Lab 2: Data Carving

What You Need for this lab

- Install Virtualbox : https://www.virtualbox.org/wiki/Downloads
- Install Kali 2021.4. : https://old.kali.org/kali-images/kali-2021.4/
  - Notes: Suggest You configure the disk size of Kali VM 80G because the size of each leakage cases image is 30G+
- Run a tool installation script instructions, or you can simply follow the commands below : (the script ONLY is tested on Kali 2021.4)
  - wget https://raw.githubusercontent.com/frankwxu/digital-forensics-lab/main/Help/tool-install-zsh.sh
  - chmod +x tool-install-zsh.sh
  - ./tool-install-zsh.sh

Example scenarios

- Scenario 1: A file (A) is hidden inside of another file (B). You can't open the file B because the B's header is corrupted.
- Scenario 2: A suspect deleted files. The files contains an important information. A file occupies a few clusters. Unfortunately, some clusters are reused (overwritten) by new files.

A forensic expert really wants to recover files, even a partial files.

1. **Extracting images from a corrupted Word document**

Step 1.

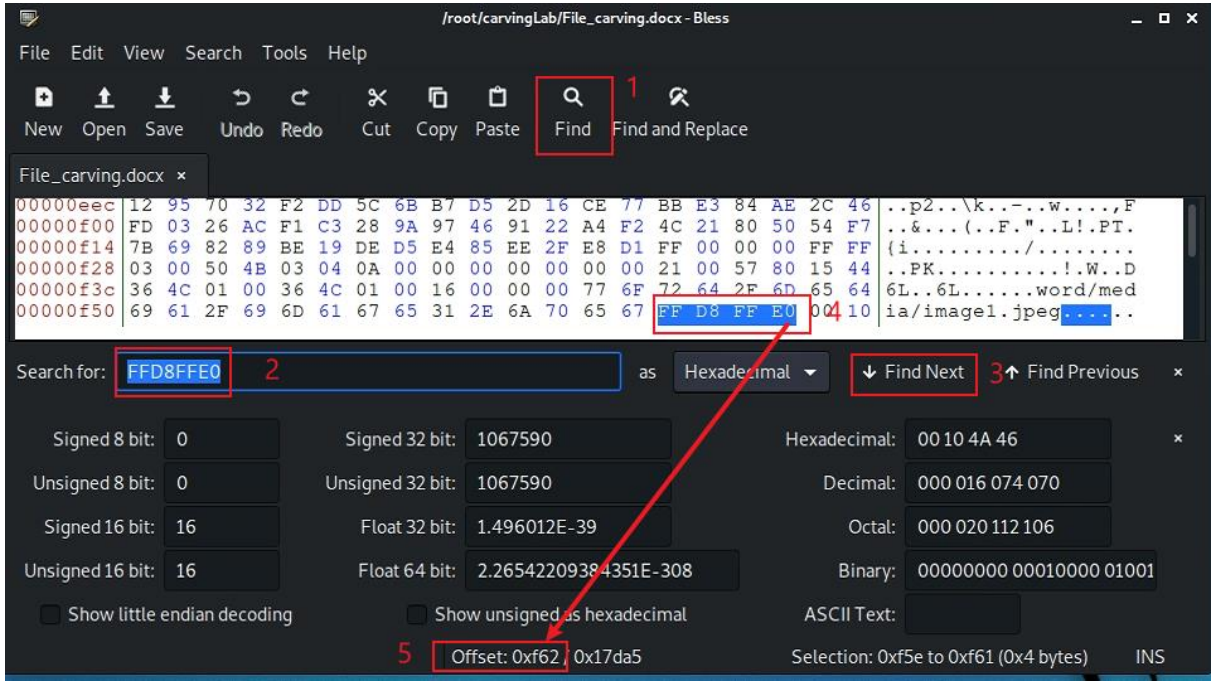
- Prepare required files

Step 2.

- View the file in a hex editor

Step 3.

