Lab Requirements

1. Microsoft Windows virtual machine
2. Kali Linux virtual machine
3. Python 3.x

Part I: Drives and Partitions in Linux

STEP 1: Power on a Kali VM and open a terminal.

STEP 2: Type the following command and analyze the displayed result.

```
kali@kali [~] $ sudo fdisk -l
   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: 0x95ba73e4

   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: 10 GiB, 10737418240 bytes, 20971520 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: 0xd6dfcbd2

   Device       Boot     Start         End   Sectors Size Id Type
   /dev/sdb1              2048    8390655    8388608     4G   c W95 FAT32 (LBA)
   /dev/sdb2           8390656  20971519  12580864     6G 83  Linux

   Disk /dev/sdc: 14.42 GiB, 15483273216 bytes, 30240768 sectors
```

```
Disk model: DataTraveler 2.0
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: 0x46664c21

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdc1   *     2048 30240767 30238720 14.4G  c W95 FAT32 (LBA)
```

NOTE 2-1: In Windows, drives are recognized as **Disk 0**, **Disk 1**, and so forth. Windows drives and partitions is the subject of Part III of this lab.

NOTE 2-2: In Linux, drives are recognized as **/dev/sda**, **/dev/sdb**, **/dev/sdc**, and so forth.

NOTE 2-3: **sd** stands for SCSI Mass-Storage Driver. The following letter represent the drive number (**a**, **b**, **c**, …).

NOTE 2-4: **/dev** is the path of all devices and drives recognized by Linux. To obtain a list of all the recognized devices and drives, type the following two commands.

```
kali@kali [~] $ cd /dev
kali@kali [/dev] $ ls
...
mqueue  net     null    nvram   port    ppp     psaux   ptmx
pts     random  rfkill  rtc     rtc0    sda     sda1    sda2
sda5    sdb     sdb1    sdb1    sdc     sdc1    snapshot snd
sr0     stderr  stdin   stdout
...
```

NOTE 2-5: **sda** is the first drive, **sda1** is the first partition on **sda** drive, **sda2** is the second partition on **sda** drive, and **sda5** is the third partition on **sda** drive. **sdb** has 2 partitions: **sdb1** (4 GB) and **sdb2** (6 GB) is the only partition on **sdb** (of size 4 GiB). **sdc** is a USB flash drive and it has one partition **sdc1**.

NOTE 2-6: Remark that **/dev/sda1** partition is the booting partition (a start is placed under the Boot header).

STEP 3: Use the **lshw** command to display hardware information. If **lshw** is not found, use **sudo apt install lshw** to install it.

```
# Display disk (drive) information
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/c/0/2/0.0.0      /dev/sdc    disk        15GB  DataTraveler 2.0
/0/100/c/0/2/0.0.0/0    /dev/sdc    disk        15GB
/0/100/d/0              /dev/sda    disk        85GB  VBOX HARDDISK
/0/100/d/1              /dev/sdb    disk        10GB  VBOX HARDDISK
```

STEP 4: Use the **lshw** command to display hardware information.

```
# Display volume (partition) information
kali@kali [~] $ sudo lshw -class volume -short
H/W path                Device      Class       Description
=====================================================
/0/100/c/0/2/0.0.0/0/1  /dev/sdc1   volume      14GiB Windows FAT volume
/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      4GiB Windows FAT volume
/0/100/d/1/2            /dev/sdb2   volume      6143MiB EXT4 volume
```

Part II: Linux Hashing Commands

STEP 5: Linux has several build-in commands to hash files, drives, or strings. Available commands: **md5sum** (128 bits), **sha1sum** (160 bits), **sha224sum**, **sha256sum**, **sha384sum**, and **sha512sum**.

```
# hash a string
kali@kali [~] $ printf cs362 | sha1sum
ee337f581bdf94a9270c7d6ac33acb58659d40a2  -

kali@kali [~] $ printf cs362 | md5sum
21e807599f8ec807297d3f9d9bcbb635  -

kali@kali [~] $ printf cs362 | sha512sum
be47fe03860b2c7330b2d15bb7911fbd4b5e73327b35d1a1857537948f92fbe3aaf28fb56bc595d
5d8f0a9fdf580fb294840f33a2df3c4fd46f07cc2cfefbd97  -

# hash a file: create a file with the content "this is a text file" and hash it
kali@kali [~] $ echo this is a text file > file1.txt
kali@kali [~] $ md5sum file1.txt
f518eceba46f10d7a008e6aee90d42f4  file1.txt

# hash all the files in a directory
kali@kali [~] $ md5sum Downloads/*
6b5aa64320761e647192b57a1083af8d  Downloads/image1.jpg
6b5aa64320761e647192b57a1083af8d  Downloads/image2.jpg
bee31199176666f434f5cd02eb0bcfc9  Downloads/sift-cli-linux.sig
```

```
708aabed96ff57c89f7dc276b734e783    Downloads/sift-cli.pub
```

