085760   4347903    262144   128M Microsoft reserved
/dev/sda5    4347904 740085759 735737856 350.8G Microsoft basic data
/dev/sda6  740085760 740087807      2048     1M BIOS boot
/dev/sda7  740087808 861655039 121567232    58G Microsoft basic data
/dev/sda8  861657088 869804031   8146944   3.9G Linux swap
/dev/sda9  869804032 913455103  43651072  20.8G Microsoft basic data
/dev/sda10 913457152 976773119  63315968  30.2G Windows recovery environment


Disk /dev/sdb: 7.23 GiB, 7761035264 bytes, 15158272 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1        2048 15157247 15155200  7.2G  b W95 FAT32
root@kali:~# _
```

Figure 4.9 - Checking Storage Partitions

Step 5 : /dev/sdb is the drive that is attached to the machine. Let’s see how to use wordlist scanner. It will start generating the wordlist from the portable drive’s document.

Step 6: Final Output of the scanner



```
root@kali:~/Desktop# bulk_extractor -o kali-pendrive /dev/sdb -E wordlist
bulk_extractor version: 1.6.0
Hostname: kali
Input file: /dev/sdb
Output directory: kali-pendrive
Disk Size: 7761035264
Threads: 4
Attempt to open /dev/sdb
19:07:55 Offset 67MB (0.86%) Done in  0:08:06 at 19:16:01
19:07:58 Offset 150MB (1.95%) Done in  0:06:48 at 19:14:46
19:08:02 Offset 234MB (3.03%) Done in  0:06:23 at 19:14:25
19:08:06 Offset 318MB (4.11%) Done in  0:06:08 at 19:14:14
19:08:10 Offset 402MB (5.19%) Done in  0:05:58 at 19:14:08
19:08:14 Offset 486MB (6.27%) Done in  0:05:49 at 19:14:03
19:08:18 Offset 570MB (7.35%) Done in  0:05:43 at 19:14:01
```

Warning, you are using the root account, you may harm your system.

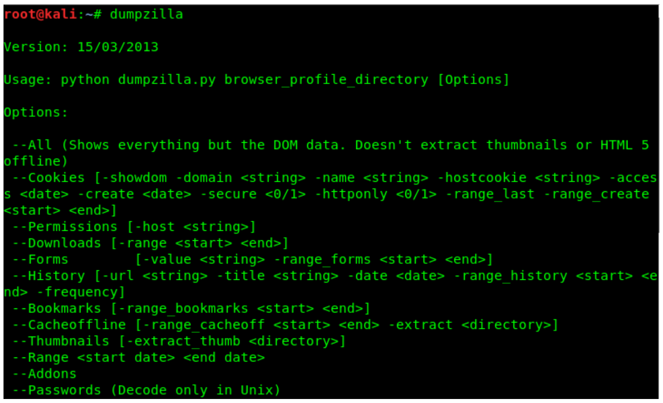
3 items: 4.8 MiB (50,50,616 bytes), Free space: 19.6 GiB

Figure 4.10 - Final Output of Bulk Extractor

Tool :**Dumpzilla**

Dumpzilla is a powerful browser forensic tool based on command line. It works with all the major operating systems, including Windows, Mac, and Linux. It is already installed in Kali Linux operating system. The code of the Dumpzilla has been writtern in Python3 programming language. It offers to extract the information browsers, such as Firefox, Seamonkey and so on. Apart from this, Dumpzilla can extract the information from browser's cookies, DOM, history, web forms, browser passwords, SSL certificated and many more.

Step 1 : To start the Dumpzilla tool, open the Kali terminal and enter the following command.



```
root@kali:~# dumpzilla

Version: 15/03/2013

Usage: python dumpzilla.py browser_profile_directory [Options]

Options:

 --All (Shows everything but the DOM data. Doesn't extract thumbnails or HTML 5
offline)
 --Cookies [-showdom -domain <string> -name <string> -hostcookie <string> -acces
s <date> -create <date> -secure <0/1> -httponly <0/1> -range_last -range_create
<start> <end>]
 --Permissions [-host <string>]
 --Downloads [-range <start> <end>]
 --Forms       [-value <string> -range_forms <start> <end>]
 --History [-url <string> -title <string> -date <date> -range_history <start> <e
nd> -frequency]
 --Bookmarks [-range_bookmarks <start> <end>]
 --Cacheoffline [-range_cacheoff <start> <end> -extract <directory>]
 --Thumbnails [-extract_thumb <directory>]
 --Range <start date> <end date>
 --Addons
 --Passwords (Decode only in Unix)
```

Figure 4.11 - Starting Dumpzilla

In Mozilla Firefox browser, the data is saved in profiles; therefore, in order to fetch the data for forensic analysis, dumpzilla is helpful. However, there is a necessity to get the default path of the profile. For different operating systems, the paths are different.

For Windows:

C:\Documents and Settings\xx\Application Data\Mozilla\Firefox\Profiles\xxxx.default

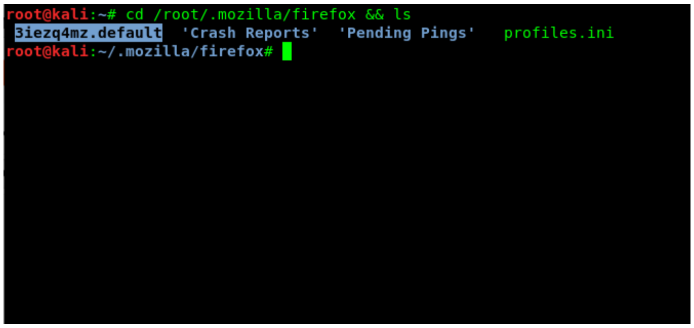
For Mac Operating System:

/users/$USER/.mozilla/firefox/xxxx.default

For Linux :

/home/$USER/.mozilla/firefox/xxxx.default

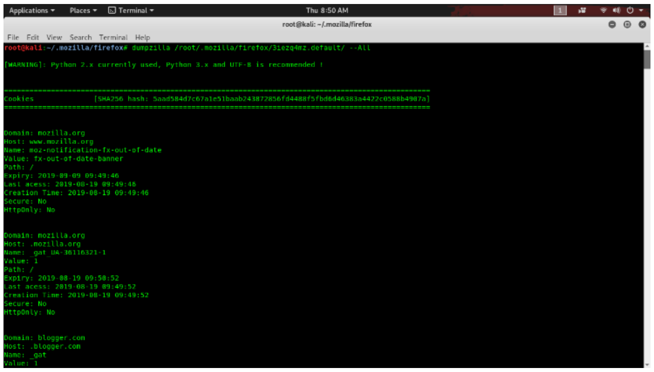
Step 2: use the following command to add the path in Kali Linux environment.



```
root@kali:~# cd /root/.mozilla/firefox && ls
 3iezq4mz.default  'Crash Reports'  'Pending Pings'   profiles.ini
root@kali:~/.mozilla/firefox# 
```

Figure 4.12 - Adding path in Kali Linux environment

Step 3: To run the Dumpzilla with default profile.

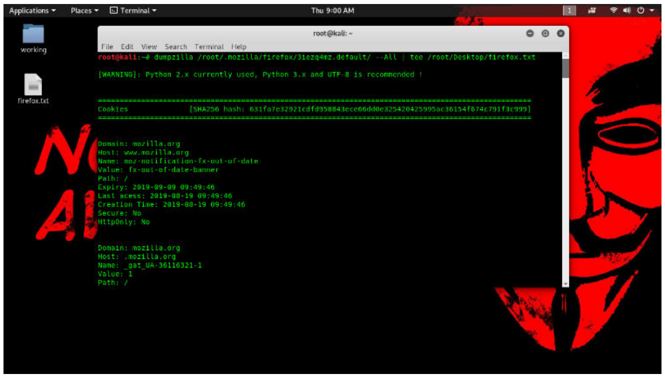


Applications
Places
Terminal
Thu 8:50 AM
root@kali: ~/.mozilla/firefox
File Edit View Search Terminal Help

```
root@kali:~/.mozilla/firefox# dumpzilla /root/.mozilla/firefox/3iezq4mz.default/ --All

[WARNING]: Python 2.x currently used, Python 3.x and UTF-8 is recommended !


=========================================================================================
Cookies               [SHA256 hash: 5aad584d7c67a1e51baab243872856fd4488f5fbd6d46383a4422c0588b4907a]
=========================================================================================


Domain: mozilla.org
Host: www.mozilla.org
Name: moz-notification-fx-out-of-date
Value: fx-out-of-date-banner
Path: /
Expiry: 2019-09-09 09:49:46
Last acess: 2019-08-19 09:49:48
Creation Time: 2019-08-19 09:49:46
Secure: No
HttpOnly: No


Domain: mozilla.org
Host: .mozilla.org
Name: _gat_UA-36116321-1
Value: 1
Path: /
Expiry: 2019-08-19 09:50:52
Last acess: 2019-08-19 09:49:52
Creation Time: 2019-08-19 09:49:52
Secure: No
HttpOnly: No