– Search file header start offset – 0F5E



– Search file trailer ends offset – 15B93

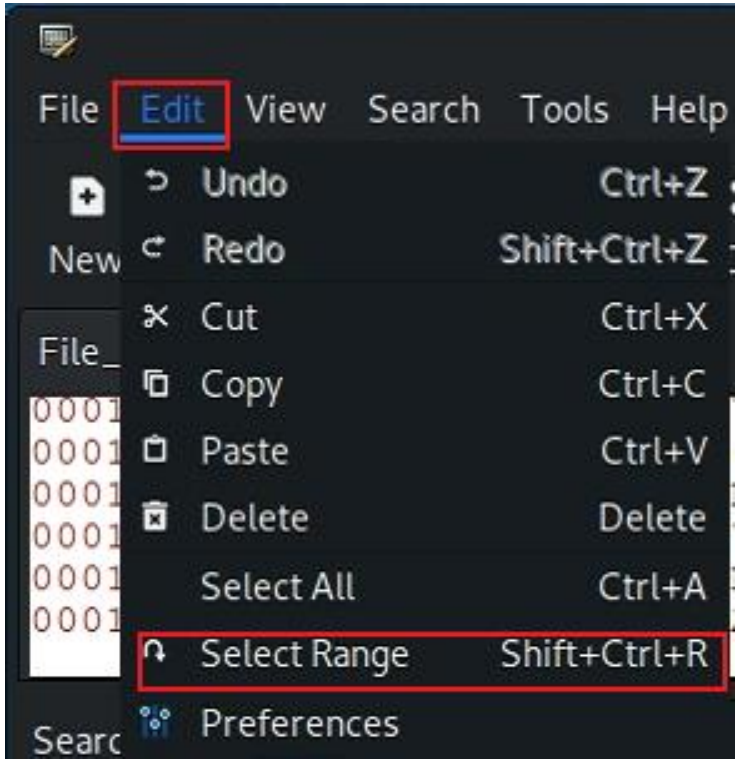
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00015A70  FF 00 0B 3B EF FE 3B 47 D6 BC 83 FB 3B FB DF 81  ÿ..;ïþ;GÖ¼ƒû;ûß.
00015A80  FD C4 3C 3B 53 7B 8E BC 67 1D 6A 14 4D DC 8E 3E  ýÄ<;S{Ž¼g.j.MÜŽ>
00015A90  6E 80 57 F0 FE BF B5 A7 ED 56 C3 27 F6 9A F8 85  n€Wðþ¿µ§íVÃ'öšø…
00015AA0  FF 00 85 A5 F7 FF 00 1D A0 FE D6 9F B5 50 19 1F  ÿ.…¥÷ÿ.. þÖŸµP..
00015AB0  B4 D7 C4 2F FC 2D 2F BF F8 ED 1F 5B 7D 83 FB 3B  ´×Ä/ü-/¿øí.[}ƒû;
00015AC0  FB DF 81 FD C4 7D E1 CD 46 F1 EC 1F BC 50 43 1F  ûß.ýÄ}áÍFñì.¼PC.
00015AD0  EF 57 F0 F3 FF 00 0D 6D FB 56 7F D1 CD 7C 41 FF  ïWðóÿ..mûV.ÑÍ|Aÿ
00015AE0  00 C2 CE FB FF 00 8E D1 FF 00 0D 6D FB 56 7F D1  .ÂÎûÿ.ŽÑÿ..mûV.Ñ
00015AF0  CD 7C 41 FF 00 C2 CE FB FF 00 8E D1 F5 AF 20 FE  Í|Aÿ.ÂÎûÿ.ŽÑõ¯ þ
00015B00  CF 8A FB 47 F7 04 D6 D0 B6 D5 31 9A 9E 2C A6 62  ÏŠûG÷.ÖÐ¶Õ1šž,¦b
00015B10  41 81 D2 BF 87 8F F8 6B 3F DA AB FE 8E 6B E2 17  A.Ò¿‡.øk?Ú«þŽkâ.
00015B20  FE 16 77 DF FC 76 8F F8 6B 3F DA AB FE 8E 6B E2  þ.wßüv.øk?Ú«þŽkâ
00015B30  17 FE 16 77 DF FC 76 8F AD 79 0D 60 2D F6 BF 03  .þ.wßüv..y.`-ö¿.
00015B40  FA 7E FF 00 83 A2 C2 8F F8 22 BF C5 1C 28 1F F1  ú~ÿ.ƒ¢Â.ø"¿Å.(.ñ
00015B50  34 F0 EF FE 9E AC E8 AF E5 D7 C5 9F B4 1F C7 AF  4ðïþž¬è¯å×ÅŸ´.Ç¯
00015B60  1D F8 72 4F 0C F8 EB E3 67 8B B5 BD 36 E3 6F 9F  .ørO.øëãg‹µ½6ãoŸ
00015B70  A7 EA DE 24 BA B8 82 42 AF B9 4B 23 C8 55 88 65  §êÞ$º¸‚B¯¹K#ÈU^e
00015B80  04 64 1C 10 28 AE 5A B5 39 E7 73 BF 0F 47 D9 D3  .d..(®Zµ9çs¿.GÙÓ
00015B90  B5 CF FF D9 50 4B 03 04 14 00 06 00 08 00 00 00  µÏÿÙPK..........
00015BA0  21 00 AA 52 25 DF 23 06 00 00 8B 1A 00 00 15 00  !.ªR%ß#...‹.....
00015BB0  00 00 77 6F 72 64 2F 74 68 65 6D 65 2F 74 68 65  ..word/theme/the
00015BC0  6D 65 31 2E 78 6D 6C EC 59 4D 8B 1B 37 18 BE 17  me1.xmlìYM‹.7.¾.
00015BD0  FA 1F C4 DC 1D 7F CD F8 63 89 37 D8 63 3B 69 B3  ú.ÄÜ..Íøc‰7Øc;i³
00015BE0  9B 84 EC 26 25 47 79 46 9E 51 AC 19 19 49 DE 5D  ›„ì&%GyFžQ¬..IÞ]
00015BF0  13 02 25 39 16 0A A5 69 E9 A1 81 DE 7A 28 6D 03  ..%9..¥ié¡.Þz(m.
```