NOTE 5-1: SHA stands for secure hash algorithm. **SHA-2** is a set of cryptographic hash functions that includes **SHA-224**, **SHA-256**, **SHA-384**, and **SHA-512**.

STEP 6: SHA-3 is another set of powerful hash functions, but there are no built-in commands for them. The **openssl** software library includes **SHA-3** hash functions among many other algorithms.

```
# hash a string using SHA3-256
kali@kali [~] $ printf cs-362 | openssl dgst -sha3-256
(stdin)= 4d63918983720f86be63147b99b44b402b397110cd93933af307422417095937

kali@kali [~] $ openssl dgst -sha3-256 Downloads/*
SHA3-256(Downloads/image1.jpg)=
59730559fe073706a2b0ec4c850f9ca67bb31621378b7f04978c6c3002365647
SHA3-256(Downloads/image2.jpg)=
59730559fe073706a2b0ec4c850f9ca67bb31621378b7f04978c6c3002365647
SHA3-256(Downloads/sift-cli-linux.sig)=
2bd0a704170129aa4426000938c90868e077443016f73a9e7e4669f23f92fca7
SHA3-256(Downloads/sift-cli.pub)=
d1731751dae89132975d8fcc44b7033d16ca9f1971abec38926aadf6e76e9fc0
```

Part III: Image Acquisition using dc3dd and dd Commands

STEP 7: dd, dc3dd, and dcfldd perform similar functionalities. The latter two have additional functionalities. You can install dc3dd using the command **sudo apt-get install dc3dd**. You can use the dc3dd command to create a raw image of the **/dev/sdc/** drive (USB flash drive).

```
# hash a string using SHA3-256
kali@kali [~] $ sudo dc3dd if=/dev/sdc hash=sha1 log=usb_forensics.log
of=usb_image.dd

dc3dd 7.2.646 started at 2021-01-01 02:31:09 -0500
compiled options:
command line: dc3dd if=/dev/sdc hash=sha1 log=usb_forensics.log of=usb_image.dd
device size: 30240768 sectors (probed),    15,483,273,216 bytes
sector size: 512 bytes (probed)
 15483273216 bytes ( 14 G ) copied ( 100% ), 1155 s, 13 M/s

input results for device `/dev/sdc':
    30240768 sectors in
    0 bad sectors replaced by zeros
    2f9e8a2a37ba027daa7438e86b0fbb71154a05c4 (sha1)
```

```
output results for file `usb_image.dd':
    30240768 sectors out

dc3dd completed at 2021-01-01 02:50:24 -0500
```

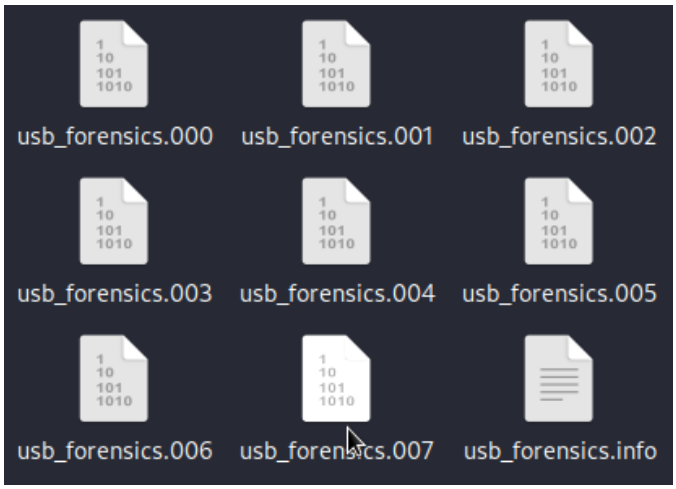
- **if**: input file/drive
- **hash**: hashing algorithm
- **log**: the name of the log file that will contain the log of the acquisition process
- **of**: the output file's name

STEP 8: As a result of the above command, two files will be created: **usb_forensics.log** (log file) **usb_image.dd** (the raw image). Verify the content of the log file.

