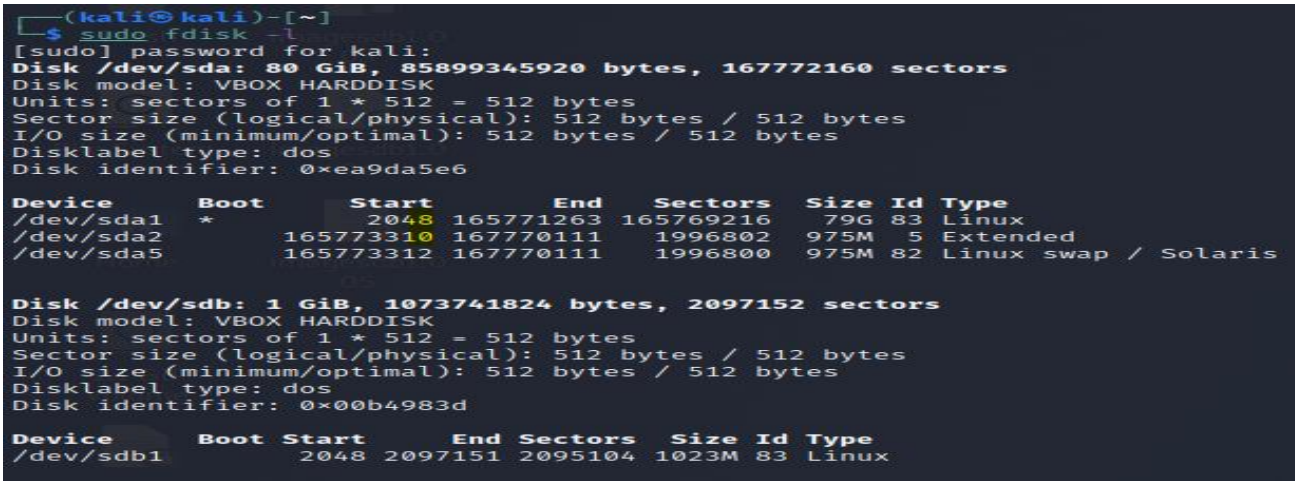
Report 4

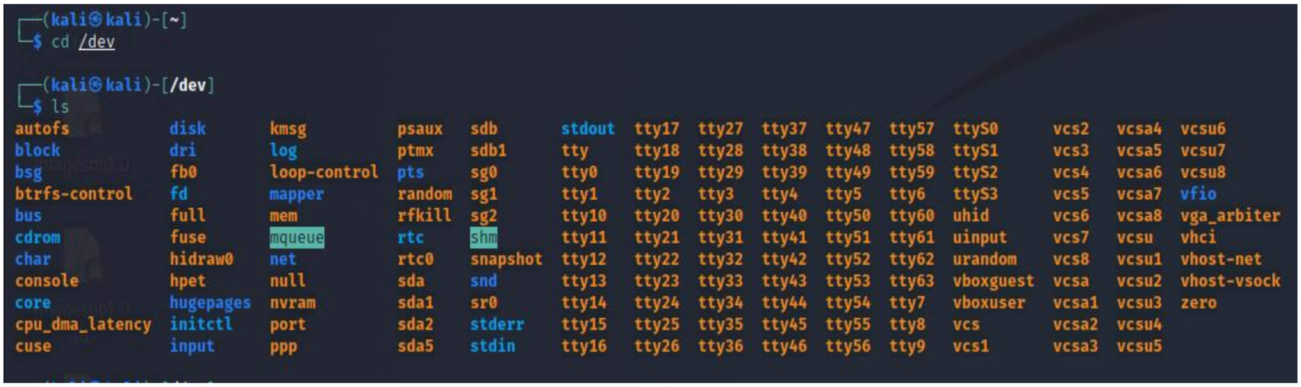
This week we worked on disk forensics. The first part of the lab involved using Kali Linux and using the terminal to analyze Drives and Partitions. The next part of the lab involved hashing files. Then the next part was on image acquisition using **dc3d**d and **dd** commands. We also utilized the **Guymager** for acquiring a drive image. Then retrieving the Master Boot Record. The last part of the lab used PowerShell on Windows to investigate file records.

The first task was to display the drives in my Linux machine using the command **sudo fdisk -l.** In Linux, drives are represented as /**dev/sda, dev/sdb,** etc. The **sd** stands for the SCSI Mass-Storage Driver. The subsequent letters **a** and **b** represent the number of drives. Here below is the information on my disk in Linux.



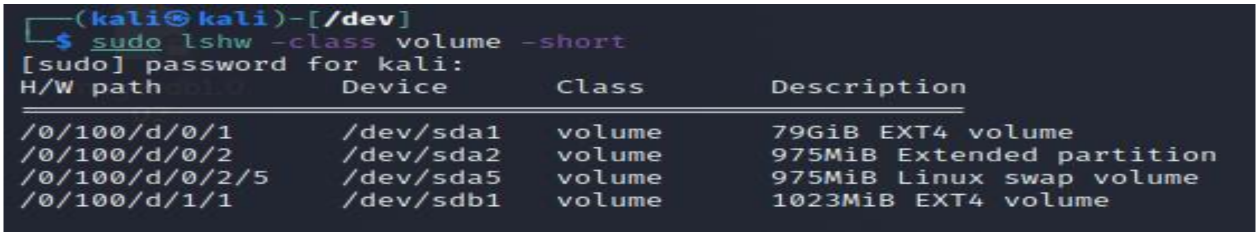
```
┌──(kali㉿kali)-[~]
└─$ sudo fdisk -l
[sudo] password for kali:
Disk /dev/sda: 80 GiB, 85899345920 bytes, 167772160 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xea9da5e6

Device     Boot     Start       End   Sectors  Size Id Type
/dev/sda1  *         2048 165771263 165769216   79G 83 Linux
/dev/sda2       165773310 167770111   1996802  975M  5 Extended
/dev/sda5       165773312 167770111   1996800  975M 82 Linux swap / Solaris

Disk /dev/sdb: 1 GiB, 1073741824 bytes, 2097152 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: 0x00b4983d

Device     Boot Start     End Sectors  Size Id Type
/dev/sdb1        2048 2097151 2095104 1023M 83 Linux
```

The **/dev** is the path of all drives and devices that is acknowledged by Linux. Using the **cd/dev** followed by the **ls** commands you can display the directory. These consists of files that represent devices that are attached to the local system. Here below you can see the **sda** drive has **3** partitions attached to These include **sda1**, **sda2**, and **sda5**.



