Task-1

Hash Functions on Windows and Linux

Using Command Prompt to view image hashes

For single image use command “**certutil -hashfile img.jfif sha384**”

For all images use command “**FORFILES /M *.jfif /C "cmd /C certutil -hashfile @FILE"**”

```
Command Prompt
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\shehr>cd ../../tools

C:\tools>certutil -hashfile img.jfif sha384
SHA384 hash of img.jfif:
a2f1ed5d1caa8a0aabf92f22f4978b61d94cf647256e4fc63914de4fdddd517604192a840c86b730e7f2a5df8af291a6
CertUtil: -hashfile command completed successfully.

C:\tools>FORFILES /M *.jfif /C "cmd /C certutil -hashfile @FILE"

SHA1 hash of img.jfif:
16b2fdec0c053b704ebb7cd27d89434c8b2797dc
CertUtil: -hashfile command completed successfully.
SHA1 hash of img1.jfif:
5329826911901ac091490b85794a0815e91ca7e5
CertUtil: -hashfile command completed successfully.
SHA1 hash of img2.jfif:
edbf767fca255c1418245ee758868c971eb1c40b
CertUtil: -hashfile command completed successfully.
SHA1 hash of img3.jfif:
36acbce794cd3749ff8eaebf95e6b5a92c80ef13
CertUtil: -hashfile command completed successfully.
SHA1 hash of img4s.jfif:
9452b1b5d4e45ba00b0ebfc088cc847312614c62
CertUtil: -hashfile command completed successfully.
```

Using Windows-Powershell to view image hashes

For all images use command “**Get-FileHash -A SHA1 *.jfif**”

```
Administrator: Windows PowerShell
PS C:\tools> Get-FileHash -A SHA1 *.jfif

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA1            16B2FDEC0C053B704EBB7CD27D89434C8B2797DC                               C:\tools\img.jfif
SHA1            5329826911901AC091490B85794A0815E91CA7E5                               C:\tools\img1.jfif
SHA1            EDBF767FCA255C1418245EE758868C971EB1C40B                               C:\tools\img2.jfif
SHA1            36ACBCE794CD3749FF8EAEBF95E6B5A92C80EF13                               C:\tools\img3.jfif
SHA1            9452B1B5D4E45BA00B0EBFC088CC847312614C62                               C:\tools\img4s.jfif
```

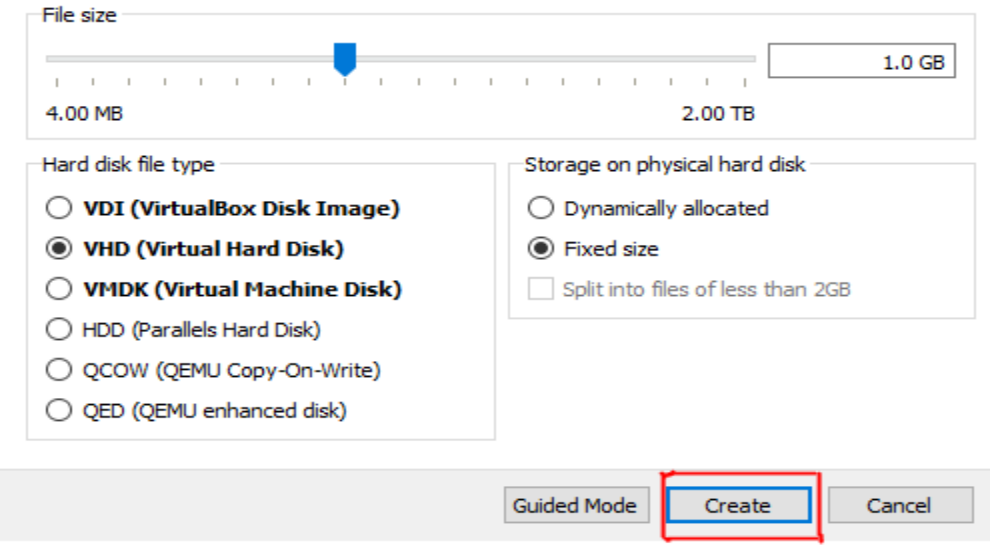
Using Kali-Linux to view hashes of a file