STEP 9: If the drive size is large, it is a good idea to split the created image into several files, which will be easy to process and transfer. To do that, type the following command:

```
# hash a string using SHA3-256
kali@kali [~] $ sudo dc3dd if=/dev/sdc hash=sha1 log=usb_forensics.info
ofsz=550M ofs=usb_forensics.000
```

- **ofsz**: the size of each of the output files
- **ofs**: output files. The output files will be named **usb_forensics.000**, **usb_forensics.001**, **usb_forensics.002**, etc.



(**NOTE**: The above screenshot is the output of the image acquiring using guymager, which results in exact output of dc3dd, apart from the content of the .info file)

STEP 10: You can compute the hash function of the split files as follows.

```
# compute the hash digest of a set of files
kali@kali [~] $ cat usb_forensics.0* sha1sum
2f9e8a2a37ba027daa7438e86b0fbb71154a05c4     -
```

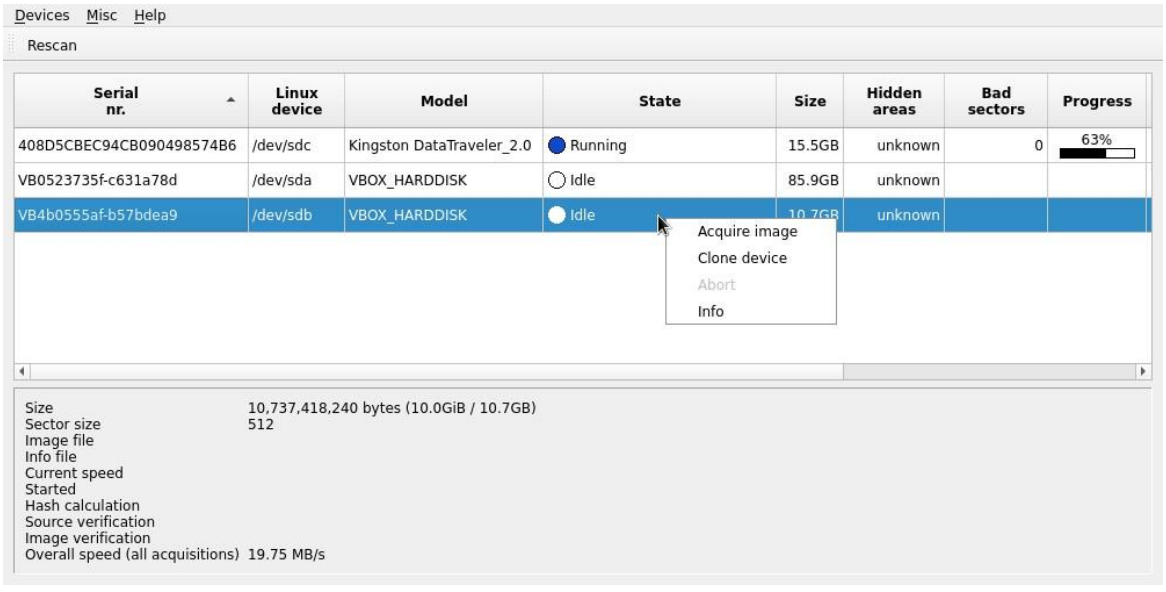
This hash value should be identical to the value obtained when the raw image was created. Check LINE 5 of the code snippet in STEP 7.

STEP 11: **dc3dd** can be used to wipe the contents of a drive to avoid recovery of deleted files. To do that, type the following commands.

```
# 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
```



