Toan Ngo
date:2/14/2024

Lab4

In this lab, we will be talking about how you can format your hard drive to allocate some resources for other types of data. However, before we could go deep into hard drive formatting, we must know what the drive in your computer does and what partition your hard drive means in the context of the computer.

First, what does the hard drive on your computer do? Hard drive is a hardware component that exists in the computer that stores all your digital content in the computer so that when you turn it off, the data that you save is still on the machine when you turn it back on. You can consider a hard drive like a memory bank where everything that you save will be stored in the memory until you need it then you can call it. For example, when you download some files/applications from the internet, those files/applications will be stored in the hard disk so that if you were to turn the machine off those files are still there when you turn it back on. Next, what does formatting/partition a hard drive do ? What a partition does for the hard drive is that it breaks up some space that is available in your hard drive and allows the breakup space to function like a normal storage drive that you can use for any data type. The most common used of a partition hard drive are for files organization and for files authenticating because you can hash everything that you store in the drive and comparing them to the original to see if it the same or not, and also one thing that are interesting about partitioning a drive is that you can have multiple partition of the drive. The only limitation that it has is that it runs depending on the original drive, so if your original drive has no more space then you can no longer create anymore partition drives. Think of this concept like you separate a store's space to store and organize stuff by store's label.

Now let's get to the code part of how to do some of this, however we will not be doing the disk partition just yet in this lap, but in the next lap we will learn how to do that. For now we will use a drive that is already partitioned and we will be formatting that drive.

Toan Ngo
date:2/14/2024

```
┌──(kali㉿kali)-[~]
└─$ sudo fdisk -l
[sudo] password for kali:
Disk /dev/sdb: 5 GiB, 5368709120 bytes, 10485760 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: VBOX HARDDISK
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×fe0e1bd5

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 50427903 50425856   24G 83 Linux
/dev/sda2       50429950 52426751  1996802  975M  f W95 Ext'd (LBA)
/dev/sda5       50429952 52426751  1996800  975M 82 Linux swap / Solaris
```

(display the list of drives in the system. Sda1,sda2,sda5 are the drives that currently in the system, and the number next to the “a” is representing the number of partition from the main hard disk)

```
┌──(kali㉿kali)-[~]
└─$ cd /dev

┌──(kali㉿kali)-[/dev]
└─$ ls
autofs           loop3   snapshot  tty29  tty55       vcs6
block            loop4   snd       tty3   tty56       vcs7
bsg              loop5   sr0       tty30  tty57       vcsa
btrfs-control    loop6   stderr    tty31  tty58       vcsa1
bus              loop7   stdin     tty32  tty59       vcsa2
cdrom            mapper  stdout    tty33  tty6        vcsa3
char             mem     tty       tty34  tty60       vcsa4
console          mqueue  tty0      tty35  tty61       vcsa5
core             net     tty1      tty36  tty62       vcsa6
cpu_dma_latency  null    tty10     tty37  tty63       vcsa7
cuse             nvram   tty11     tty38  tty7        vcsu
disk             port    tty12     tty39  tty8        vcsu1
dri              ppp     tty13     tty4   tty9        vcsu2
fb0              psaux   tty14     tty40  ttyS0       vcsu3
fd               ptmx    tty15     tty41  ttyS1       vcsu4
full             pts     tty16     tty42  ttyS2       vcsu5
fuse             random  tty17     tty43  ttyS3       vcsu6
hidraw0          rfkill  tty18     tty44  uhid        vcsu7
```

(list of all device/driver that recognize by the system linux)

Toan Ngo
date:2/14/2024

```
┌──(kali㉿kali)-[~]
└─$ sudo lshw -class disk -short
H/W path           Device      Class       Description
=========================================================
/0/100/1.1/0.0.0   /dev/cdrom  disk        CD-ROM
/0/100/d/0         /dev/sda    disk        26GB VBOX HARDDISK
/0/100/d/1         /dev/sdb    disk        5368MB VBOX HARDDISK

┌──(kali㉿kali)-[~]
└─$
```

(show a condensed version of the list of drivers in the category disk. Importance Note: that the function lshw is not a built in function for linux, therefore if you want to use it, then you have to install it. Inorder to install lshw run this code: sudo apt install lshw)

```
┌──(kali㉿kali)-[~]
└─$ sudo lshw -class volume -short
H/W path           Device      Class       Description
=========================================================
/0/100/d/0/1       /dev/sda1   volume      24GiB 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
```

(show the size of the driver that available for the machine.)