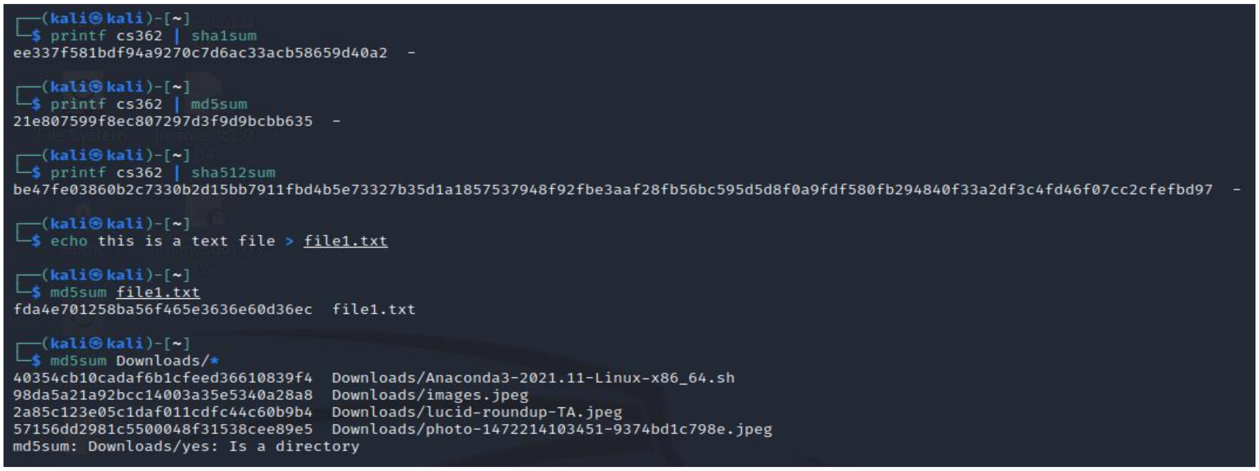
```
┌──(kali㉿kali)-[~]
└─$ cd /dev

┌──(kali㉿kali)-[/dev]
└─$ ls
autofs           disk       kmsg          psaux   sdb       stdout  tty17  tty27  tty37  tty47  tty57  ttyS0      vcs2   vcsa4  vcsu6
block            dri        log           ptmx    sdb1      tty     tty18  tty28  tty38  tty48  tty58  ttyS1      vcs3   vcsa5  vcsu7
bsg              fb0        loop-control  pts     sg0       tty0    tty19  tty29  tty39  tty49  tty59  ttyS2      vcs4   vcsa6  vcsu8
btrfs-control    fd         mapper        random  sg1       tty1    tty2   tty3   tty4   tty5   tty6   ttyS3      vcs5   vcsa7  vfio
bus              full       mem           rfkill  sg2       tty10   tty20  tty30  tty40  tty50  tty60  uhid       vcs6   vcsa8  vga_arbiter
cdrom            fuse       mqueue        rtc     shm       tty11   tty21  tty31  tty41  tty51  tty61  uinput     vcs7   vcsu   vhci
char             hidraw0    net           rtc0    snapshot  tty12   tty22  tty32  tty42  tty52  tty62  urandom    vcs8   vcsu1  vhost-net
console          hpet       null          sda     snd       tty13   tty23  tty33  tty43  tty53  tty63  vboxguest  vcsa   vcsu2  vhost-vsock
core             hugepages  nvram         sda1    sr0       tty14   tty24  tty34  tty44  tty54  tty7   vboxuser   vcsa1  vcsu3  zero
cpu_dma_latency  initctl    port          sda2    stderr    tty15   tty25  tty35  tty45  tty55  tty8   vcs        vcsa2  vcsu4
cuse             input      ppp           sda5    stdin     tty16   tty26  tty36  tty46  tty56  tty9   vcs1       vcsa3  vcsu5
```

You can also display the hardware information that is on your Linux machine. You first need to install the package using the **sudo lshw -class disk -short** command. Next, to display your hardware information you can use the **sudo lshw -class volume -short** command to display it. Here below is the hardware information of my system.



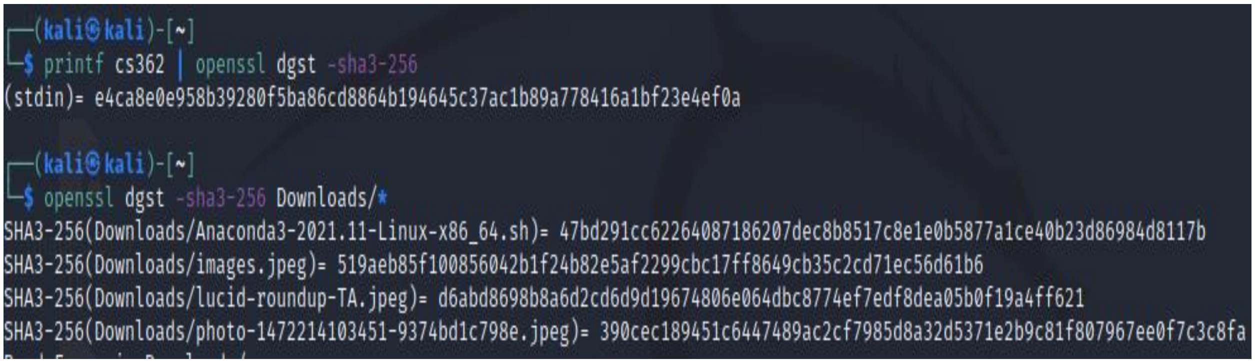
```
┌──(kali㉿kali)-[/dev]
└─$ sudo lshw -class volume -short
[sudo] password for kali:
H/W path            Device       Class        Description
==========================================================
/0/100/d/0/1        /dev/sda1    volume       79GiB EXT4 volume
/0/100/d/0/2        /dev/sda2    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
```

The next task was to use hash files using the several built-in commands that Linux has available. You can use **printf** to hash a sting like **cs362** then utilize the many hashing commands such as **sha1sum** and **md5sum** to print the hash value of that string. You can also create a text file using **echo** then write your desired text followed by saving the file using the > to overwrite the existing file or creates a file if the file of the mentioned name is not present in the directory**.** You can also hash files on Linux. The last command **md5sum Downloads/*** shows the hashed values of all the files that are in my /Downloads directory.