Domain: blogger.com
Host: .blogger.com
Name: _gat
Value: 1
```

Figure 4.13 - Running Dumpzilla using the default profile

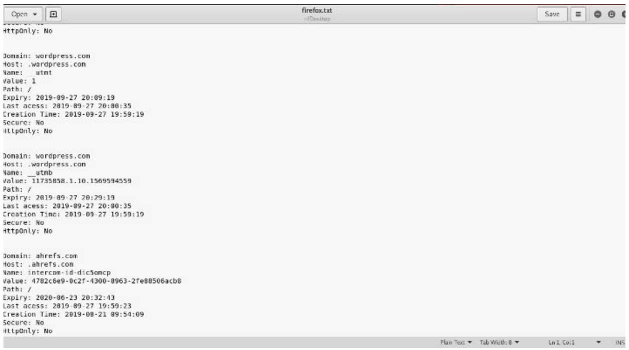
Step 4: For extraction of the whole data from text file.



```
root@kali:~# dumpzilla /root/.mozilla/firefox/3iezq4mz.default/ --All | tee /root/Desktop/firefox.txt

[WARNING]: Python 2.x currently used, Python 3.x and UTF-8 is recommended !


=========================================================================================
Cookies               [SHA256 hash: 631fa7e32921cdfd958843ecc66dd0e325420425995ac36154f674c791f3c999]
=========================================================================================


Domain: mozilla.org
Host: www.mozilla.org
Name: moz-notification-fx-out-of-date
Value: fx-out-of-date-banner
Path: /
Expiry: 2019-09-09 09:49:46
Last acess: 2019-08-19 09:49:46
Creation Time: 2019-08-19 09:49:46
Secure: No
HttpOnly: No


Domain: mozilla.org
Host: .mozilla.org
Name: _gat_UA-36116321-1
Value: 1
Path: /
```

Figure 4.14 - Extraction of data using the text file in Dumpzilla

Step 5: After the extraction of data from Mozilla Firefox, the final outcome will look like this.



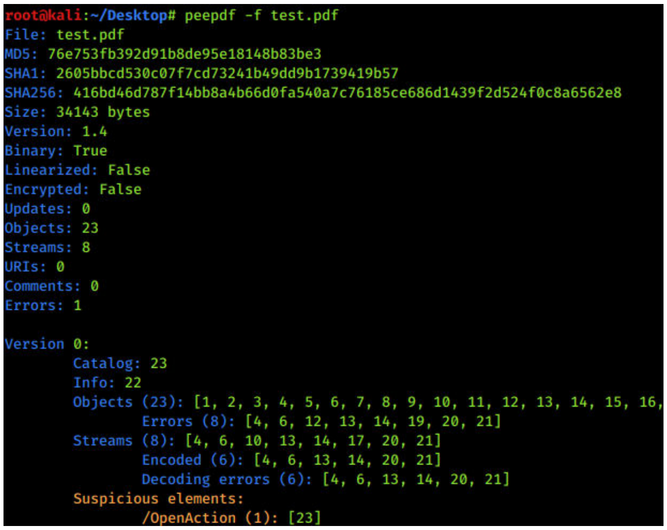
```
HttpOnly: No


Domain: wordpress.com
Host: .wordpress.com
Name:   utmt
Value: 1
Path: /
Expiry: 2019-09-27 20:09:19
Last acess: 2019-09-27 20:00:35
Creation Time: 2019-09-27 19:59:19
Secure: No
HttpOnly: No


Domain: wordpress.com
Host: .wordpress.com
Name: __utmb
Value: 11735858.1.10.1569594559
Path: /
Expiry: 2019-09-27 20:29:19
Last acess: 2019-09-27 20:00:35
Creation Time: 2019-09-27 19:59:19
Secure: No
HttpOnly: No


Domain: ahrefs.com
Host: .ahrefs.com
Name: intercom-id-dic5omcp
Value: 4782c6e9-0c2f-4300-8963-2fe88506acb8
Path: /
Expiry: 2020-06-23 20:32:43
Last acess: 2019-09-27 19:59:23
Creation Time: 2019-08-21 09:54:09
Secure: No
HttpOnly: No
```

Figure 4.15 - Final output in Firefox

Tool :**Peepdf**

Peeppdf is quite useful tool to investigate the pdf file that might be infected with virus and malwares or payload. It is python written command line tool which examine the pdf file for any kind of discrepancy. Some of the primary features of this tool are: decoding, string analysis, physical structure analysis, metadata, shellcode, hash checking and so on.

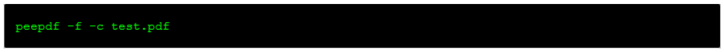
Step 1: To run the Peepdf, run the following command. For referencing purposes, we are using test.pdf and test2.pdf files, in which one has malicious content and another has no malicious content.