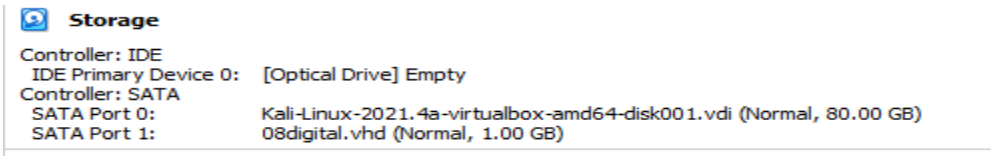
```
kali@kali: ~/Desktop/Digital-Forensics
File  Actions  Edit  View  Help

┌──(kali㉿kali)-[~/Desktop/Digital-Forensics]
└─$ printf hello | sha1sum
aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d  -

┌──(kali㉿kali)-[~/Desktop/Digital-Forensics]
└─$ openssl dgst -sha3-256 ./file1.txt
SHA3-256(./file1.txt)= e5a28afe5da4e4906c92e6e36a09fd76e76d0745f057c034b882b0a0f9ad6dfa

┌──(kali㉿kali)-[~/Desktop/Digital-Forensics]
└─$
```

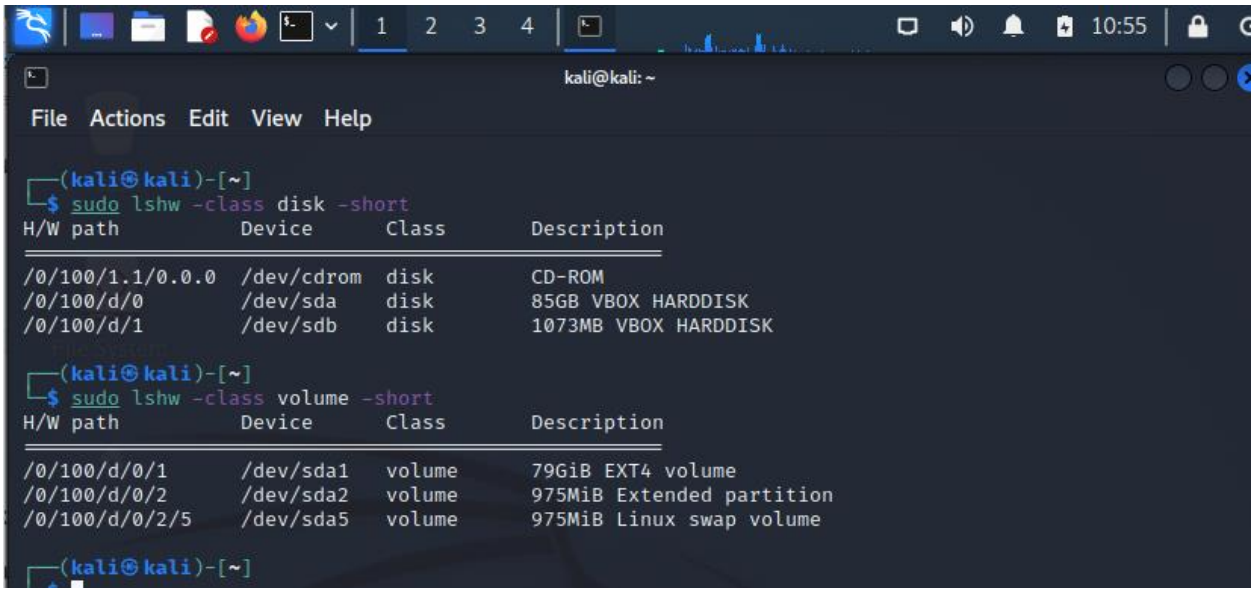
Task-2

Creating a Virtual Hard Drive and Attaching it to a VM using Oracle VirtualBox

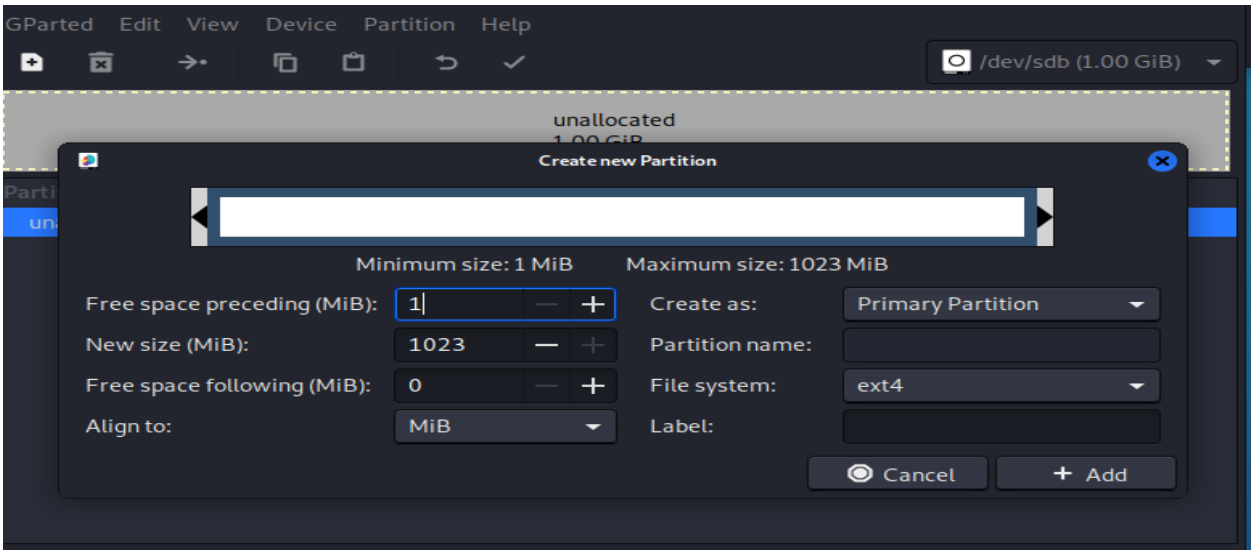
Here we are creating 1GB virtual hard disk







After creating the hard disk, we have to create new partition in our Linux machine using gparted software




```
┌──(kali㉿kali)-[~]
└─$ sudo lshw -class volume -short
H/W path              Device      Class       Description
=====================================================================
/0/100/d/0/1          /dev/sda1   volume      79GiB EXT4 volume
/0/100/d/0/2          /dev/sda2   volume      975MiB Extended partition
/0/100/d/0/2/5        /dev/sda5   volume      975MiB Linux swap volume
/0/100/d/1/1          /dev/sdb1   volume      1023MiB EXT4 volume

┌──(kali㉿kali)-[~]
└─$
```

