

Lab 2: Data Carving

Group: CyberSec_N00b

Member:

- *Huỳnh Ngọc Quang (SE181838)*
- *Hồ Tài Liên Vy Kha (SE181818)*
- *Hoàng Kim Long (DE180860)*
- *Phạm Thành Long (SE181692)*
- *Nguyễn Lê Hoàng Thông (SE182533)*

1. Extracting images from a corrupted Word document

Step 1.

The screenshot shows a terminal window titled 'Kali-quanghn' running on a Kali Linux VM. The terminal session is as follows:

```
kali@kali:~/carvingLab
mkdir carvingLab
cd carvingLab
wget -q https://github.com/frankwxu/digital-forensics-lab/raw/main/Basic_Computer_Skills_for_Forensics/file_carving/File_Carving_Manually/File_carving.docx
ls -l
total 96
-rw-r--r-- 1 kali kali 97702 Sep 13 09:13 File_carving.docx
md5sum File_carving.docx
9b1439ffd4a30d86a08299e659210590  File_carving.docx
```

Step 2.

Kali-quingh - VMware Workstation

File Edit View VM Jobs Help

Library Type here to search

My Computer OS Windows 10 Lite Black Server UbuntuServer Windows Server 2012 Windows Server 2019 Windows Server 2019.2 Windows Server 2019.2 EVS-rq Kali-quingh Windows 10x64 Ubuntu Docker MalAnalyst Win2008-DC-quingh Ubuntu IAM

Home X Kali-quingh ~ carvingLab

```
sudo apt install bless
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bless is already the newest version (0.6.3-1).
The following packages were automatically installed:
  bluez-firmware certipy-ad firmware-ath9k-htc
  firmware-iwlwifi firmware-libertas firmware-regulatory
  kali-linux-firmware libkf5config-bin libnls-de
  python3-cryptography37 python3-docker python3-distro
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

~ carvingLab