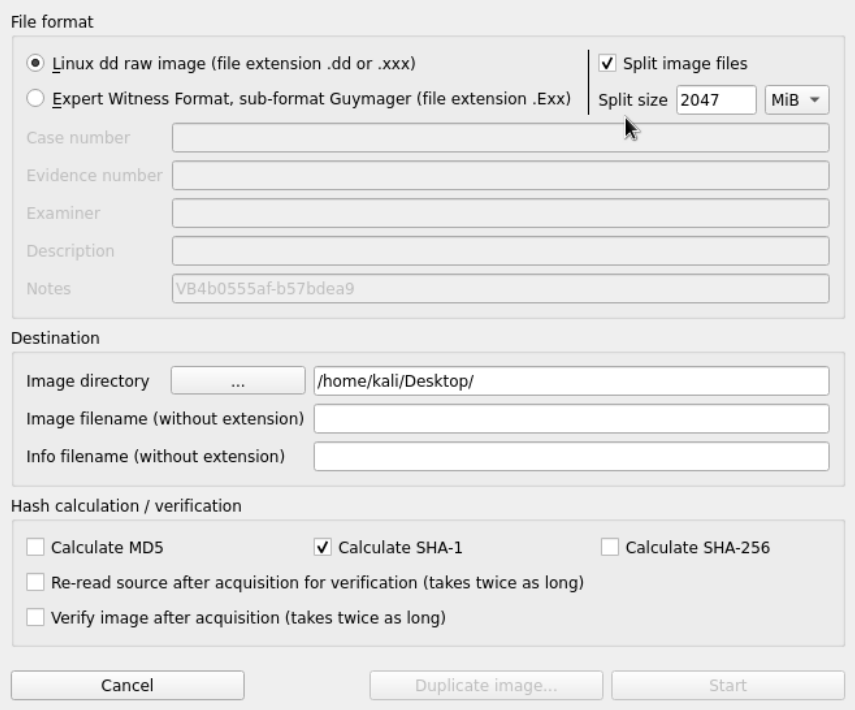
Guymager first interface.

Part IV: Image Acquisition using Guymager

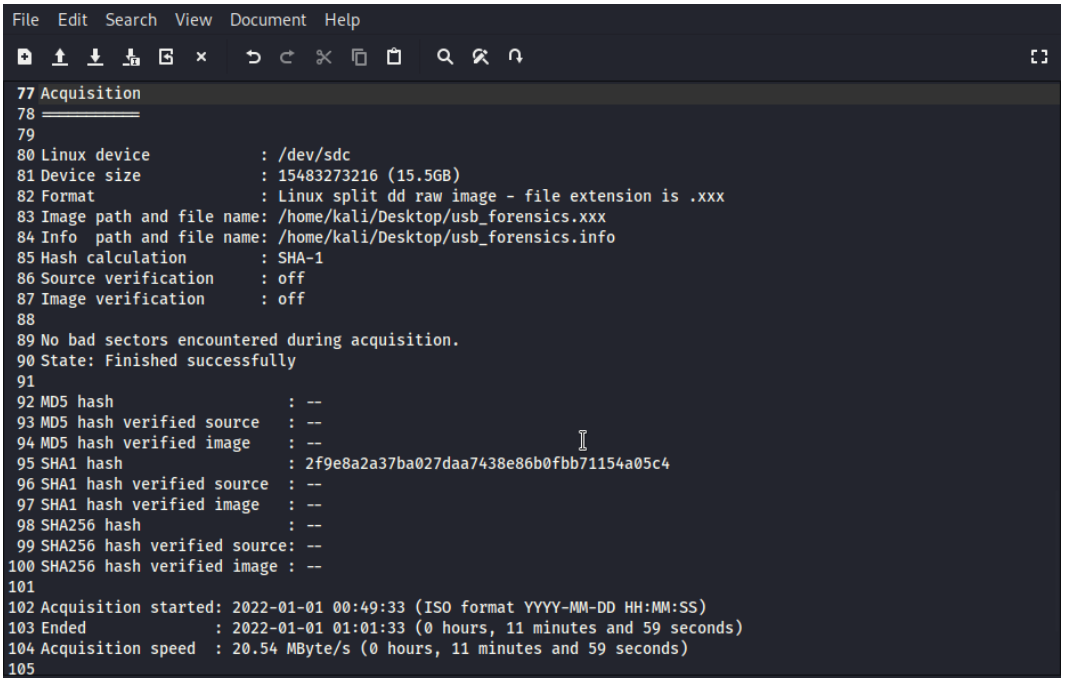
STEP 12: Run guymager as follows.

```
kali@kali [~] $ sudo guymager
```

The guymager displays the list of detected drives, along with their states, sizes, among other information (refer to the snapshot at the end of the previous page). Right-click on the **/dev/sdc** drive and select ACQUIRE IMAGE. The following snapshot is the displayed interface. You can select the format of the raw image file (.dd, .xxx, or .Exx, where x is a placeholder for a number). You should specify the image file's name and the information file's name (.info).