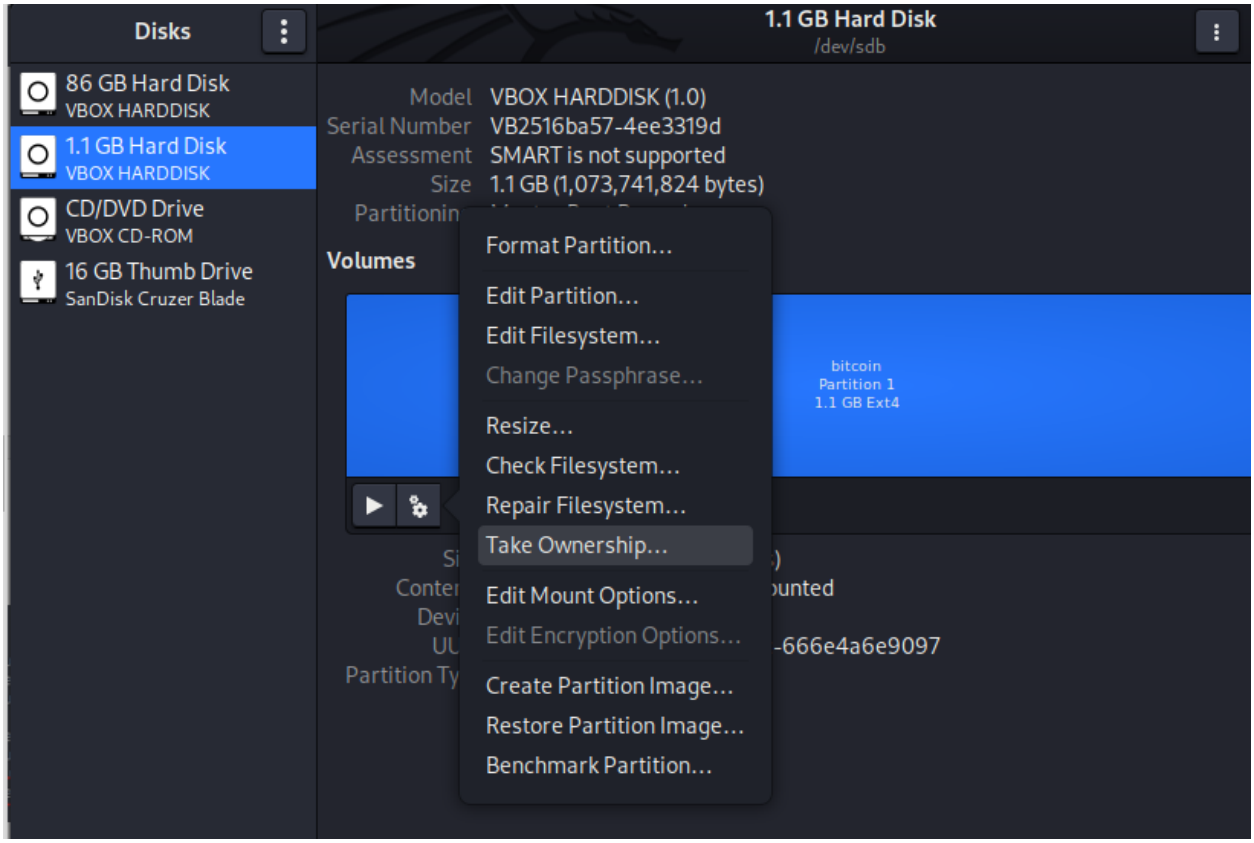
```
┌──(kali㉿kali)-[~]
└─$ sudo tune2fs -l /dev/sdb1
tune2fs 1.46.4 (18-Aug-2021)
Filesystem volume name:   <none>
Last mounted on:          <not available>
Filesystem UUID:          7d98305e-1523-4963-acfc-666e4a6e9097
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
```