```
bless File_carving.docx
Failed to open plugins directory: Could not find a part of the path '/home/kali/.config/bless/plugins'.
Failed to open plugins directory: Could not find a part of the path '/home/kali/.config/bless/plugins'.
Failed to open plugins directory: Could not find a part of the path '/home/kali/.config/bless/plugins'.
Could not find file "/home/kali/.config/bless/export_patterns"
```

File Edit View Search Tools Help

/home/kali/carvingLab/File_carving.docx - Bless

New Open Save Undo Redo Cut Copy Paste Find Find and Replace

File_carving.docx

Signed 8 bit:	80	Signed 32 bit:	1347093252	Hexadecimal:	50 4B 03 04
Unsigned 8 bit:	80	Unsigned 32 bit:	1347093252	Decimal:	080 075 003 004
Signed 16 bit:	20555	Float 32 bit:	1.362389E+10	Octal:	120 113 003 004
Unsigned 16 bit:	20555	Float 64 bit:	6.255501067542E+78	Binary:	01010000 01001011 000

Show little endian decoding Show unsigned as hexadecimal ASCII Text PK

Offset: 0x0 / 0x17da5 Selection: None INS

Step 3.

- Search file header start offset – 0F5E

Kali-quingh - VMware Workstation

File Edit View VM Jobs Help

Library Type here to search