```
00000E40  29 A7 C7 34 BB F5 D9 B4 EE 8E 0F 72 84 CC 3C DC  )§Ç4»õÙ´îŽ.r„Ì<Ü
00000E50  8F 0F 46 7D 66 8E C7 C9 68 B7 23 48 40 B4 B4 35  ..F}fŽÇÉh·#H@´´5
00000E60  D6 5D 4A D4 1F 6A 40 6A B0 F1 B0 7C 09 DE 61 69  Ö]JÔ.j@j°ñ°|.Þai
00000E70  BF A4 23 16 A8 F8 26 7E 7E C5 56 D8 6E F7 43 CE  ¿¤#.¨ø&~~ÅVØn÷CÎ
00000E80  84 E2 E2 4B CF A6 E6 84 E2 46 6F 3D 7D 43 FF EE  „ââKÏ¦æ„âFo=}Cÿî
00000E90  AE 1D 37 0E F5 8C C2 29 A6 C2 16 02 EA CF 4B 7D  ®.7.õŒÂ)¦Â..êÏK}
00000EA0  C2 85 AF F9 77 6B CF 71 07 F7 2B 43 45 7B 46 85  Â…¯ùwkÏq.÷+CE{F…
00000EB0  F4 88 B8 25 DC 6C 73 11 DD AB 9F 6D 71 FD 17 A6  ôˆ¸%Üls.Ý«Ÿmqý.¦
00000EC0  F0 C7 24 81 72 9E 54 89 F6 DE C1 38 D4 2B 2C F8  ðÇ$.ržT‰öÞÁ8Ô+,ø
00000ED0  95 D3 66 A7 5B 52 3A 2C C1 73 A5 A4 82 D1 75 E7  •Óf§[R:,Ás¥¤‚Ñuç