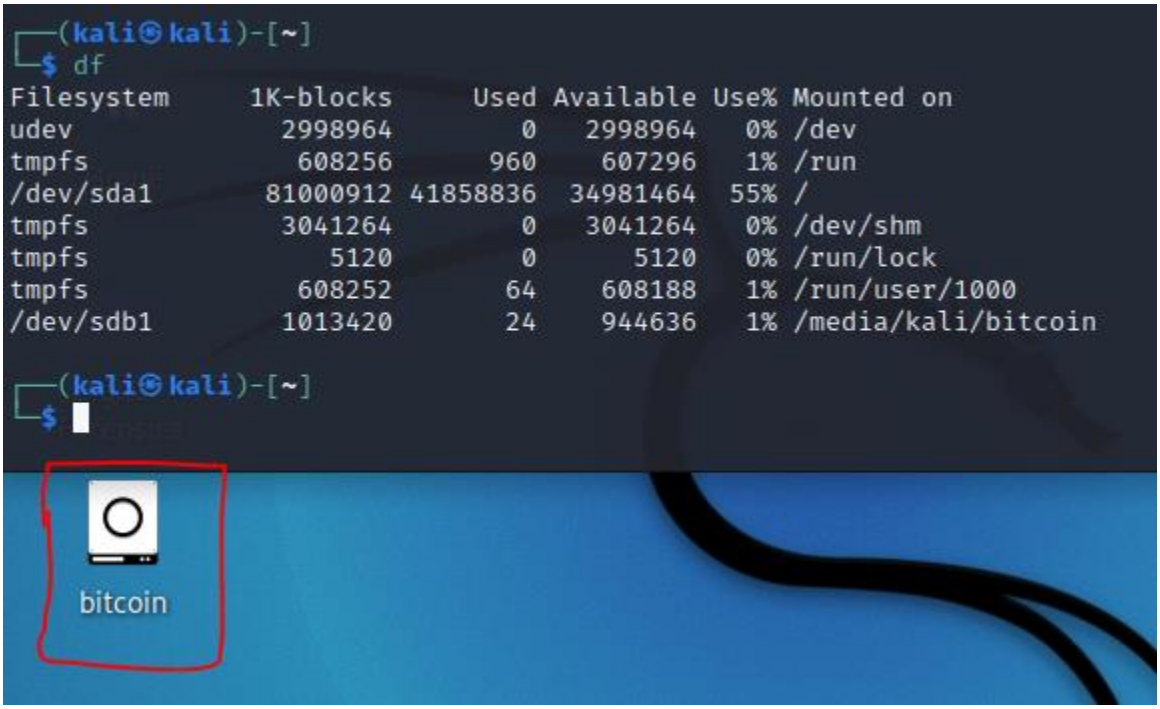
After mounting it we see that it has been given a random name so we will change it to “bitcoin” using tune2fs tool

```
┌──(kali㉿kali)-[~]
└─$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             2998960        0   2998960   0% /dev
tmpfs             608256      968    607288   1% /run
/dev/sda1       81000912 41257688  35582612  54% /
tmpfs            3041260        0   3041260   0% /dev/shm
tmpfs               5120        0      5120   0% /run/lock
tmpfs             608252       64    608188   1% /run/user/1000
/dev/sdb1        1013420       24    944636   1% /media/kali/7d98305e-1523-4963-acfc-666e4a6e9097
```

```
┌──(kali㉿kali)-[~]
└─$ sudo tune2fs -L bitcoin /dev/sdb1
tune2fs 1.46.4 (18-Aug-2021)

┌──(kali㉿kali)-[~]
└─$ reboot
```

After rebooting the name has been changed our next step is to take ownership of the hard disk





Task-4

Linux Image Carving using recoverjpeg Tool

```
┌──(kali㉿kali)-[/media/kali/bitcoin/MICC-F220]
└─$ ls -l | grep .jpg | wc -l
220

┌──(kali㉿kali)-[/media/kali/bitcoin/MICC-F220]
└─$ sudo rm *.jpg

┌──(kali㉿kali)-[/media/kali/bitcoin/MICC-F220]
└─$ ls -l | grep .jpg | wc -l
0

┌──(kali㉿kali)-[/media/kali/bitcoin/MICC-F220]
└─$ cd ../
```