My Computer OS Windows 10 Lite Black Server UbuntuServer Windows Server 2012 Windows Server 2019 Windows Server 2019.2 Windows Server 2019.2 EVS-rq Kali-quingh Windows 10x64 Ubuntu Docker MalAnalyst Win2008-DC-quingh Ubuntu IAM

Home X Kali-quingh ~ carvingLab

File Edit View Search Tools Help

/home/kali/carvingLab/File_carving.docx - Bless

Re: File_carving.docx

Search for: **0F5E**

1. Find button

2. Search term '0F5E'

3. Find Next button

4. Found offset 0x6f5e

5. Offset 0x6f5e highlighted in red

File Edit View Search Tools Help

/home/kali/carvingLab/File_carving.docx - Bless

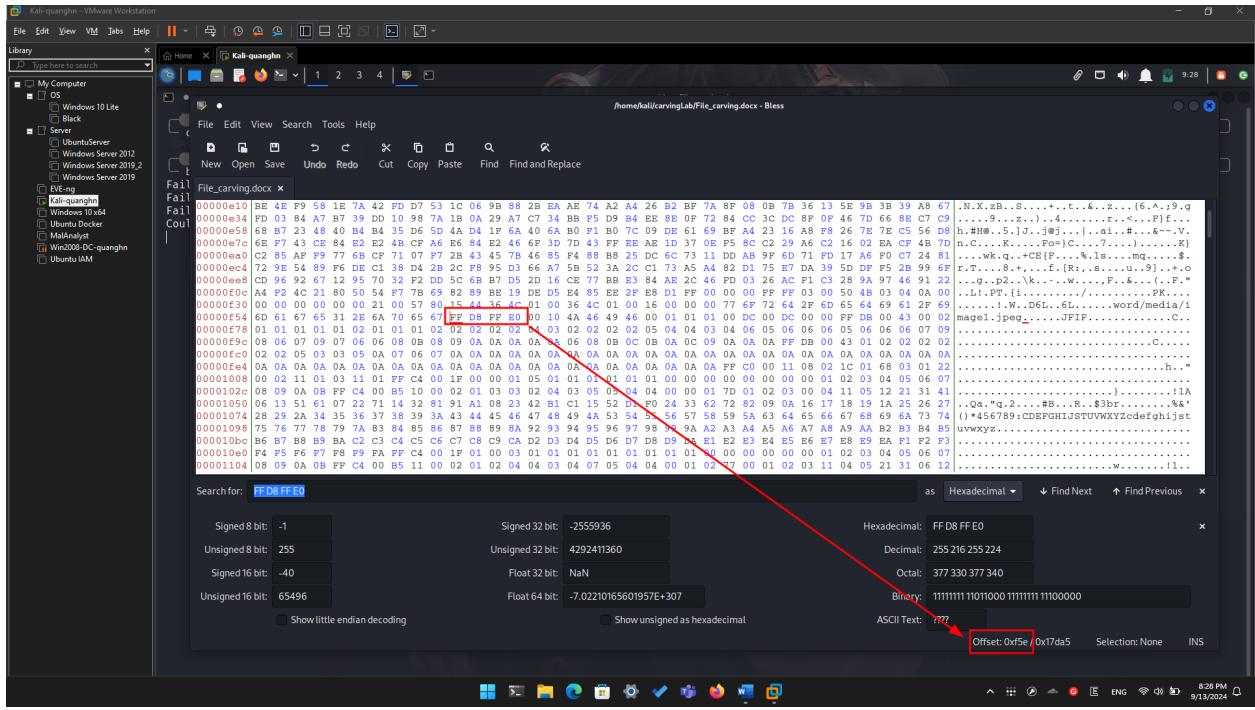