Simple hashing recap-you can hash a file/string this way

```
┌──(kali㉿kali)-[~]
└─$ printf thisisasample| sha1sum
438b11bc0e7318b6361f2b47c9abad61b3142936  -
```

(code: printf the string and pass it through the pipe of sha1sum hashing which generates the output of a hash value of the string.)

```
┌──(kali㉿kali)-[~]
└─$ printf cs362 | md5sum
21e807599f8ec807297d3f9d9bcbb635  -

┌──(kali㉿kali)-[~]
└─$ printf cs362 | sha512sum
be47fe03860b2c7330b2d15bb7911fbd4b5e73327b35d1a1857537948f92fbe3aaf28fb56bc59
5d5d8f0a9fdf580fb294840f33a2df3c4fd46f07cc2cfefbd97  -
```

(difference hash function that build in the linux machine that you can call)

```
┌──(kali㉿kali)-[~]
└─$ echo this is a sample file>test.txt

┌──(kali㉿kali)-[~]
└─$ md5sum test.txt
d7b848f17cc5220e2602ce609828a7df  test.txt
```

Toan Ngo
date:2/14/2024

(creating a file and then hashing it. Echo in this context is write >(to) destination file. Something to note if the file doesn't exist it will create one with the name after the >)

```
┌──(kali㉿kali)-[~]
└─$ printf cs362 | sha512sum| cut -d " " -f 1
be47fe03860b2c7330b2d15bb7911fbd4b5e73327b35d1a1857537948f92fbe3aaf28fb56bc59
5d5d8f0a9fdf580fb294840f33a2df3c4fd46f07cc2cfefbd97
```

(code:cut -d ” ”(this cut the data in the empty space) -f 1 (this is the output of the string, in this case 1 is the hash value, 2 is the - symbol)
slicing the output of the hash)

```
┌──(kali㉿kali)-[~]
└─$ printf cs362 | openssl dgst -sha3-256
SHA3-256(stdin)= e4ca8e0e958b39280f5ba86cd8864b194645c37ac1b89a778416a1bf23e4
ef0a
```

(use openssl libraries to digest the string (hashing the string) using the sha-256 algorithm
Code:print the cs362 send it through the pipe of openssl dgst(hashing method)
-sha3-256(hashing algorithm))

```
┌──(kali㉿kali)-[~]
└─$ openssl dgst -sha3-256 Downloads/*
SHA3-256(Downloads/sample.jpeg)= 5478709ffbed1935a9e0eada6415a8c9a6ae8ea287fe5e476
3a2021fccea411b
```

(use openssl to hash everything in the downloads folder.
Code:openssl dgst -sha3-256(select hashing algorithm) downloads/*(applying the previous command for all files in the downloads folder)
)

Image acquisition- create a disk image from the hard drive.

```
┌──(kali㉿kali)-[~]
└─$ sudo dc3dd if=/dev/sdb1 hash=sha1 log=usb_forensics.log of=usb_image.dd

dc3dd 7.2.646 started at 2024-02-14 04:14:21 -0600
compiled options:
command line dc3dd if=/dev/sdb1 hash=sha1 log=usb_forensics.log of=usb_image.dd
device size: 2095104 sectors (probed),    1,072,693,248 bytes
sector size: 512 bytes (probed)
  1072693248 bytes ( 1023 M ) copied ( 100% ),  210 s, 4.9 M/s

input results for device `/dev/sdb1':
   2095104 sectors in
   0 bad sectors replaced by zeros
   a54f13a0e074b3431743596ecbe3e0302c79c15f (sha1)

output results for file `usb_image.dd':
   2095104 sectors out

dc3dd completed at 2024-02-14 04:17:51 -0600
```

Toan Ngo
date:2/14/2024

(code: sudo dc3dd(applying dc3dd to input) if(the input) hash=sha1(hashing algorithm) log=usb_forensic.log(generate and output a log for the hashing of the drive) of=usb_image.dd (output the disk image).
>Sum it all up: I try to generate a disk image for the driver sdb1, and in order to do that I call dc3dd and it parameter, and after it done running it should give you a disk image.)

```
┌──(kali㉿kali)-[~]
└─$ sudo dc3dd if=/dev/sdb1 hash=sha1 log=usb_forensics.log ofsz=550M ofs=usb_fore
nsics.000

dc3dd 7.2.646 started at 2024-02-14 04:22:49 -0600
compiled options:
command line dc3dd if=/dev/sdb1 hash=sha1 log=usb_forensics.log ofsz=550M ofs=usb_
forensics.000
device size: 2095104 sectors (probed),    1,072,693,248 bytes
sector size: 512 bytes (probed)
  1072693248 bytes ( 1023 M ) copied ( 100% ),   57 s, 18 M/s

input results for device `/dev/sdb1':
   2095104 sectors in
   0 bad sectors replaced by zeros
   a54f13a0e074b3431743596ecbe3e0302c79c15f (sha1)

output results for files `usb_forensics.000':
   2095104 sectors out

dc3dd completed at 2024-02-14 04:23:46 -0600
```

(this code act is doing the same thing as above, however with two unique change , ofsz="the maximum size per files" ofs="a filename that it will iterate on top" (in my case usb_forensic.000 is my files name, every time a new files got created base on this code because the max size limit is 550M, the number after the "." will increments. Example usb_forensic.000,usb_forensic.001,usb_forensic.002 and so on)
>Sum it all up: i break the files in to small part and then hash them.)

```
┌──(kali㉿kali)-[~]
└─$ cat usb_forensics.0* |sha1sum
a54f13a0e074b3431743596ecbe3e0302c79c15f  -
```