```
root@kali:~/Desktop# peepdf -f test.pdf
File: test.pdf
MD5: 76e753fb392d91b8de95e18148b83be3
SHA1: 2605bbcd530c07f7cd73241b49dd9b1739419b57
SHA256: 416bd46d787f14bb8a4b66d0fa540a7c76185ce686d1439f2d524f0c8a6562e8
Size: 34143 bytes
Version: 1.4
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 23
Streams: 8
URIs: 0
Comments: 0
Errors: 1

Version 0:
        Catalog: 23
        Info: 22
        Objects (23): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
                Errors (8): [4, 6, 12, 13, 14, 19, 20, 21]
        Streams (8): [4, 6, 10, 13, 14, 17, 20, 21]
                Encoded (6): [4, 6, 13, 14, 20, 21]
                Decoding errors (6): [4, 6, 13, 14, 20, 21]
        Suspicious elements:
                /OpenAction (1): [23]
```

Figure 4.16- Running PeepPDF

Step 2: To check the file hashing, we will enter command:

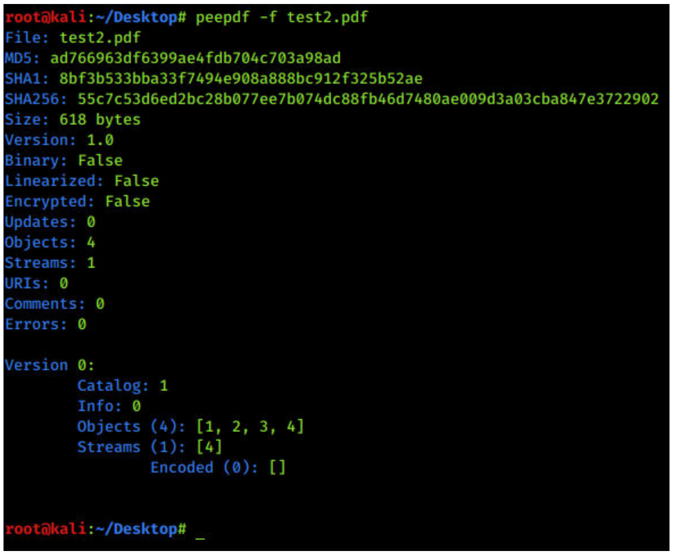


```
peepdf -f -c test.pdf
```

Figure 4.17 – Checking File Hash with PeepPDF

After this, we will get to know about the CVE and other relevant information.

Step 3: We will check the non-infected file using the similar way.



```
root@kali:~/Desktop# peepdf -f test2.pdf
File: test2.pdf
MD5: ad766963df6399ae4fdb704c703a98ad
SHA1: 8bf3b533bba33f7494e908a888bc912f325b52ae
SHA256: 55c7c53d6ed2bc28b077ee7b074dc88fb46d7480ae009d3a03cba847e3722902
Size: 618 bytes
Version: 1.0
Binary: False
Linearized: False
Encrypted: False
Updates: 0
Objects: 4
Streams: 1
URIs: 0
Comments: 0
Errors: 0

Version 0:
        Catalog: 1
        Info: 0
        Objects (4): [1, 2, 3, 4]
        Streams (1): [4]
                Encoded (0): []