00000EE0  DA 39 5D DF F5 2B 99 6F CD 96 92 67 12 95 70 32  Ú9]ßõ+™oÍ–’g.•p2
00000EF0  F2 DD 5C 6B B7 D5 2D 16 CE 77 BB E3 84 AE 2C 46  òÝ\k·Õ-.Îw»ã„®,F
00000F00  FD 03 26 AC F1 C3 28 9A 97 46 91 22 A4 F2 4C 21  ý.&¬ñÃ(š—F‘"¤òL!
00000F10  80 50 54 F7 7B 69 82 89 BE 19 DE D5 E4 85 EE 2F  €PT÷{i‚‰¾.ÞÕä…î/
00000F20  E8 D1 FF 00 00 00 FF FF 03 00 50 4B 03 04 0A 00  èÑÿ...ÿÿ..PK....
00000F30  00 00 00 00 00 00 21 00 57 80 15 44 36 4C 01 00  ......!.W€.D6L..
00000F40  36 4C 01 00 16 00 00 00 77 6F 72 64 2F 6D 65 64  6L......word/med
00000F50  69 61 2F 69 6D 61 67 65 31 2E 6A 70 65 67 FF D8  ia/image1.jpegÿØ
00000F60  FF E0 00 10 4A 46 49 46 00 01 01 01 00 DC 00 DC  ÿà..JFIF.....Ü.Ü
00000F70  00 00 FF DB 00 43 00 02 01 01 01 01 01 02 01 01  ..ÿÛ.C..........