New Open Save Undo Redo Cut Copy Paste Find Find and Replace

File_carving.docx

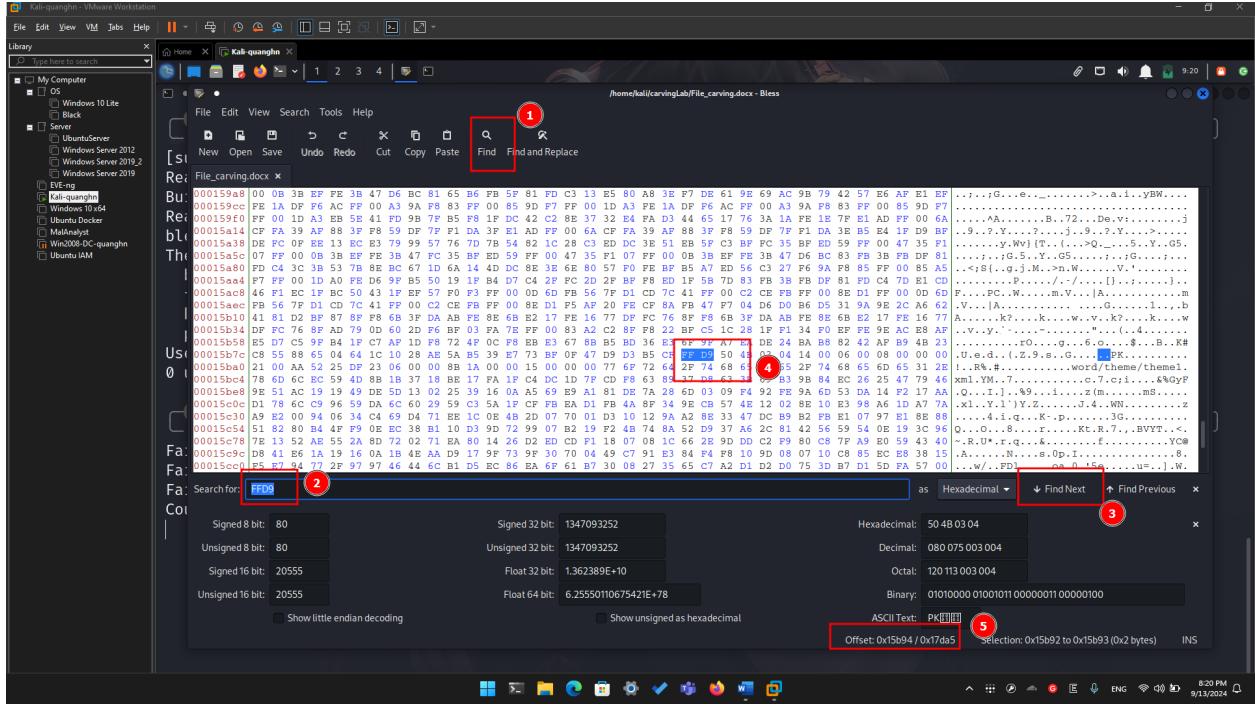
Signed 8 bit:	0	Signed 32 bit:	1067590	Hexadecimal:	00 10 4A 46
Unsigned 8 bit:	0	Unsigned 32 bit:	1067590	Decimal:	000 016 074 070
Signed 16 bit:	16	Float 32 bit:	1.496012E-39	Octal:	000 020 112 106
Unsigned 16 bit:	16	Float 64 bit:	2.26542209384351E-308	Binary:	000 000000 00010000 01001010 01000110

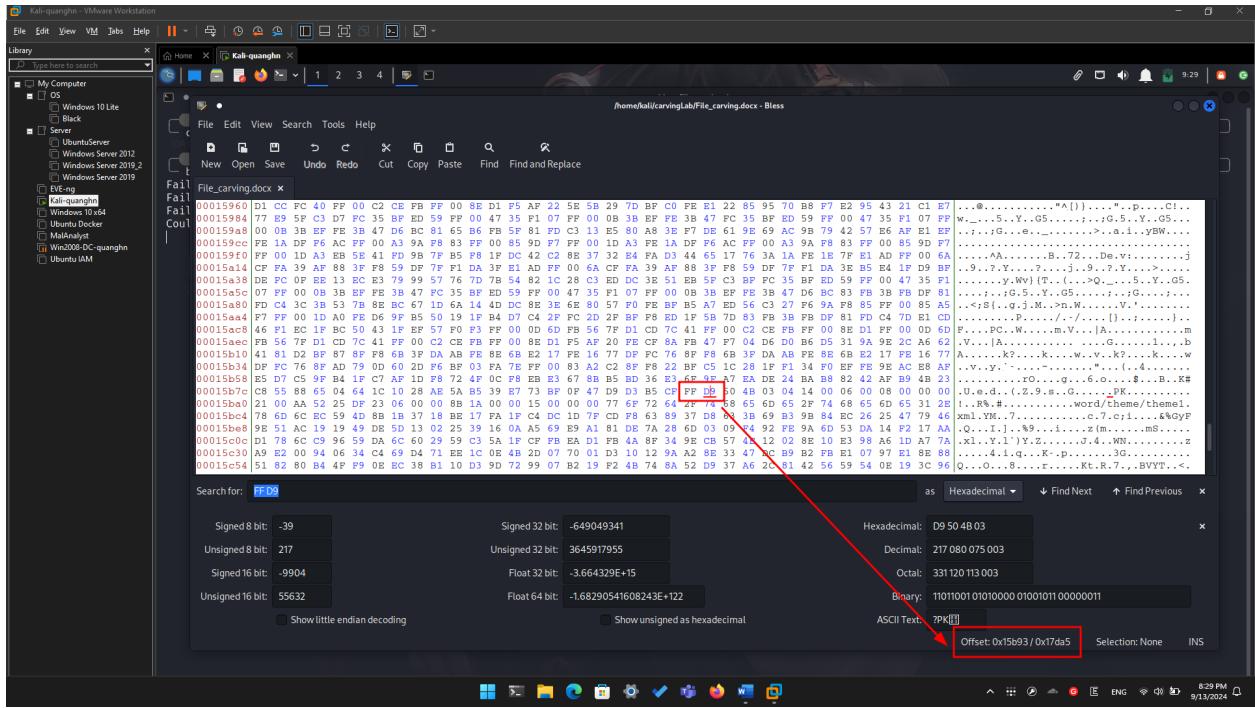
Show little endian decoding Show unsigned as hexadecimal ASCII Text PK

Offset: 0x6f5e / 0x17da5 Selection: 0x6f5e (0x4 bytes) INS