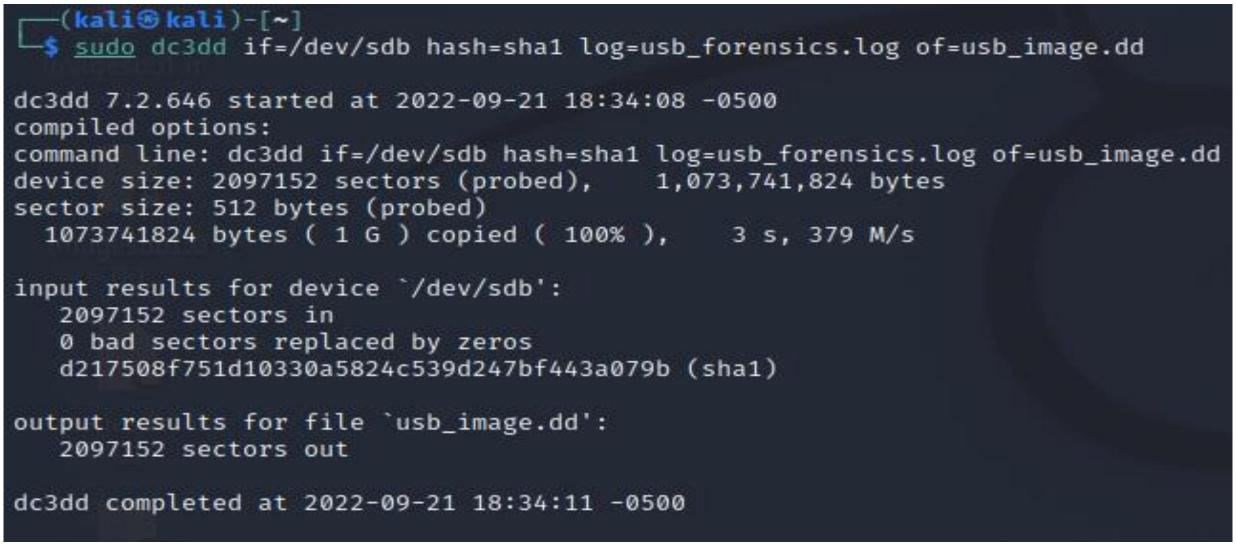
```
┌──(kali㉿kali)-[~]
└─$ printf cs362 | sha1sum
ee337f581bdf94a9270c7d6ac33acb58659d40a2  -

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

┌──(kali㉿kali)-[~]
└─$ printf cs362 | sha512sum
be47fe03860b2c7330b2d15bb7911fbd4b5e73327b35d1a1857537948f92fbe3aaf28fb56bc595d5d8f0a9fdf580fb294840f33a2df3c4fd46f07cc2cfefbd97  -

┌──(kali㉿kali)-[~]
└─$ echo this is a text file > file1.txt

┌──(kali㉿kali)-[~]
└─$ md5sum file1.txt
fda4e701258ba56f465e3636e60d36ec  file1.txt

┌──(kali㉿kali)-[~]
└─$ md5sum Downloads/*
40354cb10cadaf6b1cfeed36610839f4  Downloads/Anaconda3-2021.11-Linux-x86_64.sh
98da5a21a92bcc14003a35e5340a28a8  Downloads/images.jpeg
2a85c123e05c1daf011cdfc44c60b9b4  Downloads/lucid-roundup-TA.jpeg
57156dd2981c5500048f31538cee89e5  Downloads/photo-1472214103451-9374bd1c798e.jpeg
md5sum: Downloads/yes: Is a directory
```

SHA-3 is another tool you can use to hash files and strings in Linux. You also need to use the **openssl** software library. Here below is an example of the hashing a string and files using **printf cs-362 | openssl dgst -sha3-256** and **openssl dgst -sha3-256 Downloads/***command



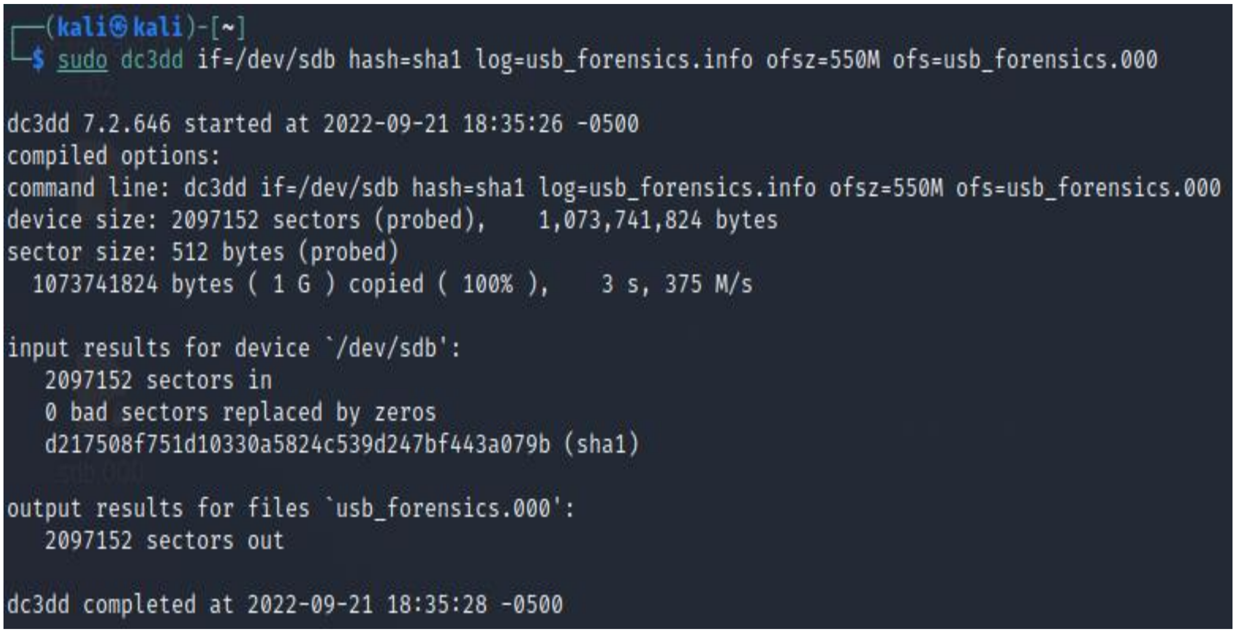
```
┌──(kali㉿kali)-[~]
└─$ printf cs362 | openssl dgst -sha3-256
(stdin)= e4ca8e0e958b39280f5ba86cd8864b194645c37ac1b89a778416a1bf23e4ef0a

┌──(kali㉿kali)-[~]
└─$ openssl dgst -sha3-256 Downloads/*
SHA3-256(Downloads/Anaconda3-2021.11-Linux-x86_64.sh)= 47bd291cc62264087186207dec8b8517c8e1e0b5877a1ce40b23d86984d8117b
SHA3-256(Downloads/images.jpeg)= 519aeb85f100856042b1f24b82e5af2299cbc17ff8649cb35c2cd71ec56d61b6
SHA3-256(Downloads/lucid-roundup-TA.jpeg)= d6abd8698b8a6d2cd6d9d19674806e064dbc8774ef7edf8dea05b0f19a4ff621
SHA3-256(Downloads/photo-1472214103451-9374bd1c798e.jpeg)= 390cec189451c6447489ac2cf7985d8a32d5371e2b9c81f807967ee0f7c3c8fa
```

The next part of the lab involved using the dc3dd and dd commands for image acquisition. You first need to install the package by using the command **sudo apt-get install dc3dd.** To create a raw image file, you need to use the command **sudo dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.log of=usb_image.dd**. Here below is the raw image of **sdb**.



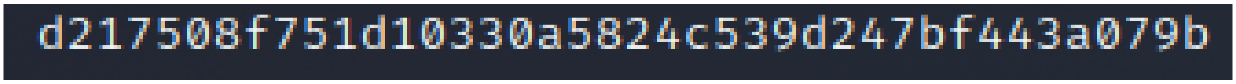
```
┌──(kali㉿kali)-[~]
└─$ sudo dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.log of=usb_image.dd

dc3dd 7.2.646 started at 2022-09-21 18:34:08 -0500
compiled options:
command line: dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.log of=usb_image.dd
device size: 2097152 sectors (probed),    1,073,741,824 bytes
sector size: 512 bytes (probed)
  1073741824 bytes ( 1 G ) copied ( 100% ),    3 s, 379 M/s

input results for device `/dev/sdb':
   2097152 sectors in
   0 bad sectors replaced by zeros
   d217508f751d10330a5824c539d247bf443a079b (sha1)

output results for file `usb_image.dd':
   2097152 sectors out

dc3dd completed at 2022-09-21 18:34:11 -0500
```

Since the drive size is large, we split the image into several files. To that we can type **sudo dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.info ofsz=550M ofs=usb_forensics.000.** Here below is splitting the image files of sdb.