00000F80  01 02 02 02 02 02 04 03 02 02 02 02 05 04 04 03  ................
00000F90  04 06 05 06 06 06 05 06 06 06 07 09 08 06 07 09  ................
00000FA0  07 06 06 08 0B 08 09 0A 0A 0A 0A 0A 06 08 0B 0C  ................
00000FB0  0B 0A 0C 09 0A 0A 0A FF DB 00 43 01 02 02 02 02  .......ÿÛ.C.....
00000FC0  02 02 05 03 03 05 0A 07 06 07 0A 0A 0A 0A 0A 0A  ................
00000FD0  0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A  ................
00000FE0  0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A  ................
00000FF0  0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A FF C0 00 11  ............ÿÀ..
00001000  08 02 1C 01 68 03 01 22 00 02 11 01 03 11 01 FF  ....h.."......ÿ
00001010  C4 00 1F 00 00 01 05 01 01 01 01 01 01 00 00 00  Ä...............
00001020  00 00 00 00 00 01 02 03 04 05 06 07 08 09 0A 0B  ................
00001030  FF C4 00 B5 10 00 02 01 03 03 02 04 03 05 05 04  ÿÄ.µ............
```

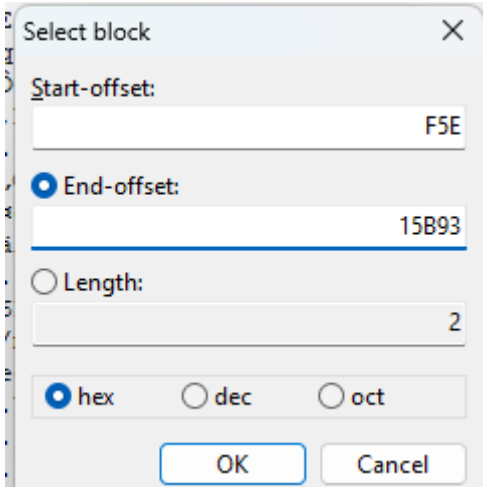
– Select hex from header to tail

- (0F5E)16=(3934)10
- (15B93)16=(88979)10

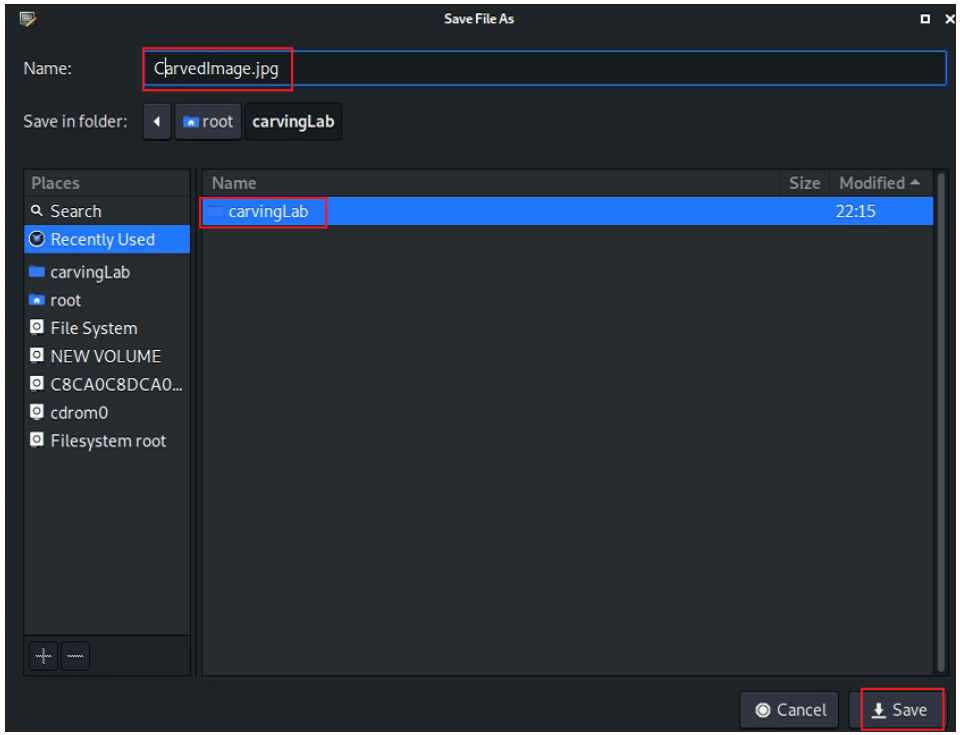


– Copy the selection

– Paste the selection

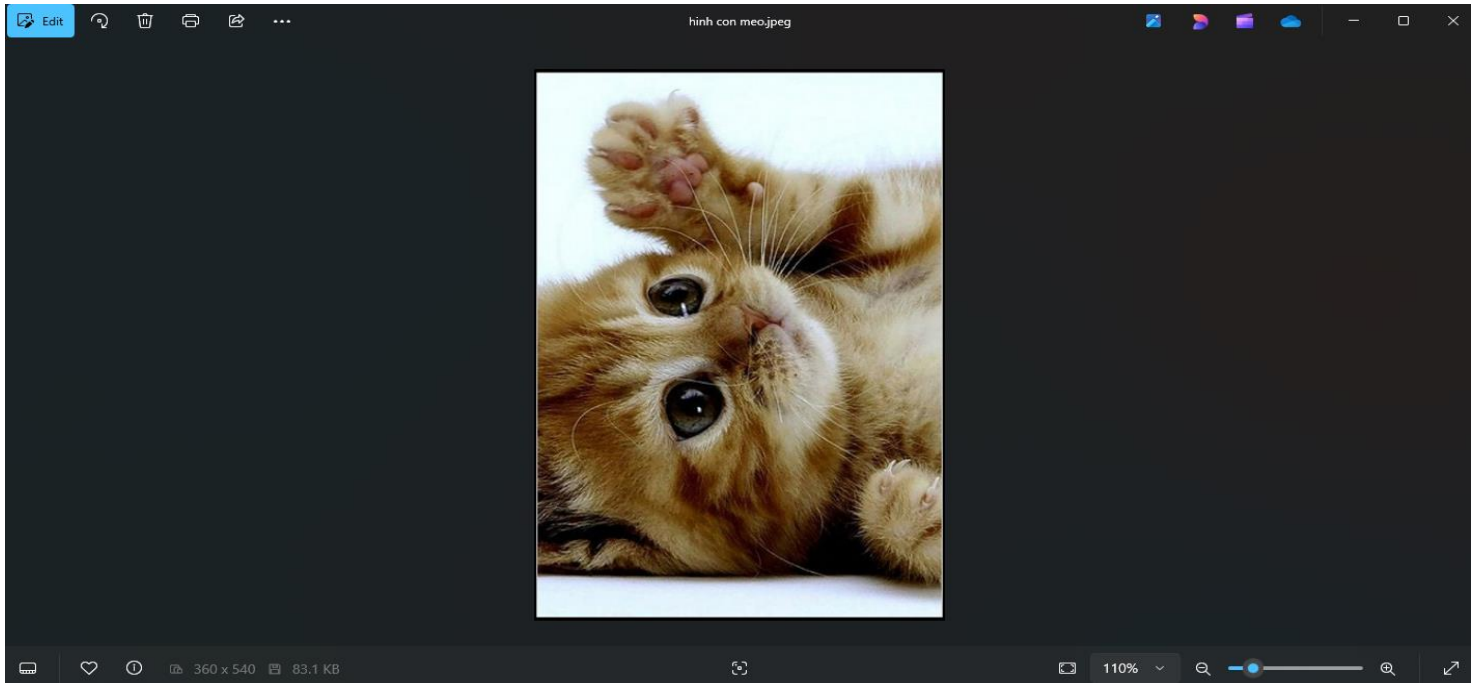


– Save the image



Step 4.

– Show the carved image

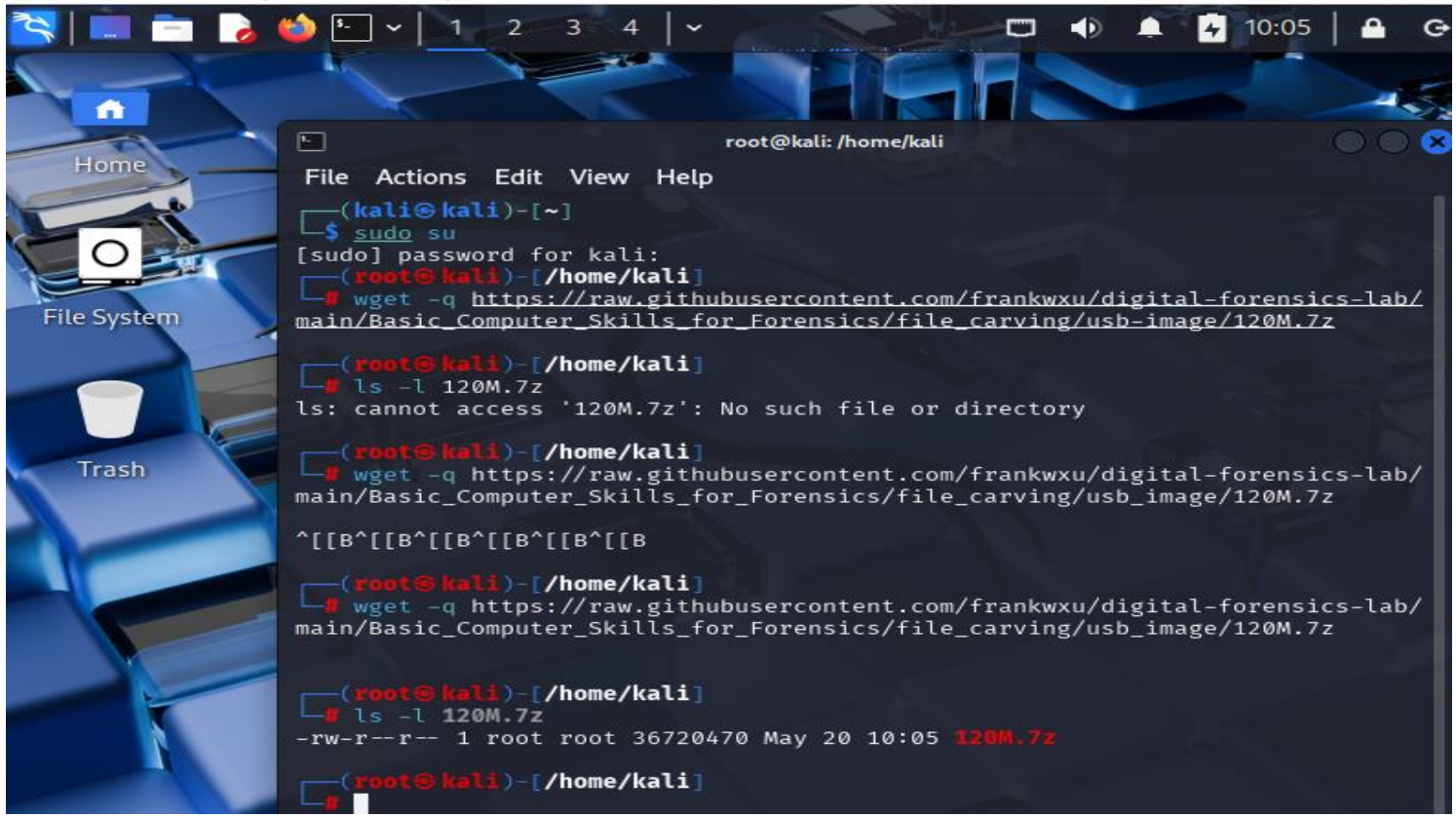


2. Carving/Recovering a USB image

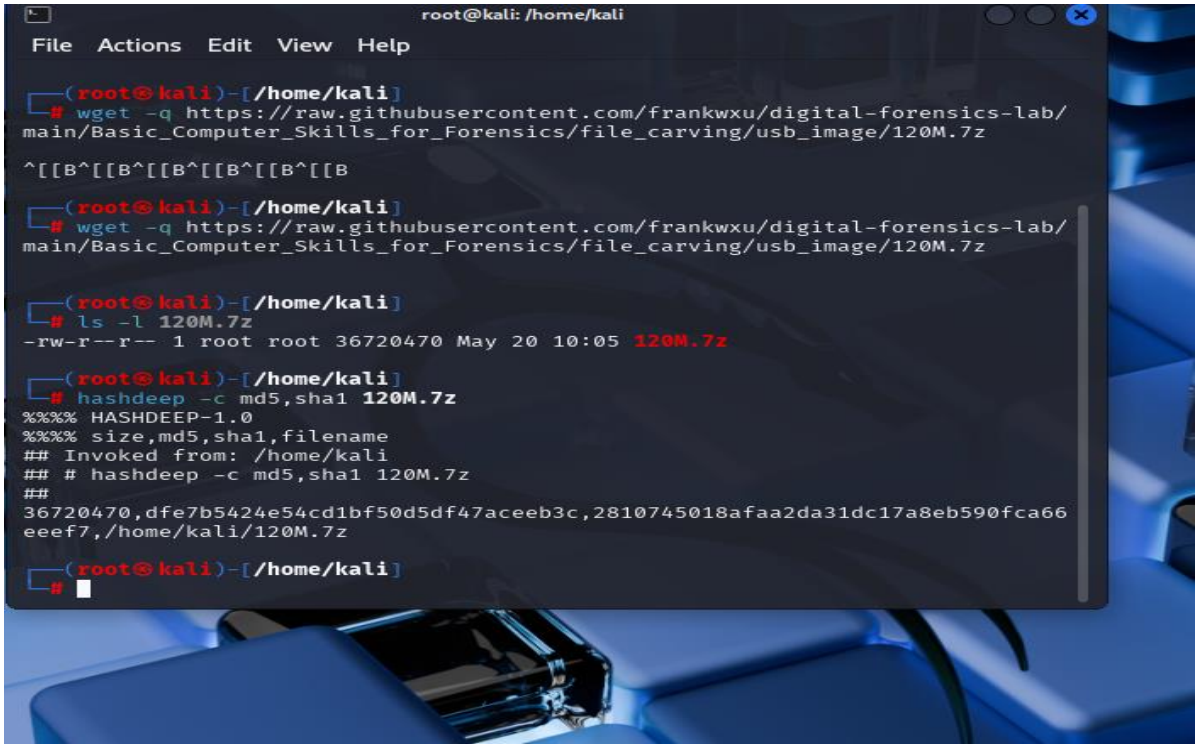
- Prepare a USB image for file carving

Step 1.

- Download the zipped USB image



– Compute hashes



– List the content of the zipped file

–

– List the content of the zipped file