(combine all usb_forensics.0*(started from usb_forensics.000 to usb_forensics.n) and pass it through the pipe of sha1sum algorithm and output a hash value of the combine file.)

```
# Overwrite the drive with zeros
kali@kali [~] $ dc3dd wipe=/dev/sdc


# Overwrite the drive with a pattern (in hexadecimal)
kali@kali [~] $ dc3dd wipe=/dev/sdc pat=ABCDEF


# Overwrite the drive with a text pattern
kali@kali [~] $ dc3dd wipe=/dev/sdc tpat=happyholidays
```

Toan Ngo
date:2/14/2024

(this is a way to wipe the drive. The code is basically read as overwrite you drive with all zeros value, or overwrite your drive with hex pattern with pat=ABCDEF, and overwrite your drive with a custom text pattern tpat=happyholidays.)

```
┌──(kali㉿kali)-[~]
└─$ sudo dd if=/dev/sdb1 bs=512 of=mbr.image count=1
1+0 records in
1+0 records out
512 bytes copied, 0.0557692 s, 9.2 kB/s
```

(retrieving the master boots record
Code:applying dd to input driver with the block size of 512(bs=512) and number of block to be read in 512(count=1))

```
┌──(kali㉿kali)-[~]
└─$ xxd mbr.image
00000000: fab8 0010 8ed0 bc00 b0b8 0000 8ed8 8ec0  ..............
00000010: fbbe 007c bf00 06b9 0002 f3a4 ea21 0600  ...|.........!..
00000020: 00be be07 3804 750b 83c6 1081 fefe 0775  ....8.u........u
00000030: f3eb 16b4 02b0 01bb 007c b280 8a74 018b  .........|...t..
00000040: 4c02 cd13 ea00 7c00 00eb fe00 0000 0000  L.....|.........
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
```

(this show the hex table of the master boot record of the previous image)

The following is done in window powershells

```
PS C:\Users\Tcngo> Find-Module -Name *forensic*
```

```
Version    Name                                Repository           Description
-------    ----                                ----------           -----------
1.1.1      PowerForensics                      PSGallery            A Digital Forensics framework for Windows PowerS...
1.1.1      PowerForensicsv2                    PSGallery            A Digital Forensics framework for Windows PowerS...
1.1.1      PowerForensicsPortable              PSGallery            A Digital Forensics framework for Windows PowerS...
1.0.0.0    Forensics                           PSGallery            The module can be used for performing some Evide...
```

code:Install-Module -Name PowerForensics
(this will install the module name PowerForensics on to the powershell environment)

Toan Ngo
date:2/14/2024

```
PS C:\Windows\system32> Import-Module -Name Powerforensics
PS C:\Windows\system32> Get-Command -Module PowerForensics

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          ConvertFrom-BinaryData                             1.1.1      Powerforensics
Cmdlet          ConvertTo-ForensicTimeline                         1.1.1      Powerforensics
Cmdlet          Copy-ForensicFile                                  1.1.1      Powerforensics
Cmdlet          Get-ForensicAlternateDataStream                    1.1.1      Powerforensics
Cmdlet          Get-ForensicAmcache                                1.1.1      Powerforensics
Cmdlet          Get-ForensicAttrDef                                1.1.1      Powerforensics
Cmdlet          Get-ForensicBitmap                                 1.1.1      Powerforensics
Cmdlet          Get-ForensicBootSector                             1.1.1      Powerforensics
Cmdlet          Get-ForensicChildItem                              1.1.1      Powerforensics
```

(this get the command of the module powerForensics,ie you get what command available for powerforensics.)

```
PS C:\Windows\system32> Get-ForensicFileRecord -Path C:\Users\Tcngo\Downloads\sample.txt


FullName             : C:\\Users\Tcngo\Downloads\sample.txt
Name                 : sample.txt
SequenceNumber       : 2
RecordNumber         : 119486
ParentSequenceNumber : 2
ParentRecordNumber   : 35476
Directory            : False
Deleted              : False
ModifiedTime         : 2/14/2024 11:28:44 AM
AccessedTime         : 2/14/2024 11:30:59 AM
ChangedTime          : 2/14/2024 11:28:45 AM
BornTime             : 2/14/2024 11:28:44 AM
FNModifiedTime       : 2/14/2024 11:28:44 AM
FNAccessedTime       : 2/14/2024 11:28:44 AM
FNChangedTime        : 2/14/2024 11:28:44 AM
FNBornTime           : 2/14/2024 11:28:44 AM
```

(in this step i perform a get file record on an absolute path to the file sampe.txt. This should output all the relative information that relates to the files like the day creation, modify time ect….)

Closing thought as you can see you can do alot of stuff with your hard drive, you can make more space from the hard drive to store stuff by partitioning the drive, you can wipe your hard drive if you want and so on. As we can see these are some of the things you can do to your hard drive, however there should be more things you can do if you look hard enough from external sources.