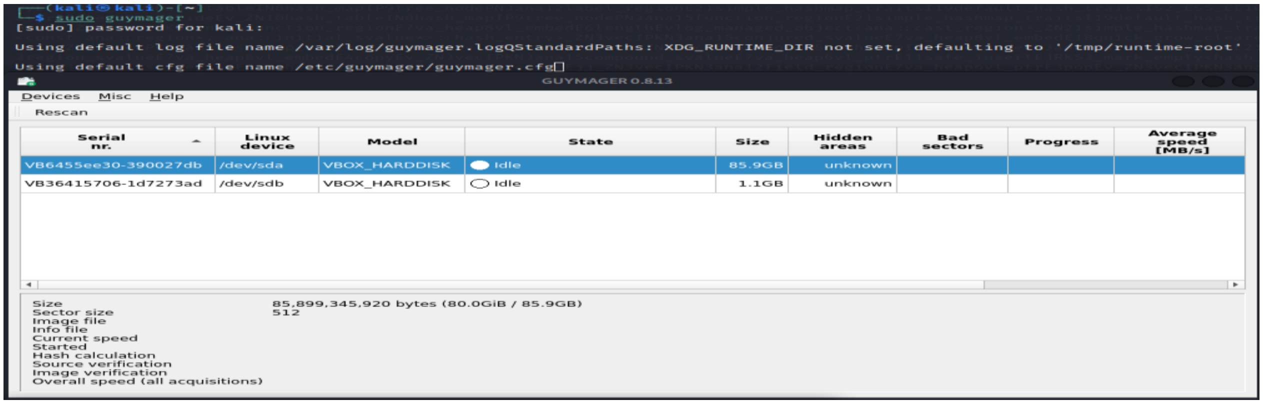
```
┌──(kali㉿kali)-[~]
└─$ sudo dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.info ofsz=550M ofs=usb_forensics.000

dc3dd 7.2.646 started at 2022-09-21 18:35:26 -0500
compiled options:
command line: dc3dd if=/dev/sdb hash=sha1 log=usb_forensics.info ofsz=550M ofs=usb_forensics.000
device size: 2097152 sectors (probed),    1,073,741,824 bytes
sector size: 512 bytes (probed)
  1073741824 bytes ( 1 G ) copied ( 100% ),    3 s, 375 M/s

input results for device `/dev/sdb':
   2097152 sectors in
   0 bad sectors replaced by zeros
   d217508f751d10330a5824c539d247bf443a079b (sha1)

output results for files `usb_forensics.000':
   2097152 sectors out

dc3dd completed at 2022-09-21 18:35:28 -0500
```

You can also compute the hash value to verify the of the files using **cat usb_forensics.0* sha1sum.**



```
d217508f751d10330a5824c539d247bf443a079b
```

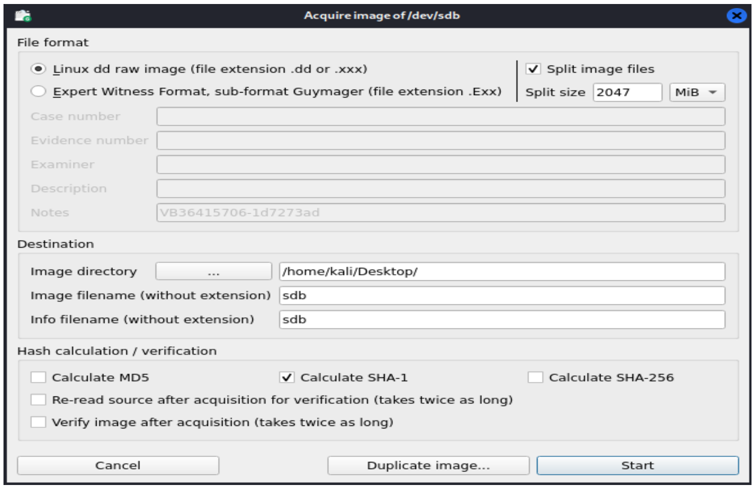
The next part of the lab also includes acquiring images using Guymager. You use the command **sudo guymager** and a new window opens,



```
┌──(kali㉿kali)-[~]
└─$ sudo guymager
[sudo] password for kali:
Using default log file name /var/log/guymager.logQStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
Using default cfg file name /etc/guymager/guymager.cfg
```

Serial nr.	Linux device	Model	State	Size	Hidden areas	Bad sectors	Progress	Average speed [MB/s]
VB6455ee30-390027db	/dev/sda	VBOX_HARDDISK	● Idle	85.9GB	unknown			
VB36415706-1d7273ad	/dev/sdb	VBOX_HARDDISK	○ Idle	1.1GB	unknown			

```
Size                               85,899,345,920 bytes (80.0GiB / 85.9GB)
Sector size                        512
Image file
Info file
Current speed
Started
Hash calculation
Source verification
Image verification
Overall speed (all acquisitions)
```



Acquire image of /dev/sdb

File format

● Linux dd raw image (file extension .dd or .xxx) ☑ Split image files

○ Expert Witness Format, sub-format Guymager (file extension .Exx) Split size 2047 MiB

Case number

Evidence number

Examiner

Description

Notes VB36415706-1d7273ad

Destination

Image directory ... /home/kali/Desktop/

Image filename (without extension) sdb

Info filename (without extension) sdb

Hash calculation / verification

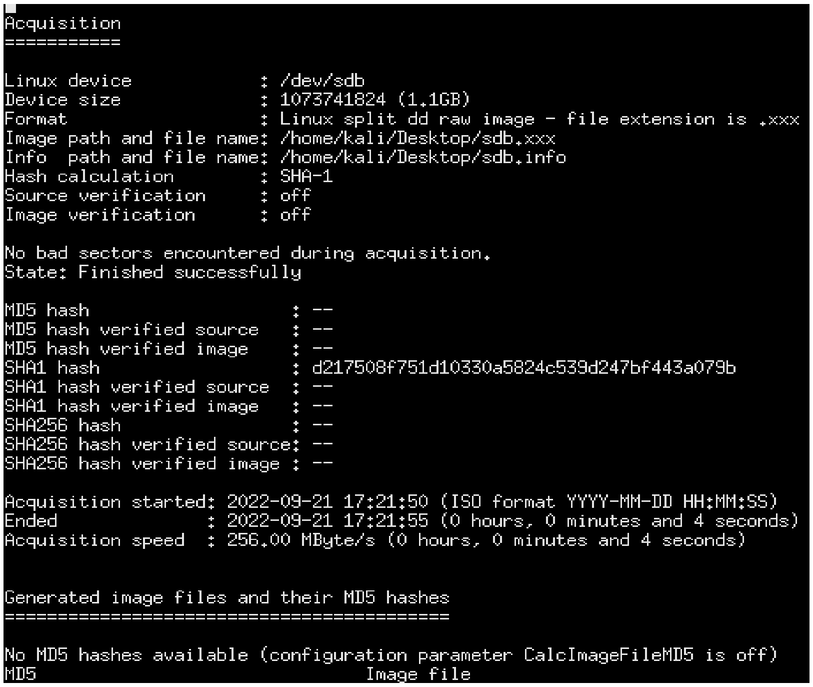
□ Calculate MD5 ☑ Calculate SHA-1 □ Calculate SHA-256