After downloading the folders and unzipping them, we deleted .jpg extension images. After that we use **recoverjpeg tool** to recover the images back in to **jpegrecovery** folder.

```
┌──(kali㉿kali)-[/media/kali/bitcoin/jpegrecovery]
└─$ sudo recoverjpeg /dev/sdb1
Restored 228 pictures
```

```
┌──(kali㉿kali)-[/media/kali/bitcoin/jpegrecovery]
└─$ ls
image00000.jpg  image00038.jpg  image00076.jpg  image00114.jpg  image00152.jpg  image00190.jpg
image00001.jpg  image00039.jpg  image00077.jpg  image00115.jpg  image00153.jpg  image00191.jpg
image00002.jpg  image00040.jpg  image00078.jpg  image00116.jpg  image00154.jpg  image00192.jpg
image00003.jpg  image00041.jpg  image00079.jpg  image00117.jpg  image00155.jpg  image00193.jpg
image00004.jpg  image00042.jpg  image00080.jpg  image00118.jpg  image00156.jpg  image00194.jpg
image00005.jpg  image00043.jpg  image00081.jpg  image00119.jpg  image00157.jpg  image00195.jpg
image00006.jpg  image00044.jpg  image00082.jpg  image00120.jpg  image00158.jpg  image00196.jpg
image00007.jpg  image00045.jpg  image00083.jpg  image00121.jpg  image00159.jpg  image00197.jpg
image00008.jpg  image00046.jpg  image00084.jpg  image00122.jpg  image00160.jpg  image00198.jpg
image00009.jpg  image00047.jpg  image00085.jpg  image00123.jpg  image00161.jpg  image00199.jpg
image00010.jpg  image00048.jpg  image00086.jpg  image00124.jpg  image00162.jpg  image00200.jpg
```

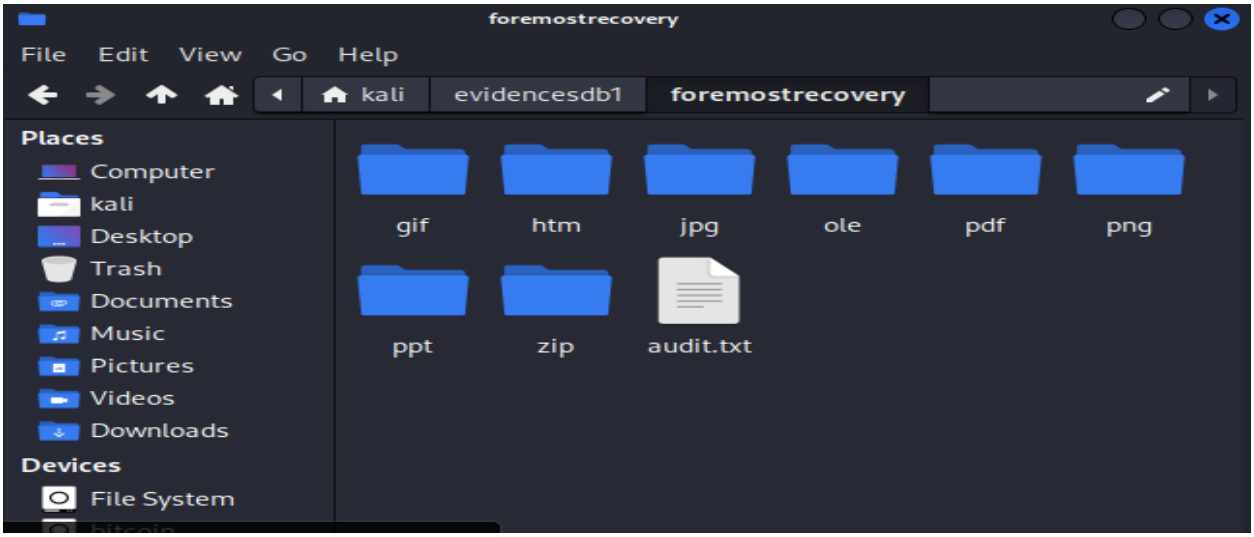
Task-5

Data Recovery using foremost Tool

Using Foremost tool to recover the images

```
┌──(kali㉿kali)-[~]
└─$ sudo dcfldd if=/dev/sdb1 of=./evidencesdb1/sdb1image.dd hash=sha1
32512 blocks (1016Mb) written.
Total (sha1): 271dfd23f8ed69700f620d06c41f88232b6045ec

32736+0 records in
32736+0 records out
```

```
┌──(kali㉿kali)-[~/evidencesdb1]
└─$ cat ./hashlog.log

Total (sha1): 271dfd23f8ed69700f620d06c41f88232b6045ec
```