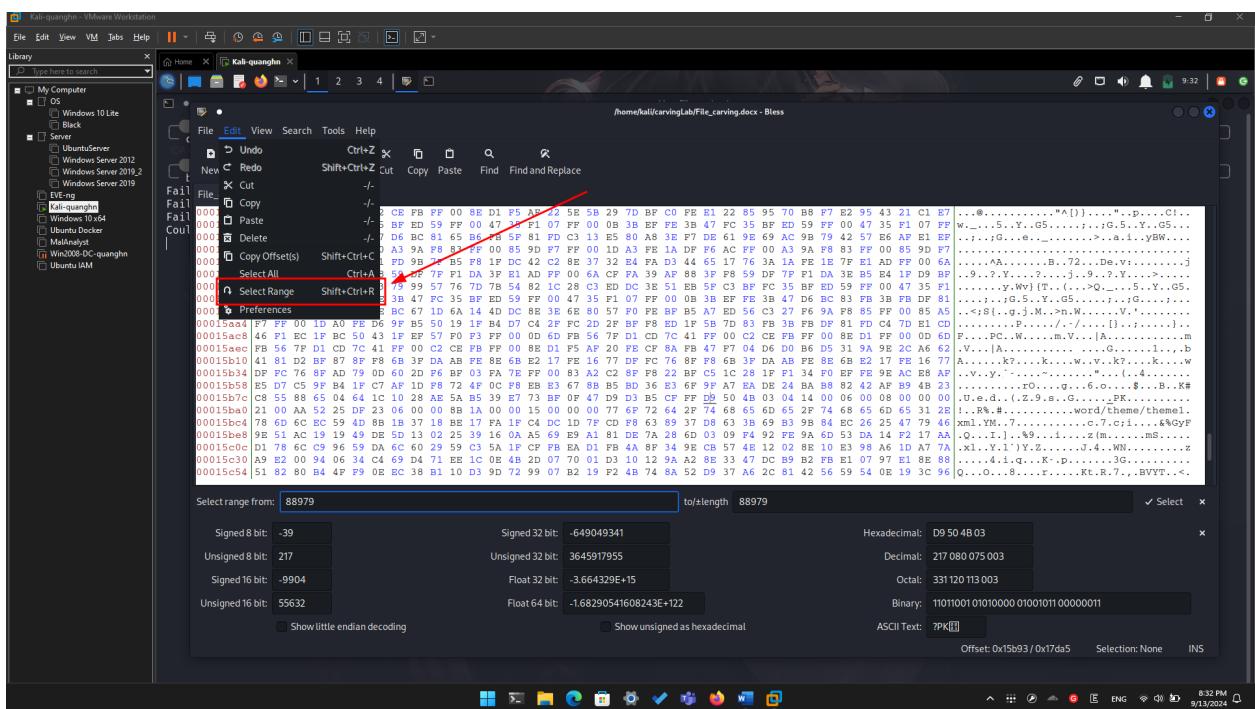


– Search file trailer ends offset – 15B93

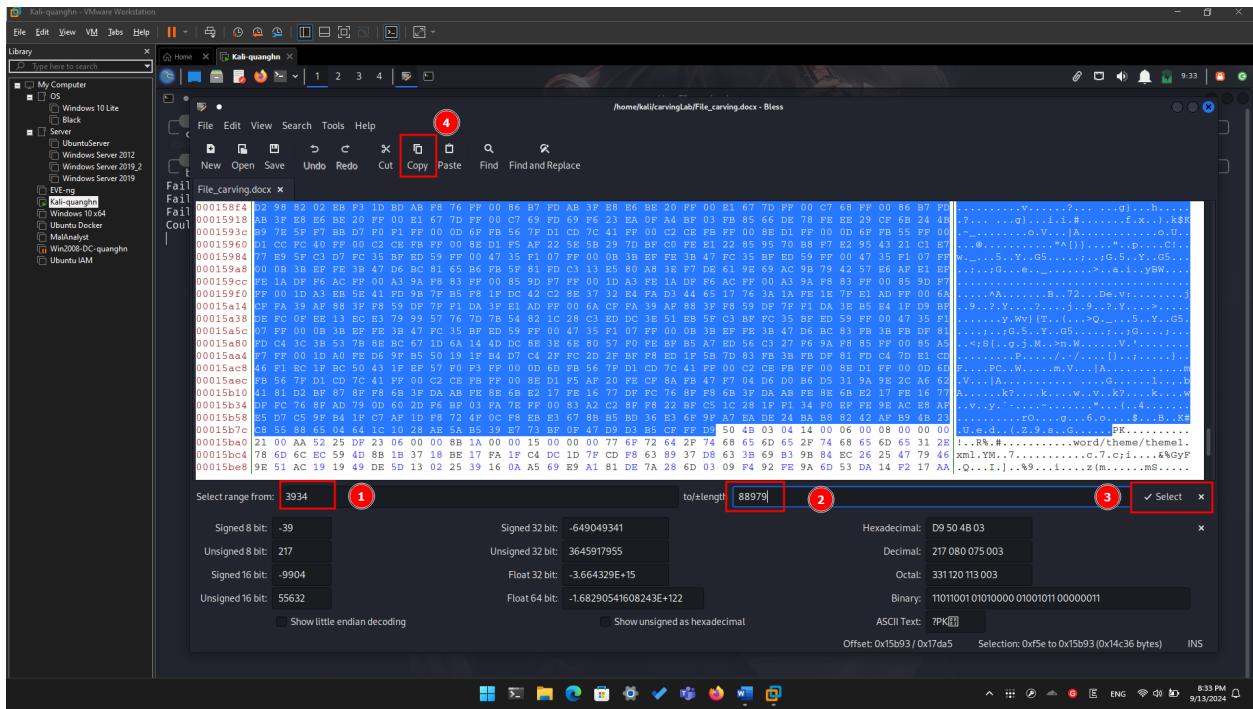




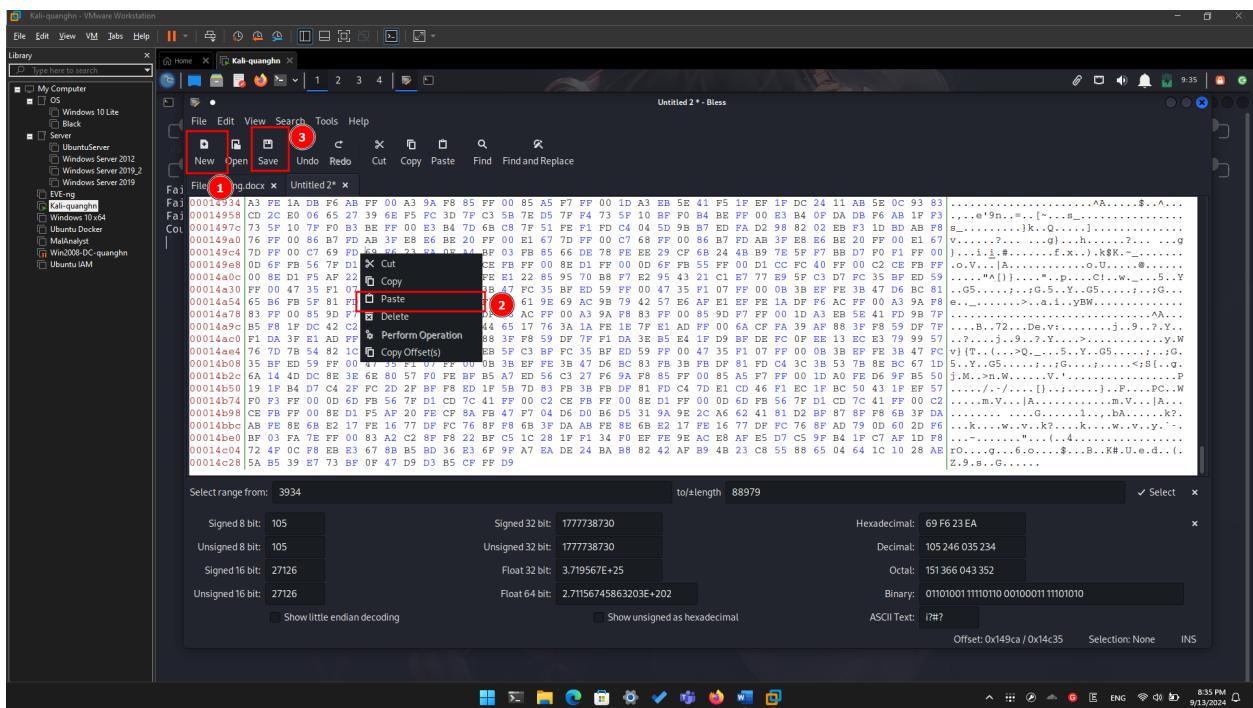
- Select hex from header to tail
- $(0F5E)_{16} = (3934)_{10}$
- $(15B93)_{16} = (88979)_{10}$



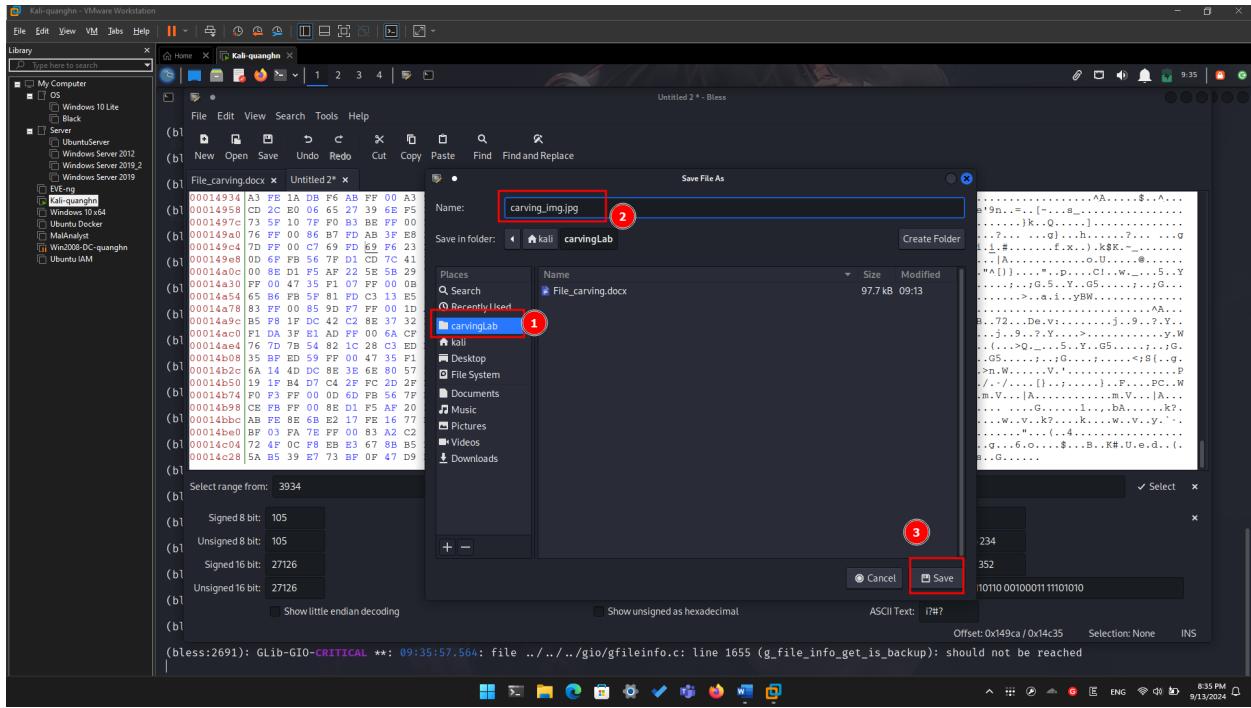
- Copy the selection



- Paste the selection

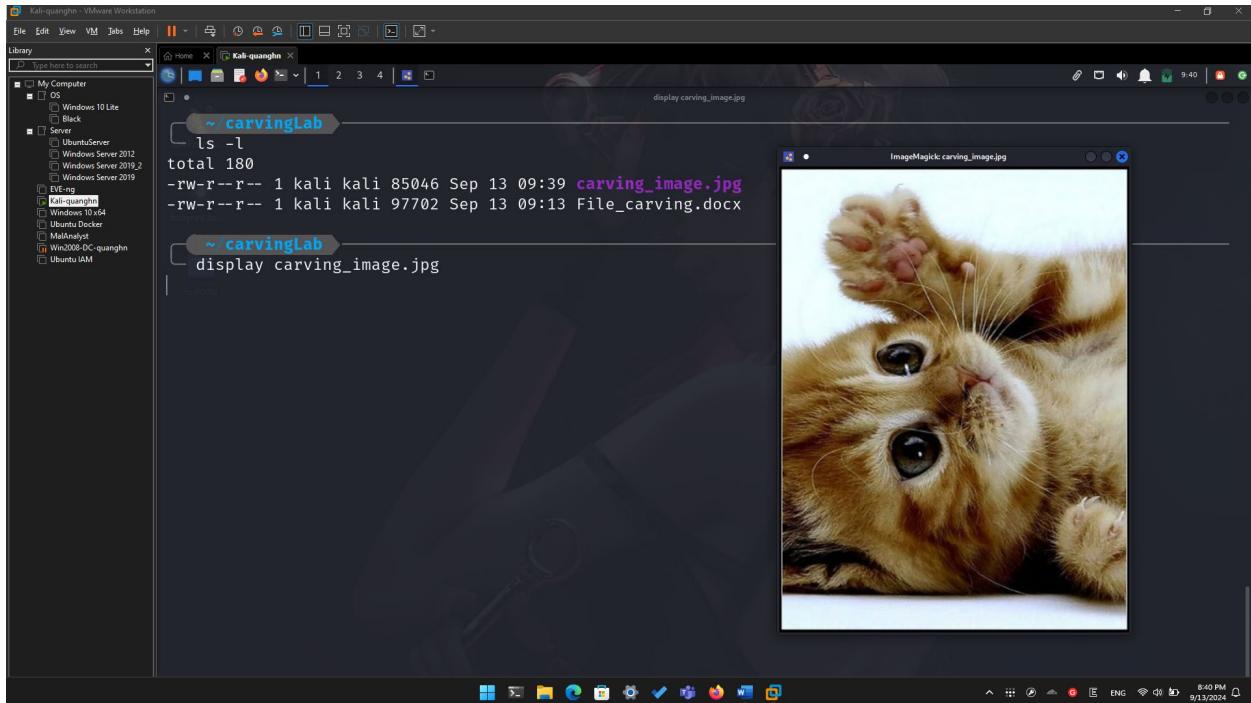


- Save the image



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

A screenshot of a Kali Linux terminal window titled "Kali-quanghn - VMware Workstation". The terminal shows the following command sequence:

```
wget -q https://github.com/frankwxu/digital-forensics-lab/raw/main/Basic_Computer_Skills_for_Forensics/file_carving/usb_image/120M.7z
ls -l 120M.7z
-rw-r--r-- 1 kali kali 36720470 Sep 13 09:42 120M.7z
```

The terminal interface includes a sidebar with a "Library" section containing various OS and Server options. The status bar at the bottom right shows the date and time as "9/13/2024 8:42 PM".

- Compute hashes

A screenshot of a Kali Linux terminal window titled "Kali-quanghn - VMware Workstation". The terminal shows the following command sequence:

```
hashdeep -c md5,sha1 120M.7z
%%%% HASHDEEP-1.0
%%%% size,md5,sha1,filename
## Invoked from: /home/kali/carvingLab
## $ hashdeep -c md5,sha1 120M.7z
##
36720470,dfc7b5424e54cd1bf50d5df47aceeb3c,2810745018afaa2da31dc17a8eb590fca66eeef7,/home/kali/carvingLab/120M.7z
```

The terminal interface includes a sidebar with a "Library" section containing various OS and Server options. The status bar at the bottom right shows the date and time as "9/13/2024 8:42 PM".

- List the content of the zipped file

```

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 12th Gen Intel(R) Core(TM) i7-12700H (906A3),AES-NI)

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

Listing archive: 120M.7z

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

          Date      Time     Attr            Size   Compressed  Name
-----  -- 2021-09-22 22:59:31 D...A           0       0  120M
2021-09-22 22:59:31 ....A 124780544    36720256 120M/usb_fat_carving.001
2021-09-22 22:59:32 ....A        1685          1685 120M/usb_fat_carving.001.txt
-----  -- 2021-09-22 22:59:32           124782229  36720256 2 files, 1 folders

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 12th Gen Intel(R) Core(TM) i7-12700H (906A3),AES-NI)

```

- Extract the content of the zipped file

```

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 12th Gen Intel(R) Core(TM) i7-12700H (906A3),AES-NI)

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

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,128 CPUs 12th Gen Intel(R) Core(TM) i7-12700H (906A3),AES-NI)

```

- Verify the hashes

```
hashdeep -c md5,sha1 usb_fat_carving.001
%% HASHDEEP-1.0
%% size,md5,sha1,filename
## Invoked from: /home/kali/carvingLab
## $ hashdeep -c md5,sha1 usb_fat_carving.001
##
124780544,ba4a1d0ba49f4a6667b00a3b3e85e604,bcc2d49fd49c9521ecb1739f6542c6bf327375ef,/home/kali/carvingLab/usb_fat_carving.001

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

```
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: 0xa1159a00

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

- Find deleted files

```

fls -o 128 usb_fat_carving.001
r/r 3: USB (Volume Label Entry)
d/d 6: System Volume Information
r/r * 7: _est
r/r 10: .dropbox.device
d/d * 13: old_File_Carving_files
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_zoom.png
r/r * 57: pd_Evidence_search_techniques.pdf
r/r * 61: pd_Forensic_Report_Template.pdf
r/r * 64: pp_Number_Systems.pptx
r/r * 67: pp_one_page.pptx
r/r 69: readme.docx
r/r 70: readme.txt
r/r * 71: _tf_1.rtf
r/r * 74: T_Eubie-iphone-5-8.tiff
r/r * 78: T_youknowus-iphone-5-8.tiff

```

- Decide which file types need to carve

```

19 # Here is an example of how to use the no extension option. Any files
18 # beginning with the string "FOREMOST" are carved and no file extensions
17 # are used. No footer is defined and the max carve size is 1000 bytes.
16 #
15 #      NONE      y      1000      FOREMOST
14 #
13 #
12 # GRAPHICS FILES
11 #
10 #
9 #
8 # AOL ART files
7 #   art y  150000  \x4a\x47\x04\x0e  \xcf\xc7\xcb
6 #   -art-y  150000  \x4a\x47\x03\x0e  \xd0\xcb\x00\x00
5 #
4 # GIF and JPG files (very common)
3 #   gif y  5000000  \x47\x49\x46\x38\x37\x61  \x00\x3b
2 #   gif y  5000000  \x47\x49\x40\x38\x39\x61  \x00\x3b
   - jpg y  5242880  \xff\xd8\xff??Exif  \xff\xd9  REVERSE
88   - jpg y  5242880  \xff\xd8\xff??JFIF  \xff\xd9  REVERSE
#
2 #
3 # PNG  uncomment
4 #   png y  2000000  \x50\x4e\x47\x0d\x0a\x1a\x00\x00
5 #
6 #
7 # BMP  (used by MSWindows, use only if you have reason to think there are
8 #       BMP files worth digging for. This often kicks back a lot of false
9 #       positives

```

INSERT > /etc/scalpel/scalpel.conf[+] conf utf-8 /GRAPHICS...[1/1] 36% 88/240= %: W10: Warning: Changing a readonly file -- INSERT --

- Save it and quit!
- Show help