□ Re-read source after acquisition for verification (takes twice as long)

□ Verify image after acquisition (takes twice as long)

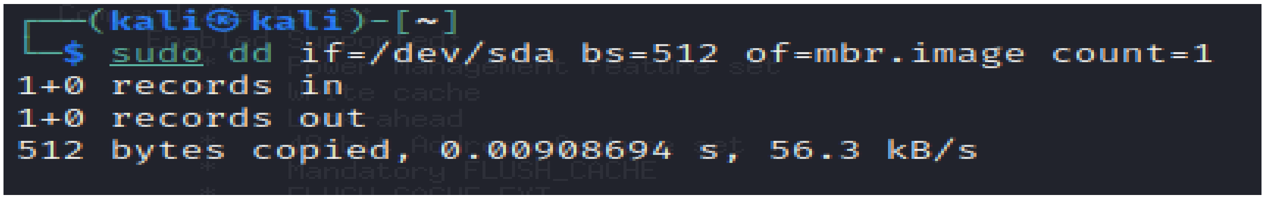
Here I can create an image file and hashes to that as well. You can select the file extension to save the image Here is the save the image naming it sdb.

Here is the acquisition of that image. Note that it has the same SHA1 hash value when we hashed it using **catusb_forensics.0* sha1sum** in the terminal.



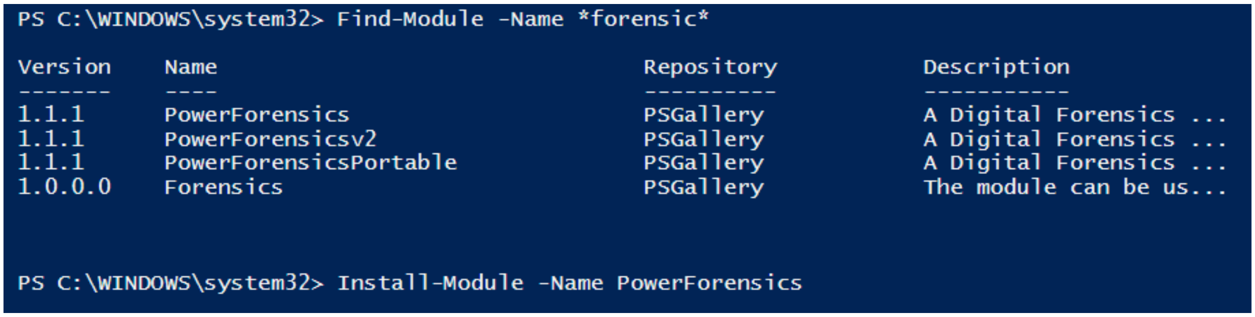
```
Acquisition
===========

Linux device            : /dev/sdb
Device size             : 1073741824 (1.1GB)
Format                  : Linux split dd raw image - file extension is .xxx
Image path and file name: /home/kali/Desktop/sdb.xxx
Info  path and file name: /home/kali/Desktop/sdb.info
Hash calculation        : SHA-1
Source verification     : off
Image verification      : off

No bad sectors encountered during acquisition.
State: Finished successfully

MD5 hash                   : --
MD5 hash verified source   : --
MD5 hash verified image    : --
SHA1 hash                  : d217508f751d10330a5824c539d247bf443a079b
SHA1 hash verified source  : --
SHA1 hash verified image   : --
SHA256 hash                : --
SHA256 hash verified source: --
SHA256 hash verified image : --

Acquisition started: 2022-09-21 17:21:50 (ISO format YYYY-MM-DD HH:MM:SS)
Ended              : 2022-09-21 17:21:55 (0 hours, 0 minutes and 4 seconds)
Acquisition speed  : 256.00 MByte/s (0 hours, 0 minutes and 4 seconds)


Generated image files and their MD5 hashes
==========================================

No MD5 hashes available (configuration parameter CalcImageFileMD5 is off)
MD5                                  Image file
```

The next part involved retrieving the Master Boot Record using the dd command. Using the sudo **dd if=/dev/sda bs=512 of=mbr.image count=1** command you can retrieve the first cluster of a drive. In this case, I will retrieve the **/dev/sda** drive**.**



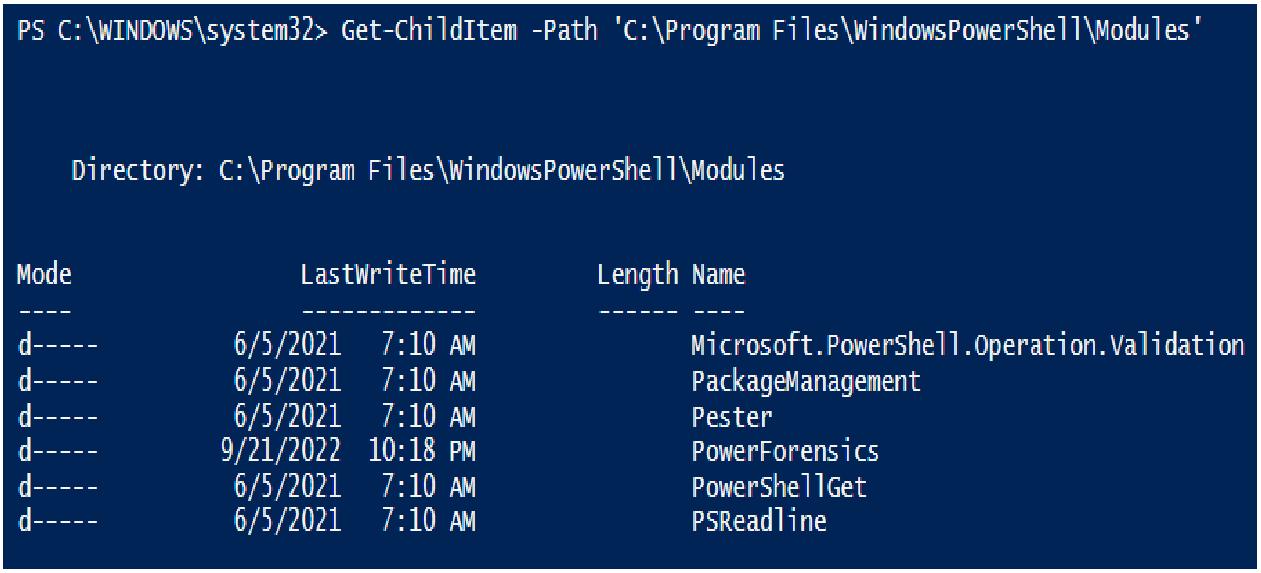
```
┌──(kali㉿kali)-[~]
└─$ sudo dd if=/dev/sda bs=512 of=mbr.image count=1
1+0 records in
1+0 records out
512 bytes copied, 0.00908694 s, 56.3 kB/s
```

The last part of the lab involved using PowerShell to investigate information on a Windows Disk. The first task is to find the module needed to do the lab. To do this, you can type **Find-Module -Name *forensic*** and then choose the module you want to install. In this case, we will add the PowerForensics Module by typing **Install-Module -Name PowerForensics** command in PowerShell.