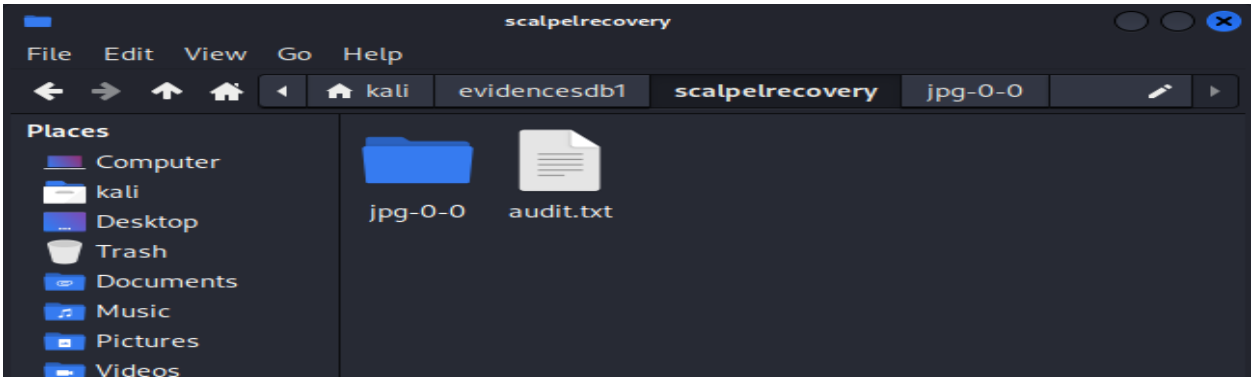


Task-6

Data Recovery using scalpel Tool

Using scalpel tool to recover deleted images

```
┌──(kali㉿kali)-[~/evidencesdb1]
└─$ sudo scalpel -o ./evidencesdb1/scalpelrecovery/ ../evidencesdb1/sdb1image.dd
Scalpel version 1.60
Written by Golden G. Richard III, based on Foremost 0.69.

Opening target "/home/kali/evidencesdb1/sdb1image.dd"

Image file pass 1/2.
../evidencesdb1/sdb1image.dd: 100.0% |*************************************| 1023.0 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\x4a\x46\x49\x46" and footer "\xff\xd9" --> 657 files
Carving files from image.
Image file pass 2/2.
../evidencesdb1/sdb1image.dd: 100.0% |*************************************| 1023.0 MB    00:00 ETA
Processing of image file complete. Cleaning up ...
Done.
Scalpel is done, files carved = 657, elapsed = 59 seconds.
```