Guymager's "Acquire image" interface.



usb_forensics.info

Part V: Retrieve the Master Boot Record (MBR) using the dd Command

STEP 12: The master boot record is stored in the first sector of the booting drive (/dev/sda in our case). You can retrieve the first cluster of the drive as follows.

```
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, 0.000113894 s, 4.5 MB/s
```

- **bs**: block size in bytes (this is equivalent to one sector)
- **count**: the number of blocks of 512 bytes to be read (only one is needed as the MBR occupies the first sector)
- **1+0 :** 1 complete block and 0 partial blocks were processed. If the copied part is 600 bytes, the result will 1+1 (one complete block of 512 bytes and 1 partial block of 88 bytes).

STEP 12: Open the file mbr.image in HxD and examine its content.

mbr.image

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	63	90	10	8E	D0	BC	00	B0	B8	00	00	8E	D8	8E	C0	ëc..ŽĐ¼.°¸..ŽØŽÀ
00000010	FB	BE	00	7C	BF	00	06	B9	00	02	F3	A4	EA	21	06	00	û¾.|¿..¹..ó¤ê!..
00000020	00	BE	BE	07	38	04	75	0B	83	C6	10	81	FE	FE	07	75	.¾¾.8.u.ƒÆ..þþ.u
00000030	F3	EB	16	B4	02	B0	01	BB	00	7C	B2	80	8A	74	01	8B	óë.´.°.».|²€Št.‹
00000040	4C	02	CD	13	EA	00	7C	00	00	EB	FE	00	00	00	00	00	L.Í.ê.|..ëþ.....
00000050	00	00	00	00	00	00	00	00	00	00	00	80	01	00	00	00	.....€....
00000060	00	00	00	00	FF	FA	90	90	F6	C2	80	74	05	F6	C2	70	....ÿú..öÂ€t.öÂp
00000070	74	02	B2	80	EA	79	7C	00	00	31	C0	8E	D8	8E	D0	BC	t.²€êy|..1ÀŽØŽĐ¼
00000080	00	20	FB	A0	64	7C	3C	FF	74	02	88	C2	52	BE	80	7D	. û d|<ÿt.^ÂR¾€}
00000090	E8	17	01	BE	05	7C	B4	41	BB	AA	55	CD	13	5A	52	72	è..¾.|´A»ªUÍ.ZRr
000000A0	3D	81	FB	55	AA	75	37	83	E1	01	74	32	31	C0	89	44	=.ûUªu7ƒá.t21À‰D
000000B0	04	40	88	44	FF	89	44	02	C7	04	10	00	66	8B	1E	5C	.@^Dÿ‰D.Ç...f‹.\
000000C0	7C	66	89	5C	08	66	8B	1E	60	7C	66	89	5C	0C	C7	44	|f‰\.f‹.`|f‰\.ÇD
000000D0	06	00	70	B4	42	CD	13	72	05	BB	00	70	EB	76	B4	08	..p´BÍ.r.».pëv´.
000000E0	CD	13	73	0D	5A	84	D2	0F	83	D8	00	BE	8B	7D	E9	82	Í.s.Z„Ò.ƒØ.¾‹}é,
000000F0	00	66	0F	B6	C6	88	64	FF	40	66	89	44	04	0F	B6	D1	.f.¶Æ^dÿ@f‰D..¶Ñ
00000100	C1	E2	02	88	E8	88	F4	40	89	44	08	0F	B6	C2	C0	E8	Áâ.^è^ô@‰D..¶ÂÀè
00000110	02	66	89	04	66	A1	60	7C	66	09	C0	75	4E	66	A1	5C	.f‰.f¡`|f.ÀuNf¡\
00000120	7C	66	31	D2	66	F7	34	88	D1	31	D2	66	F7	74	04	3B	|f1Òf÷4^Ñ1Òf÷t.;
00000130	44	08	7D	37	FE	C1	88	C5	30	C0	C1	E8	02	08	C1	88	D.}7þÁ^Å0ÀÁè..Á^
00000140	D0	5A	88	C6	BB	00	70	8E	C3	31	DB	B8	01	02	CD	13	ĐZ^Æ».pŽÃ1Û¸..Í.
00000150	72	1E	8C	C3	60	1E	B9	00	01	8E	DB	31	F6	BF	00	80	r.ŒÃ`.¹..ŽÛ1ö¿.€
00000160	8E	C6	FC	F3	A5	1F	61	FF	26	5A	7C	BE	86	7D	EB	03	ŽÆüó¥.aÿ&Z|¾†}ë.
00000170	BE	95	7D	E8	34	00	BE	9A	7D	E8	2E	00	CD	18	EB	FE	¾•}è4.¾š}è..Í.ëþ
00000180	47	52	55	42	20	00	47	65	6F	6D	00	48	61	72	64	20	GRUB .Geom.Hard
00000190	44	69	73	6B	00	52	65	61	64	00	20	45	72	72	6F	72	Disk.Read. Error
000001A0	0D	0A	00	BB	01	00	B4	0E	CD	10	AC	3C	00	75	F4	C3	...».´.Í.¬<.uôÃ
000001B0	00	00	00	00	00	00	00	00	E4	73	BA	95	00	00	80	04	........äsº•..€.
000001C0	01	04	83	FE	C2	FF	00	08	00	00	00	70	E1	09	00	FE	..ƒþÂÿ.....pá..þ
000001D0	C2	FF	05	FE	C2	FF	FE	7F	E1	09	02	78	1E	00	00	00	Âÿ.þÂÿþ.á..x....
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	.....Uª