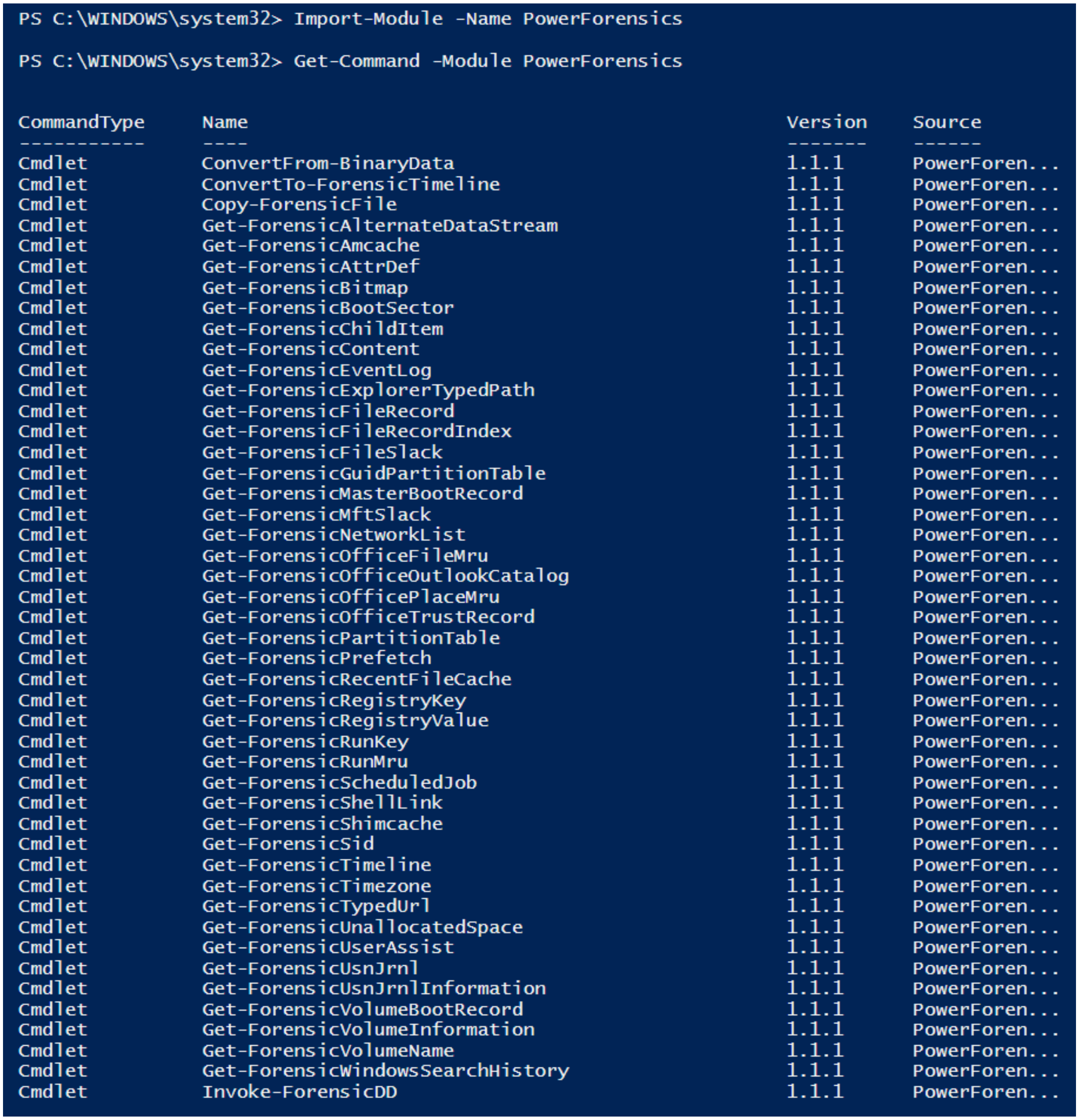
```
PS C:\WINDOWS\system32> Find-Module -Name *forensic*

Version    Name                                Repository           Description
-------    ----                                ----------           -----------
1.1.1      PowerForensics                      PSGallery            A Digital Forensics ...
1.1.1      PowerForensicsv2                    PSGallery            A Digital Forensics ...
1.1.1      PowerForensicsPortable              PSGallery            A Digital Forensics ...
1.0.0.0    Forensics                           PSGallery            The module can be us...


PS C:\WINDOWS\system32> Install-Module -Name PowerForensics
```

We can use the **Get-ChildItem -Path 'C:\ProgramFiles\WindowsPowerShell\Modules'** command to view Widowns PowerShell Modules that are in the Directory of C: as you see below.



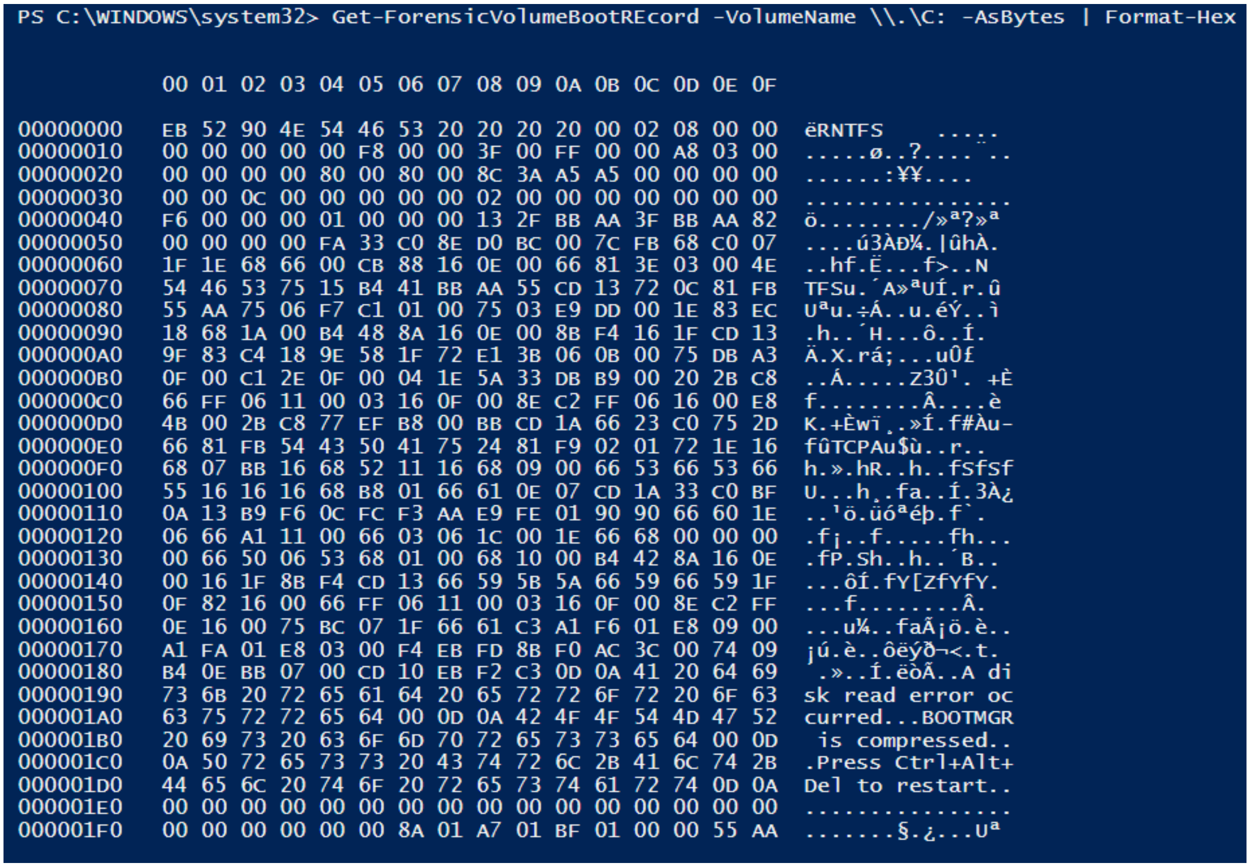
```
PS C:\WINDOWS\system32> Get-ChildItem -Path 'C:\Program Files\WindowsPowerShell\Modules'


    Directory: C:\Program Files\WindowsPowerShell\Modules


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          6/5/2021   7:10 AM                Microsoft.PowerShell.Operation.Validation
d-----          6/5/2021   7:10 AM                PackageManagement
d-----          6/5/2021   7:10 AM                Pester
d-----         9/21/2022  10:18 PM                PowerForensics
d-----          6/5/2021   7:10 AM                PowerShellGet
d-----          6/5/2021   7:10 AM                PSReadline
```