```
File  Actions  Edit  View  Help
┌──(root㉿kali)-[/home/kali]
└─# 7z e 120M.7z


7-Zip 24.09 (x64) : Copyright (c) 1999-2024 Igor Pavlov : 2024-11-29
 64-bit locale=en_US.UTF-8 Threads:2 OPEN_MAX:1024, ASM

Scanning the drive for archives:
1 file, 36720470 bytes (36 MiB)

Extracting archive: 120M.7z
--
Path = 120M.7z
Type = 7z
Physical Size = 36720470
Headers Size = 214
Method = LZMA2:24
Solid = +
Blocks = 1

Everything is Ok

Folders: 1
Files: 2
Size:       124782229
Compressed: 36720470
```

– Verify the hashes

```
┌──(root㉿kali)-[/home/kali]
└─# hashdeep -c md5,sha1 usb_fat_carving.001

%%%% HASHDEEP-1.0
%%%% size,md5,sha1,filename
## Invoked from: /home/kali
## # hashdeep -c md5,sha1 usb_fat_carving.001
##
124780544,ba4a1d0ba49f4a6667b00a3b3e85e604,bcc2d49fd49c9521ecb1739f6542c6bf32
7375ef,/home/kali/usb_fat_carving.001
```

```
┌──(root㉿kali)-[/home/kali]
└─# cat usb_fat_carving.001.txt | grep checksum

 MD5 checksum:    ba4a1d0ba49f4a6667b00a3b3e85e604
 SHA1 checksum:   bcc2d49fd49c9521ecb1739f6542c6bf327375ef
 MD5 checksum:    ba4a1d0ba49f4a6667b00a3b3e85e604 : verified
 SHA1 checksum:   bcc2d49fd49c9521ecb1739f6542c6bf327375ef : verified
```

Step 2.

- Exam the content of the USB
- Display partitions

```
┌──(root㉿kali)-[/home/kali]
└─# fdisk -l usb_fat_carving.001

Disk usb_fat_carving.001: 119 MiB, 124780544 bytes, 243712 sectors
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: 0×a1159a00

Device                Boot Start    End Sectors   Size Id Type
usb_fat_carving.001p1 *      128 243711  243584 118.9M  e W95 FAT16 (LBA)
```

– Find deleted files

```
r/r 3:  USB         (Volume Label Entry)
d/d 6:  System Volume Information
+ r/r 519:      WPSettings.dat
r/r * 7:        _est
r/r 10: .dropbox.device
d/d * 13:       old_File_Carving_files
+ r/r * 711:    CarvedImage.jpg
+ r/r * 714:    File_carving.docx
+ r/r * 717:    File_Carving.pptx
+ r/r * 720:    i-love-cats-cute-cats.jpg
r/r * 15:       B_ub_poe4.bmp
r/r * 18:       B_zoom-eubie-mono.bmp
r/r * 21:       Ballardlab8.java
r/r * 24:       brittLab10.java
r/r * 27:       DO_example.doc
r/r * 30:       DO_example2.doc
r/r * 33:       G_BuiltForThis.gif
r/r * 35:       G_zoom-sc.gif
r/r * 37:       H_Form.html
r/r * 39:       H_hello.html
r/r * 41:       J_ub_law.jpg
r/r * 44:       J_ub_night.jpg
r/r * 47:       nps-2008-jean_outlook.pst
r/r * 50:       P_CAS-zoom-6.png
r/r * 53:       P_MSB_1_zoom.png
```

– Decide which file types need to carve

```
# GIF and JPG files (very common)
        gif     y       5000000         \x47\x49\x46\x38\x37\x61        \x00>
        gif     y       5000000         \x47\x49\x46\x38\x39\x61        \x00>
        jpg     y       5242880         \xff\xd8\xff???Exif             \xff>
        jpg     y       5242880         \xff\xd8\xff???JFIF             \xff>
#
#
# PNG
        png     y       20000000        \x50\x4e\x47?   \xff\xfc\xfd\xfe
#
```

- Save it and quit!
- Show help

```
┌──(root㉿kali)-[/home/kali]
└─# scalpel -h
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.
Carves files from a disk image based on file headers and footers.