root@kali:~/Desktop# _
```

Figure 4.18 – Check File Hash of Non-Infected File

It is evident that there is no malicious content discovered by peepdf tool; hence, this tool is effective in checking the infectious content in files.

Tool: Xplico

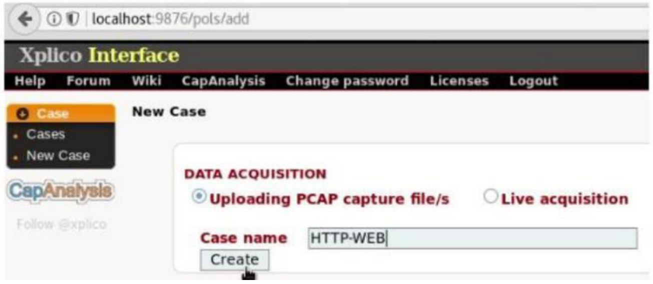
Xplico is used to fetch packets from the internet and analyze the application data under them. An example for Xplico is using a pcap file Xplico can get the HTTP content, FTP, TFTP, DNS, SIP, EMAIL protocols like SMTP, POP3 or IMAP etc. application layer protocols. It is an open-source Network Forensic Analysis Tools. Kali Linux has Xplico pre-installed and is popular among forensic testers. One can login into the Kali Linux Xplico by using the credentials given below:

- Username – xplico
- Password – xplico

Steps to use Xplico

- Packets can be captured by using tools like Wireshark and then the saved pcap file is imported into the Xplico to analyze.
- Packet Capture file i.e. pcap can be then used to investigate the application layer data.
- Data can be of any type like:
    - HTTP
    - SMTP, POP3
    - Facebook, Hangout etc. chats.
    - VoIP Traffic

To start the process, we firstly create a new case:



New Case

DATA ACQUISITION

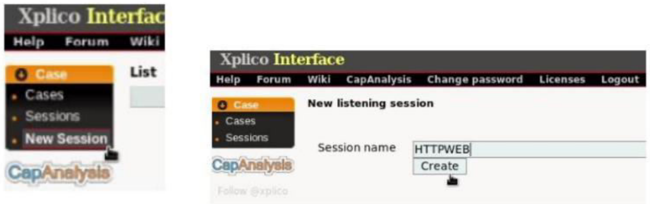
● Uploading PCAP capture file/s ○ Live acquisition

Case name: HTTP-WEB

Create

Figure 4.19 - Xplico New Case

After creating the case and adding the name of the case, we need to click on the case name to create a new session as shown below:



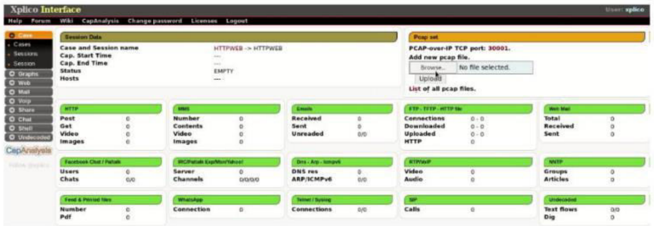
New listening session

Session name: HTTPWEB

Create

Figure 4.20 - Creating Session in Xplico

Once the case and session details are added, we enters into the Xplico interface dashboard that shows different categories of artifacts. We then upload the saved pcap file that we have captured from the tools like Wireshark as shown below:



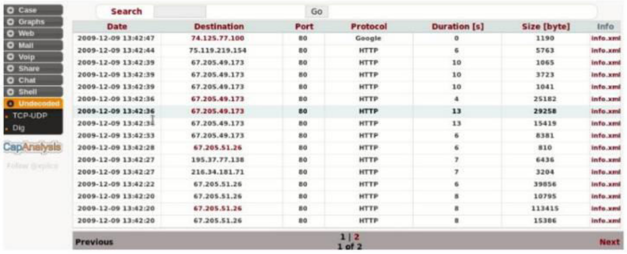
Case and Session name HTTPWEB -> HTTPWEB
Cap. Start Time ---
Cap. End Time ---
Status EMPTY
Hosts ---

PCAP-over-IP TCP port: 30001.
Add new pcap file.
List of all pcap files.

Post	0	Number	0	Received	0	Connections	0 - 0	Total	0
Get	0	Contents	0	Sent	0	Downloaded	0 - 0	Received	0
Video	0	Video	0	Unreaded	0/0	Uploaded	0 - 0	Sent	0
Images	0	Images	0			HTTP	0		
Users	0	Server	0	DNS res	0	Video	0	Groups	0
Chats	0/0	Channels	0/0/0/0	ARP/ICMPv6	0/0	Audio	0	Articles	0
Number	0	Connection	0	Connections	0/0	Calls	0	Text flows	0/0
Pdf	0							Dig	0

Figure 4.21 - Uploading the pcap file

The process related with decoding depends on the size of the pcap file and the type of data in the file which can be of single type or multiple types. When completed, the status field displays the Decoding completed message. Then we can use the Xplico Interface to shown the analysis on the basis of different protocols as shown in the figure below:



Date	Destination	Port	Protocol	Duration [s]	Size [byte]	Info
2009-12-09 13:42:47	74.125.77.100	80	Google	0	1190	info.xml
2009-12-09 13:42:44	75.119.219.154	80	HTTP	6	5763	info.xml
2009-12-09 13:42:39	67.205.49.173	80	HTTP	10	1065	info.xml
2009-12-09 13:42:39	67.205.49.173	80	HTTP	10	3723	info.xml
2009-12-09 13:42:39	67.205.49.173	80	HTTP	10	1041	info.xml
2009-12-09 13:42:36	67.205.49.173	80	HTTP	4	25182	info.xml
2009-12-09 13:42:36	67.205.49.173	80	HTTP	13	29258	info.xml
2009-12-09 13:42:34	67.205.49.173	80	HTTP	13	15419	info.xml
2009-12-09 13:42:33	67.205.49.173	80	HTTP	6	8381	info.xml
2009-12-09 13:42:28	67.205.51.26	80	HTTP	6	810	info.xml
2009-12-09 13:42:27	195.37.77.138	80	HTTP	7	6436	info.xml
2009-12-09 13:42:27	216.34.181.71	80	HTTP	7	3204	info.xml
2009-12-09 13:42:22	67.205.51.26	80	HTTP	6	39856	info.xml
2009-12-09 13:42:20	67.205.51.26	80	HTTP	8	10795	info.xml
2009-12-09 13:42:20	67.205.51.26	80	HTTP	8	113415	info.xml
2009-12-09 13:42:20	67.205.51.26	80	HTTP	8	15386	info.xml

Figure 4.22 - Different protocol analysis in Xplico

Then to show the contents of the files, we can directly open the files which are saved on the location or there is option to use the cat command to display the contents on the terminal by using command: cat /root/Documents/undecode.txt

There is also a dig sub-menu which displays various image artifacts like .gif, .png or .jpg formats, it also reveal the dates on these are viewed by using the HTTP connection.

If we click on the graph menu, then it shows the information related with the domain that consists of hostname, CName, Ipbinded with the host machine, and an xml files related with every entry.

Image Search option shows all the images. Images can also be viewed using the left web-menu and then by clicking on the images sub-menu.

Therefore Xplico is mainly used to analyze the pcap files fetched from tools like wireshark in better and organized manner. It segregates the application layer protocols and displays the content in an organized manner.

Tool – Foremost

Foremost is one of the most popular forensic tool which is use to recover the deleted files or lost files on the basis of their headers. Data can also be recovered on the basis of footers or data structures. It is used to fetch data from the hard drives, flash drives etc. This tool is mainly used to recover data files like images, documents, vides, installer files like msi or exe etc. This tool plays critical roles in order to recover any form of data from the criminal or target's storage device.

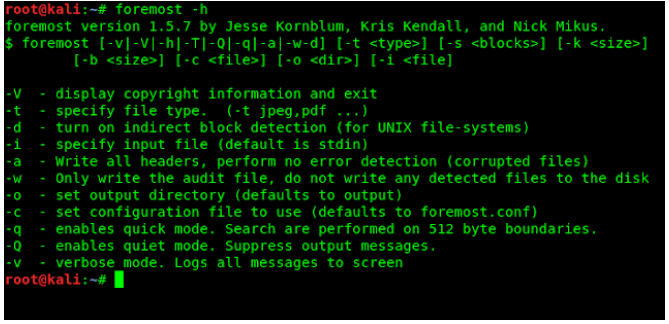
This tool works on the command line basis, where we can access it using the command line or terminal using the following command:

#apt install foremost

To check the commands and attributes related with the foremost, we can use the help command of the foremost, which is stated below:

#foremost –h

Screenshot of foremost help is shown below:



```
root@kali:~# 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 to screen
root@kali:~# 
```

Figure 4.23 - Foremost help command

Using options listed above, we can recover the files from the storage devices and we have connected a pen drive with the data shown below in the figure:

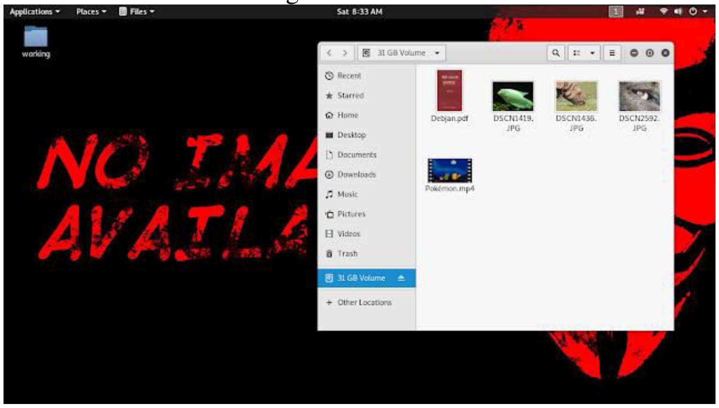
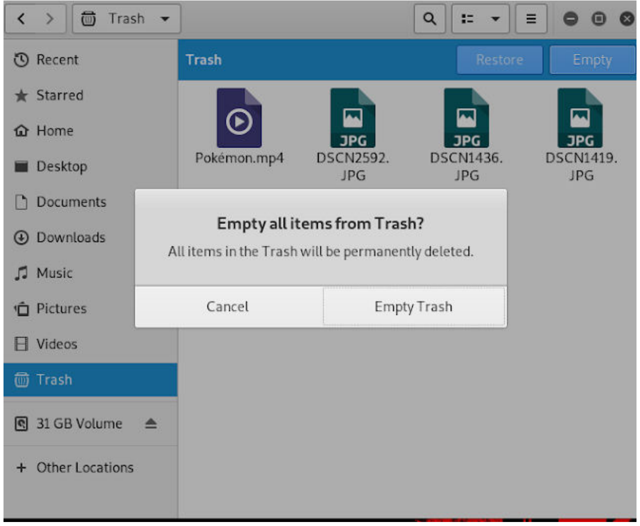


Figure 4.24 - Pen drive with original data

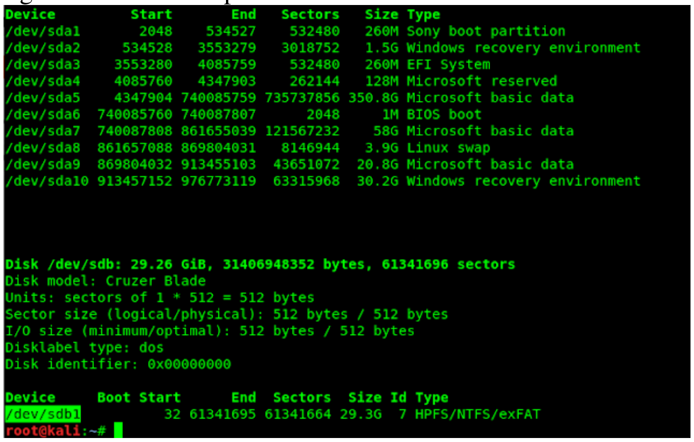
There are three image files, one pdf and mp4 file. We will delete these files as shown in the next figure. Files will be moved to trash and we will empty the trash directory.



Empty all items from Trash?
All items in the Trash will be permanently deleted.

Figure 4.25 - Empty the contents of trash folder

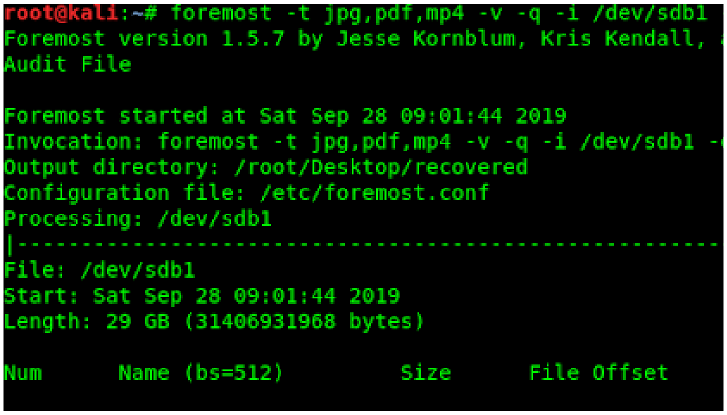
Now as files are permanently deleted. We will recover the data which was permanently deleted from our storage device and to recover the data, we need to know the storage path of the device and open that in the terminal as shown in the below figure of fdisk –l output:



```
Device         Start       End   Sectors   Size Type
/dev/sda1       2048    534527    532480   260M Sony boot partition
/dev/sda2     534528   3553279   3018752   1.5G Windows recovery environment
/dev/sda3    3553280   4085759    532480   260M EFI System
/dev/sda4    4085760   4347903    262144   128M Microsoft reserved
/dev/sda5    4347904 740085759 735737856 350.8G Microsoft basic data
/dev/sda6  740085760 740087807      2048     1M BIOS boot
/dev/sda7  740087808 861655039 121567232    58G Microsoft basic data
/dev/sda8  861657088 869804031   8146944   3.9G Linux swap
/dev/sda9  869804032 913455103  43651072  20.8G Microsoft basic data
/dev/sda10 913457152 976773119  63315968  30.2G Windows recovery environment



Disk /dev/sdb: 29.26 GiB, 31406948352 bytes, 61341696 sectors
Disk model: Cruzer Blade
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot Start      End  Sectors  Size Id Type
/dev/sdb1          32 61341695 61341664 29.3G  7 HPFS/NTFS/exFAT
root@kali:~# 
```

Figure 4.26 - fdisk –l

Drive location is /dev/sdb and the partition drive is sdb1. We can use sdb1 partition and use this in our recovery command shown below:



```
root@kali:~# foremost -t jpg,pdf,mp4 -v -q -i /dev/sdb1
Foremost version 1.5.7 by Jesse Kornblum, Kris Kendall, 
Audit File

Foremost started at Sat Sep 28 09:01:44 2019
Invocation: foremost -t jpg,pdf,mp4 -v -q -i /dev/sdb1 -
Output directory: /root/Desktop/recovered
Configuration file: /etc/foremost.conf
Processing: /dev/sdb1
|----------------------------------------------------------
File: /dev/sdb1
Start: Sat Sep 28 09:01:44 2019
Length: 29 GB (31406931968 bytes)

Num      Name (bs=512)          Size       File Offset
```

Figure 4.27 - Recovering files using foremost

Above command uses –t attribute or flag which we have used, otherwise foremost will look for all file formats and recover all file types. Flag –v is used for verbose mode and that dispays the running process related with the command we have used. Flag –q is used for quick mode and –I flag is used to enter the input storage device path, which in our case is /dev/sdb1. Flag –o shows the output directory where we want to have our recovered files. The recovery process takes time as it first scans the whole disk and then starts the recovery of files. In case the deleted files were overwritten by other files, then we may face problems in recovering the original files. As we have entered the command to recover the files, below figure shows that the files are successfully recovered:

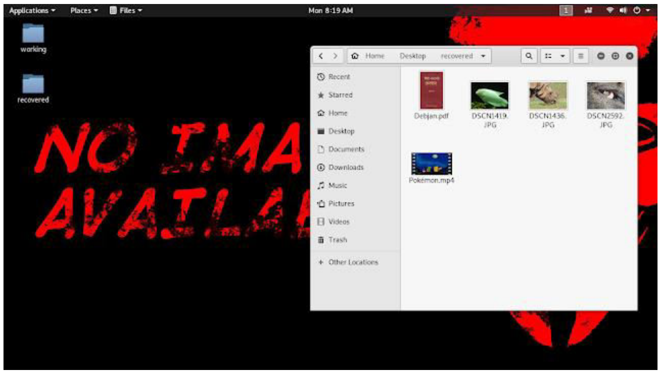


Figure 4.28 - Recovered Files

Conclusion and Future Scope

Computer Forensics is getting more and more important with technological advancements and increasing number of attacks on devices which can be on-premises or on cloud. Kali Linux has a pre-built package for popular computer forensics tools which can be used to protect the devices and digital data by providing confidentiality, integrity and authentication checks or testing. There are open source and proprietary forensic tools in Kali Linux. These forensic tools have the ability to perform forensic tasks like assessment, inspection, reporting etc. Kali Linux can be used in Forensic Mode which brings meagre changes as compared to the normal Kali Linux installation. One of the biggest advantage of using Kali Linux is that all the tools are pre-loaded and we can easily enhance the package by adding more tools and updating the existing packages as per requirement. We have done experimentation or evaluation on tools like Binwalk, Bulk-Extractor, Xplico, peepPDF, foremost etc. We found that open-source forensic tools in Kali Linux can be used in forensic research and they work without any flaws and able to perform forensic analysis and secure the systems and application accordingly. In future, we want to create a framework built using multiple open-source tools and scripts do perform forensic analysis on a single click of a mouse that will make forensic analysis much easier than before.

References

[1]Ghafarian A. (2019) Using Kali Linux Security Tools to Create Laboratory Projects for Cybersecurity Education. In: Arai K., Bhatia R., Kapoor S. (eds) Proceedings of the Future Technologies Conference (FTC) 2018. FTC 2018. Advances in Intelligent Systems and Computing, vol 881. Springer, Cham

[2]InfoSec Institute: Web Analysis, Vulnerability Assessment, and Exploitation using
Backtrack 5. [online] Available at: <http://resources.infosecinstitute.com/web-analysis-bt-5/>

[3]Danseglio, M.: Adding a new non-root user in Kali Linux. [online] Available at: <https://www.interfacett.com/blogs/adding-a-new-non-root-user-in-kali-linux/>.

[4]Top 25 best Kali Linux Tools.[online] Available at: <https://linuxhint.com/top-25-best-kali-linux-tools/>.

[5]Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/binwalk>.

[6] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/bulk-extractor>.

[7] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/cuckoo>.

[8] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/dc3dd>.

[9] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/dumpzilla>.

[10] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/peeppdf>.

[11] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/pdgmail>.

[12] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/p0f>.

[13] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/iphone-backup-analyzer >.

[14] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/foremost>.

[15] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/extundelete>.

[16] Tools.kali.org. 2020. [online] Available at: <https://tools.kali.org/forensics/xplico>.