Part VI: Windows Disk Investigation with PowerShell Cmdlets

STEP 13: In this part of the lab, we will learn how to search for, install, and import PowerShell modules and cmdlets. Let us start by searching for PowerShell gallery using the keyword “forensic”. A list of modules and scripts will be listed as a result of your search.

STEP 14: Open a PowerShell terminal and search for modules with names including the string “forensic”.

```
PS C:\tools>  Find-Module -Name *forensic*
   Version    Name                           Repository           Description
   -------    ----                           ----------           -----------
   1.1.1      PowerForensics                 PSGallery            A Digital Forens...
   1.1.1      PowerForensicsv2               PSGallery            A Digital Forens...
   1.1.1      PowerForensicsPortable         PSGallery            A Digital Forens...
   1.0.0.0    Forensics                      PSGallery            The module can b...
```

STEP 15: Install the PowerForensics module by typing the following command:

```
PS C:\tools> Install-Module -Name PowerForensics
```

STEP 15: Installed modules are found at **C:\Program Files\WindowsPowerShell\Modules**. List installed modules by typing the following command. **Get-ChildItem** is the PowerShell cmdlet for command prompt **dir** or Linux **ls** commands. Check the parameters -**Force** (show hidden or system items) and **-Recurse** (show subfolders and their contents) with the command **Get-ChildItem**.

```
PS C:\tools> 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-----        1/30/2022    4:24 PM                PowerForensics
d-----         6/5/2021   7:10 AM                PowerShellGet
d-----         6/5/2021   7:10 AM                PSReadline
```

STEP 15: To import the PowerForensics module and list its contained cmdlets, type the following two commands:

```
PS C:\tools> Import-Module -Name PowerForensics
PS C:\tools> 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
...
```

STEP 16: Use the cmdlet Get-**ForensicVolumeBootRecord** to get the master boor record of a volume. Type the following command:

```
PS C:\tools> 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  δRENTFS    .....
00000010   00 00 00 00 00 F8 00 00 3F 00 FF 00 00 A8 03 00  .....°..?....¿..
00000020   00 00 00 00 80 00 80 00 D7 E2 5A 74 00 00 00 00  ....?.?.╫ΓZt....
00000030   00 00 0C 00 00 00 00 00 02 00 00 00 00 00 00 00  ................
...
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  ......?.º.┐ ...U¬
```

STEP 16: Use the cmdlet Get-ForensicFileRecord to get the file records from the master file table (MFT). The MFT is a database stored on an NTFS volume and contains information about every file.To get the MFT entry for a given file (I use an image file in this example), type the flowing command:

```
PS C:\tools> Get-ForensicFileRecord -Path C:\tools\images\dog2.png

FullName              : c:\\tools\images\dog2.png
Name                  : dog2.png
SequenceNumber        : 11
RecordNumber          : 15374
ParentSequenceNumber  : 62
ParentRecordNumber    : 15986
Directory             : False
Deleted               : False
ModifiedTime          : 1/25/2022 3:37:18 AM
AccessedTime          : 1/26/2022 3:15:50 AM
ChangedTime           : 1/25/2022 5:50:10 AM
BornTime              : 1/25/2022 3:37:17 AM
FNModifiedTime        : 1/25/2022 5:50:02 AM
FNAccessedTime        : 1/25/2022 5:49:47 AM
FNChangedTime         : 1/25/2022 3:37:18 AM
FNBornTime            : 1/25/2022 3:37:17 AM
```