```
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 clustersize]
              [-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.
-o Set output directory for carved files.
-O Don't organize carved files by type. Default is to organize carved files
  into subdirectories.
-p Perform image file preview; audit log indicates which files
  would have been carved, but no files are actually carved.
```

Step 3.

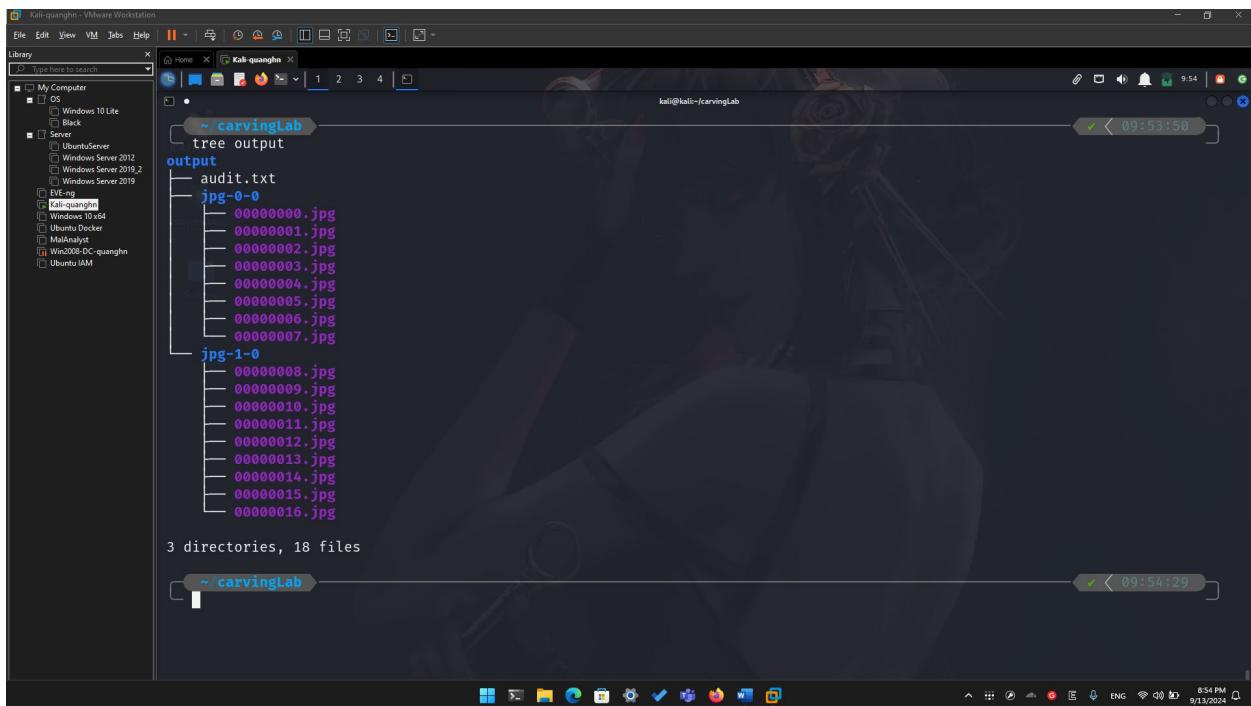
- Carving the USB image

```
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/carvingLab/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:
jpg with header "\xff\xd8\xff\x3f\x3f\x3f\x45\x78\x69\x66" and footer "\xff\xd9" → 8 files
jpg with header "\xff\xd8\xff\x3f\x3f\x3f\x4a\x46\x49\x46" and footer "\xff\xd9" → 9 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



– Show audit log

```

~/carvingLab
cat output/audit.txt
Scalpel version 1.60 audit file
Started at Fri Sep 13 09:53:49 2024
Command line:
scalpel usb_fat_carving.001 -o output
Configuration file: /etc/scalpel/scalpel.conf

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

The following files were carved:
File          Start          Chop          Length          Extracted From
00000010.jpg 3796992        NO           3148500        usb_fat_carving.001
00000009.jpg 425822         NO           4875802        usb_fat_carving.001
00000008.jpg 335872         NO           4965752        usb_fat_carving.001
00000002.jpg 6938624         NO           6868          usb_fat_carving.001
00000001.jpg 5285888         NO           1659604        usb_fat_carving.001
00000000.jpg 3897344         NO           3048148        usb_fat_carving.001
00000004.jpg 15427584        NO           4223822        usb_fat_carving.001
00000003.jpg 12881920        NO           4256310        usb_fat_carving.001
00000006.jpg 19638272        NO           5229050        usb_fat_carving.001
00000005.jpg 17121280        NO           4248530        usb_fat_carving.001
00000007.jpg 21358592        NO           5183267        usb_fat_carving.001
00000016.jpg 36671488        NO           2889262        usb_fat_carving.001
00000015.jpg 36571136        NO           2989614        usb_fat_carving.001
00000014.jpg 36393610        NO           3167140        usb_fat_carving.001
00000013.jpg 36352448        NO           3208302        usb_fat_carving.001
00000012.jpg 35590996        NO           3969754        usb_fat_carving.001
00000011.jpg 35350242        NO           4210508        usb_fat_carving.001

```

Step 4.

– Display two carved jpg image