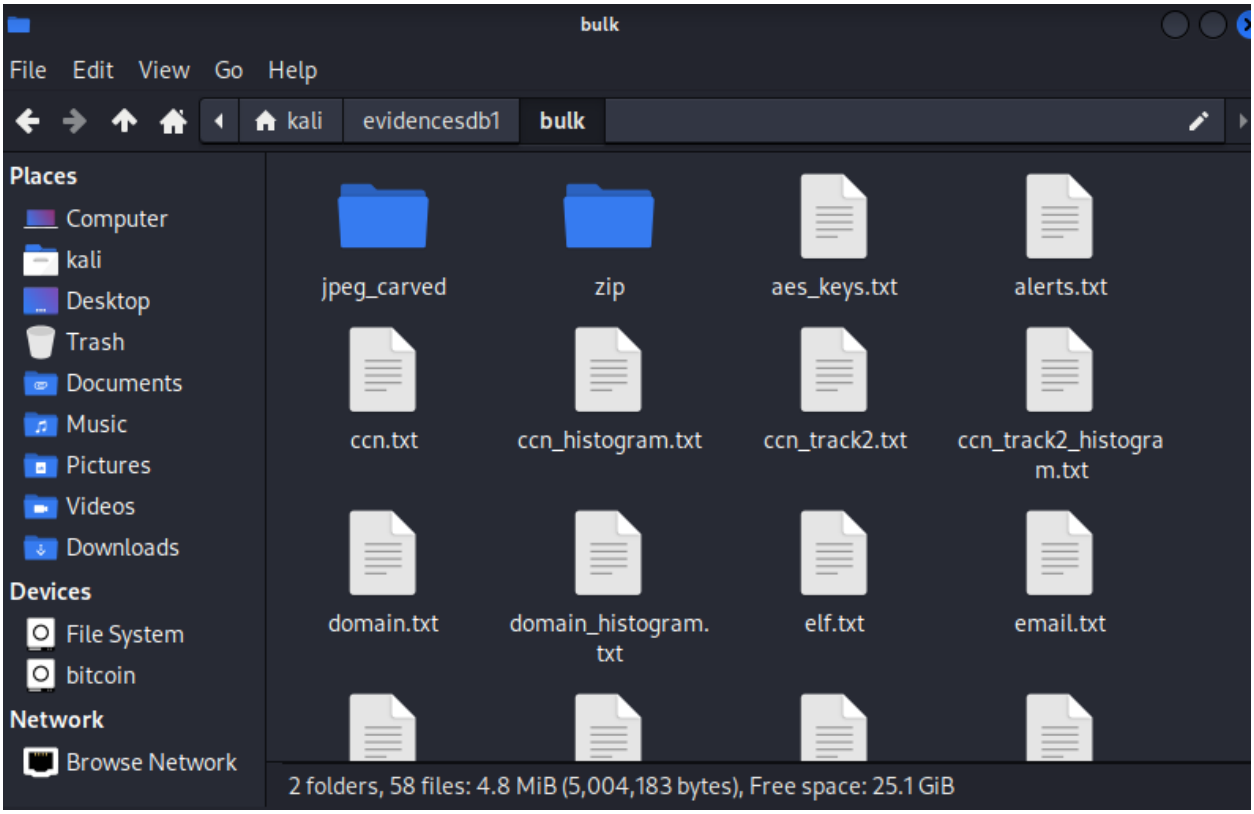


Task-7

Data Recovery and Information Retrieval using bulk_extractor

Using tool called bulk_extractor to recover deleted files

```
┌──(kali㉿kali)-[~]
└─$ sudo bulk_extractor -o ./evidencesdb1/bulk ./evidencesdb1/sdb1image.dd
mkdir "./evidencesdb1/bulk"
bulk_extractor version: 2.0.0-beta2
Input file: "./evidencesdb1/sdb1image.dd"
Output directory: "./evidencesdb1/bulk"
Disk Size: 1072693248
Scanners: aes base64 elf evtx exif facebook find gzip httplogs json kml msxml net ntfsindx ntfslogfi
le ntfsmft ntfsusn pdf rar sqlite utmp vcard windirs winlnk winpe winprefetch zip accts email gps
Threads: 3
going multi-threaded ... ( 3 )
bulk_extractor      Sun Feb 13 14:28:53 2022

available_memory: 5348384768
elapsed_time:  0:00:00
estimated_date_completion: 2022-02-13 14:28:52
estimated_time_remaining: n/a
fraction_read: 0.000000 %
max_offset: 0
sbufs_created: 0
sbufs_remaining: 0
>.....................................................................................|
```



Summary

This whole lab was to learn about creating or dealing with virtual hard-disk and deleting or recovering images using different tools. First of all, we created a virtual hard disk and gave it 1GB storage file-type ext4. After that we went to the kali linux setting and added it as virtual box storage device. Then we used **gparted** tool to allocate it as a hard-disk and then we renamed it using tool called **tune2fs.** After that we went to the image's recovery section, at first, we download two files that were given to us and then we unzipped them in our custom created hard disk. We then deleted all the images from both folders. Now we used a tool called **recoverjpeg** to recover all of the deleted images and stored it in a file called jpegrecovery. Then we use a tool called foremost to recover the deleted content from the hard disk. Scalpel is tool similar to foremost but scalpel might carve more files than foremost. We have also used it to recover the deleted content from the hard disk. Bulk_extractor is another recovery tool but digs more information such as email addresses, encryption keys, domain names, credit card numbers, among others that can also be stored on suspect media.

←END→