Usage: scalpel [-b] [-c <config file>] [-d] [-h|V] [-i <file>]
                 [-m blocksize] [-n] [-o <outputdir>] [-O num] [-q clustersiz
e]
                 [-r] [-s num] [-t <blockmap file>] [-u] [-v]
                 <imgfile> [<imgfile>] ...

-b  Carve files even if defined footers aren't discovered within
    maximum carve size for file type [foremost 0.69 compat mode].
-c  Choose configuration file.
-d  Generate header/footer database; will bypass certain optimizations
    and discover all footers, so performance suffers.  Doesn't affect
    the set of files carved.  **EXPERIMENTAL**
-h  Print this help message and exit.
-i  Read names of disk images from specified file.
-m  Generate/update carve coverage blockmap file.  The first 32bit
    unsigned int in the file identifies the block size. Thereafter
    each 32bit unsigned int entry in the blockmap file corresponds
    to one block in the image file.  Each entry counts how many
    carved files contain this block. Requires more memory and
    disk.  **EXPERIMENTAL**
-n  Don't add extensions to extracted files.
```

Step 3.

- Carving the USB image

```
└─# scalpel usb_fat_carving.001 -o output

Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.

Opening target "/home/kali/usb_fat_carving.001"

Image file pass 1/2.
usb_fat_carving.001: 100.0% |************************|  119.0 MB    00:00 ETA
Allocating work queues ...
Work queues allocation complete. Building carve lists ...
Carve lists built.  Workload:
gif with header "\x47\x49\x46\x38\x37\x61" and footer "\x00\x3b" --> 0 files
gif with header "\x47\x49\x46\x38\x39\x61" and footer "\x00\x3b" --> 0 files
jpg with header "\xff\xd8\xff\x3f\x3f\x3f\x45\x78\x69\x66" and footer "\xff\x
d9" --> 8 files
jpg with header "\xff\xd8\xff\x3f\x3f\x3f\x4a\x46\x49\x46" and footer "\xff\x
d9" --> 9 files
png with header "\x50\x4e\x47\x3f" and footer "\xff\xfc\xfd\xfe" --> 0 files
Carving files from image.
Image file pass 2/2.
usb_fat_carving.001: 100.0% |************************|  119.0 MB    00:00 ETA
Processing of image file complete. Cleaning up ...
Done.
Scalpel is done, files carved = 17, elapsed = 1 seconds.
```

– Show carved files

```
┌──(root㉿kali)-[/home/kali]
└─# tree output
output
├── audit.txt
├── jpg-2-0
│   ├── 00000000.jpg
│   ├── 00000001.jpg
│   ├── 00000002.jpg
│   ├── 00000003.jpg
│   ├── 00000004.jpg
│   ├── 00000005.jpg
│   ├── 00000006.jpg
│   └── 00000007.jpg
└── jpg-3-0
    ├── 00000008.jpg
    ├── 00000009.jpg
    ├── 00000010.jpg
    ├── 00000011.jpg
    ├── 00000012.jpg
    ├── 00000013.jpg
    ├── 00000014.jpg
    ├── 00000015.jpg
    └── 00000016.jpg
```

– Show audit log

```
Started at Tue May 20 10:22:48 2025
Command line:
scalpel usb_fat_carving.001 -o output

Output directory: /home/kali/output
Configuration file: /etc/scalpel/scalpel.conf

Opening target "/home/kali/usb_fat_carving.001"

The following files were carved:
File             Start                Chop            Length          Extra
cted From
00000010.jpg     3796992              NO          3148500             usb_f
at_carving.001
00000009.jpg      425822              NO          4875802             usb_f
at_carving.001
00000008.jpg      335872              NO          4965752             usb_f
at_carving.001
00000002.jpg     6938624              NO             6868             usb_f
at_carving.001
00000001.jpg     5285888              NO          1659604             usb_f
at_carving.001
00000000.jpg     3897344              NO          3048148             usb_f
at_carving.001
00000004.jpg    15427584              NO          4223822             usb_f
at_carving.001
00000003.jpg    12881920              NO          4256310             usb_f
at_carving.001
```

Step 4.

- Display two carved jpg image


ImageMagick: 00000000.jpg
UNIVERSITY OF
BALTIMORE


ImageMagick: 00000008.jpg