Next, we need to import the module and view the contained **Cmdlets** so that we can use them to investigate information on Windows. First, we run the **Import-Module -Name PowerForensics** command followed by the **Get-Command -Module PowerForensics** command. Here below you can see the list of Cmdlets that is included with the PowerForensics Module.



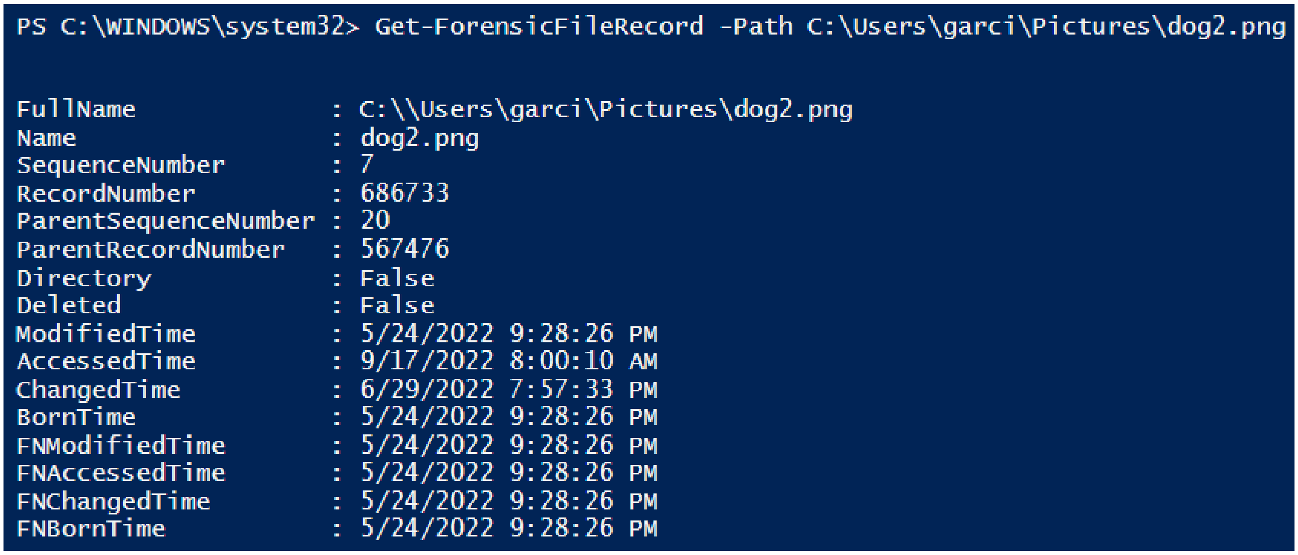
```
PS C:\WINDOWS\system32> Import-Module -Name PowerForensics

PS C:\WINDOWS\system32> Get-Command -Module PowerForensics


CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          ConvertFrom-BinaryData                             1.1.1      PowerForen...
Cmdlet          ConvertTo-ForensicTimeline                         1.1.1      PowerForen...
Cmdlet          Copy-ForensicFile                                  1.1.1      PowerForen...
Cmdlet          Get-ForensicAlternateDataStream                    1.1.1      PowerForen...
Cmdlet          Get-ForensicAmcache                                1.1.1      PowerForen...
Cmdlet          Get-ForensicAttrDef                                1.1.1      PowerForen...
Cmdlet          Get-ForensicBitmap                                 1.1.1      PowerForen...
Cmdlet          Get-ForensicBootSector                             1.1.1      PowerForen...
Cmdlet          Get-ForensicChildItem                              1.1.1      PowerForen...
Cmdlet          Get-ForensicContent                                1.1.1      PowerForen...
Cmdlet          Get-ForensicEventLog                               1.1.1      PowerForen...
Cmdlet          Get-ForensicExplorerTypedPath                      1.1.1      PowerForen...
Cmdlet          Get-ForensicFileRecord                             1.1.1      PowerForen...
Cmdlet          Get-ForensicFileRecordIndex                        1.1.1      PowerForen...
Cmdlet          Get-ForensicFileSlack                              1.1.1      PowerForen...
Cmdlet          Get-ForensicGuidPartitionTable                     1.1.1      PowerForen...
Cmdlet          Get-ForensicMasterBootRecord                       1.1.1      PowerForen...
Cmdlet          Get-ForensicMftSlack                               1.1.1      PowerForen...
Cmdlet          Get-ForensicNetworkList                            1.1.1      PowerForen...
Cmdlet          Get-ForensicOfficeFileMru                          1.1.1      PowerForen...
Cmdlet          Get-ForensicOfficeOutlookCatalog                   1.1.1      PowerForen...
Cmdlet          Get-ForensicOfficePlaceMru                         1.1.1      PowerForen...
Cmdlet          Get-ForensicOfficeTrustRecord                      1.1.1      PowerForen...
Cmdlet          Get-ForensicPartitionTable                         1.1.1      PowerForen...
Cmdlet          Get-ForensicPrefetch                               1.1.1      PowerForen...
Cmdlet          Get-ForensicRecentFileCache                        1.1.1      PowerForen...
Cmdlet          Get-ForensicRegistryKey                            1.1.1      PowerForen...
Cmdlet          Get-ForensicRegistryValue                          1.1.1      PowerForen...
Cmdlet          Get-ForensicRunKey                                 1.1.1      PowerForen...
Cmdlet          Get-ForensicRunMru                                 1.1.1      PowerForen...
Cmdlet          Get-ForensicScheduledJob                           1.1.1      PowerForen...
Cmdlet          Get-ForensicShellLink                              1.1.1      PowerForen...
Cmdlet          Get-ForensicShimcache                              1.1.1      PowerForen...
Cmdlet          Get-ForensicSid                                    1.1.1      PowerForen...
Cmdlet          Get-ForensicTimeline                               1.1.1      PowerForen...
Cmdlet          Get-ForensicTimezone                               1.1.1      PowerForen...
Cmdlet          Get-ForensicTypedUrl                               1.1.1      PowerForen...
Cmdlet          Get-ForensicUnallocatedSpace                       1.1.1      PowerForen...
Cmdlet          Get-ForensicUserAssist                             1.1.1      PowerForen...
Cmdlet          Get-ForensicUsnJrnl                                1.1.1      PowerForen...
Cmdlet          Get-ForensicUsnJrnlInformation                     1.1.1      PowerForen...
Cmdlet          Get-ForensicVolumeBootRecord                       1.1.1      PowerForen...
Cmdlet          Get-ForensicVolumeInformation                      1.1.1      PowerForen...
Cmdlet          Get-ForensicVolumeName                             1.1.1      PowerForen...
Cmdlet          Get-ForensicWindowsSearchHistory                   1.1.1      PowerForen...
Cmdlet          Invoke-ForensicDD                                  1.1.1      PowerForen...
```

We can use the **Get-ForensicVolumeBootREcord -VolumeName \\.\C: -AsBytes | Format-Hex** command to get the master boot record volume of the C drive. You can verify this is an MBR partition since the last two values are ‘**55 AA**’.



```
PS C:\WINDOWS\system32> Get-ForensicVolumeBootREcord -VolumeName \\.\C: -AsBytes | Format-Hex


           00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000   EB 52 90 4E 54 46 53 20 20 20 20 00 02 08 00 00  ëRNTFS    .....
00000010   00 00 00 00 00 F8 00 00 3F 00 FF 00 00 A8 03 00  .....ø..?....¨..
00000020   00 00 00 00 80 00 80 00 8C 3A A5 A5 00 00 00 00  ......:¥¥....
00000030   00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
00000040   F6 00 00 00 01 00 00 00 13 2F BB AA 3F BB AA 82  ö......../»ª?»ª
00000050   00 00 00 00 FA 33 C0 8E D0 BC 00 7C FB 68 C0 07  ....ú3ÀÐ¼.|ûhÀ.
00000060   1F 1E 68 66 00 CB 88 16 0E 00 66 81 3E 03 00 4E  ..hf.Ë...f>..N
00000070   54 46 53 75 15 B4 41 BB AA 55 CD 13 72 0C 81 FB  TFSu.´A»ªUÍ.r.û
00000080   55 AA 75 06 F7 C1 01 00 75 03 E9 DD 00 1E 83 EC  Uªu.÷Á..u.éÝ..ì
00000090   18 68 1A 00 B4 48 8A 16 0E 00 8B F4 16 1F CD 13  .h..´H...ô..Í.
000000A0   9F 83 C4 18 9E 58 1F 72 E1 3B 06 0B 00 75 DB A3  Ä.X.rá;...uÛ£
000000B0   0F 00 C1 2E 0F 00 04 1E 5A 33 DB B9 00 20 2B C8  ..Á.....Z3Û¹. +È
000000C0   66 FF 06 11 00 03 16 0F 00 8E C2 FF 06 16 00 E8  f........Â....è
000000D0   4B 00 2B C8 77 EF B8 00 BB CD 1A 66 23 C0 75 2D  K.+Èwï¸.»Í.f#Àu-
000000E0   66 81 FB 54 43 50 41 75 24 81 F9 02 01 72 1E 16  fûTCPAu$ù..r..
000000F0   68 07 BB 16 68 52 11 16 68 09 00 66 53 66 53 66  h.».hR..h..fSfSf
00000100   55 16 16 16 68 B8 01 66 61 0E 07 CD 1A 33 C0 BF  U...h¸.fa..Í.3À¿
00000110   0A 13 B9 F6 0C FC F3 AA E9 FE 01 90 90 66 60 1E  ..¹ö.üóªéþ.f`.
00000120   06 66 A1 11 00 66 03 06 1C 00 1E 66 68 00 00 00  .f¡..f.....fh...
00000130   00 66 50 06 53 68 01 00 68 10 00 B4 42 8A 16 0E  .fP.Sh..h..´B..
00000140   00 16 1F 8B F4 CD 13 66 59 5B 5A 66 59 66 59 1F  ...ôÍ.fY[ZfYfY.
00000150   0F 82 16 00 66 FF 06 11 00 03 16 0F 00 8E C2 FF  ...f........Â.
00000160   0E 16 00 75 BC 07 1F 66 61 C3 A1 F6 01 E8 09 00  ...u¼..faÃ¡ö.è..
00000170   A1 FA 01 E8 03 00 F4 EB FD 8B F0 AC 3C 00 74 09  ¡ú.è..ôëýð¬<.t.
00000180   B4 0E BB 07 00 CD 10 EB F2 C3 0D 0A 41 20 64 69   .»..Í.ëòÃ..A di
00000190   73 6B 20 72 65 61 64 20 65 72 72 6F 72 20 6F 63  sk read error oc
000001A0   63 75 72 72 65 64 00 0D 0A 42 4F 4F 54 4D 47 52  curred...BOOTMGR
000001B0   20 69 73 20 63 6F 6D 70 72 65 73 73 65 64 00 0D   is compressed..
000001C0   0A 50 72 65 73 73 20 43 74 72 6C 2B 41 6C 74 2B  .Press Ctrl+Alt+
000001D0   44 65 6C 20 74 6F 20 72 65 73 74 61 72 74 0D 0A  Del to restart..
000001E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001F0   00 00 00 00 00 00 8A 01 A7 01 BF 01 00 00 55 AA  .......§.¿...Uª
```

You can use the **Get-ForensicFileRecord** command to get file records from the master file table. In the example below I used the dog2.png picture I used in last week’s lab. To do that, I need to set the path to **-Path C:\Users\garci\Pictures\dog2.png** to get information on that file.



```
PS C:\WINDOWS\system32> Get-ForensicFileRecord -Path C:\Users\garci\Pictures\dog2.png


FullName              : C:\\Users\garci\Pictures\dog2.png
Name                  : dog2.png
SequenceNumber        : 7
RecordNumber          : 686733
ParentSequenceNumber  : 20
ParentRecordNumber    : 567476
Directory             : False
Deleted               : False
ModifiedTime          : 5/24/2022 9:28:26 PM
AccessedTime          : 9/17/2022 8:00:10 AM
ChangedTime           : 6/29/2022 7:57:33 PM
BornTime              : 5/24/2022 9:28:26 PM
FNModifiedTime        : 5/24/2022 9:28:26 PM
FNAccessedTime        : 5/24/2022 9:28:26 PM
FNChangedTime         : 5/24/2022 9:28:26 PM
FNBornTime            : 5/24/2022 9:28:26 PM
```