Acquiring RAID Disks (Cont'd)

Identifying RAID Drives in Linux system

- Command to check whether RAID is configured:

```
lspci | grep RAID
```

- Command to obtain essential information about active RAID devices:

```
cat /etc/mdadm.conf
```

- Command to check the current status of RAID devices:

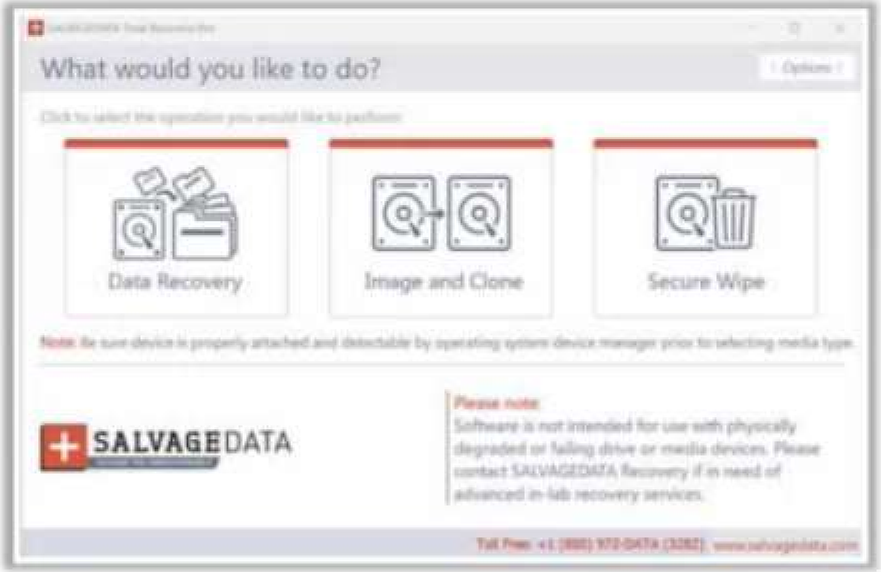
```
cat /proc/mdstat
```

- Command to examine the details of the RAID device:

```
mdadm --detail /dev/md125
```

Rebuilding RAID

SalvageData Total Recovery Pro	It is a RAID recovery tool that can help investigators to recover deleted or lost files from hard drives or external storage devices



Click to select the operation you would like to perform:

Note: Be sure device is properly attached and detectable by operating system device manager prior to selecting media type.

Please note:
Software is not intended for use with physically degraded or failing drive or media devices. Please contact SALVAGEDATA Recovery if in need of advanced in-lab recovery services.

Toll Free: +1 (800) 972-DATA (3282), www.salvagedata.com

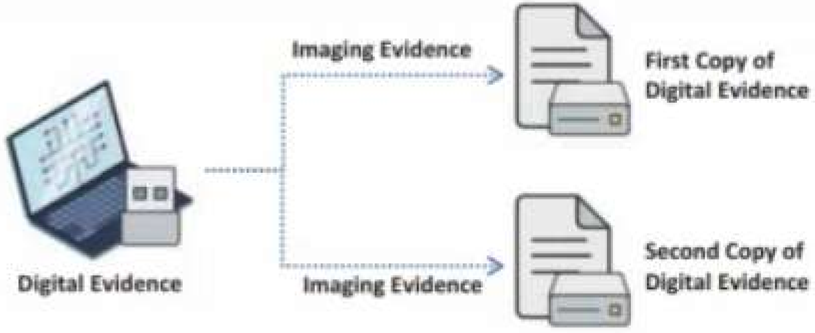
https://www.salvagedata.com

Step 7: Plan for Contingency

- Investigators must be prepared for contingencies such as when the **hardware or software does not work** or failure occurs during acquisition

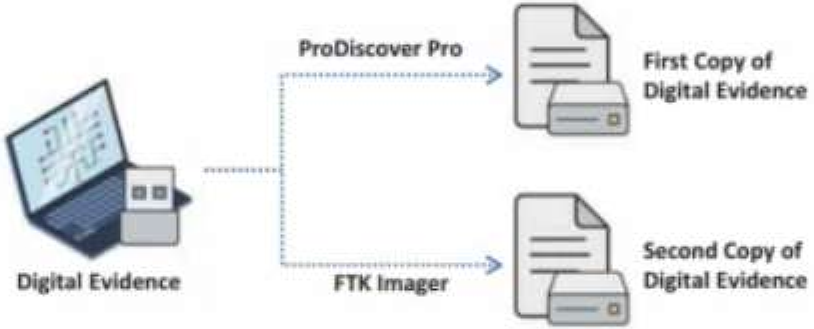
Hard Disk Data Acquisition

- Create at least **two images** of the digital evidence collected
- If one copy of the digital evidence recovered becomes **corrupt**, investigators can then use the other copy


Imaging Evidence
First Copy of Digital Evidence
Digital Evidence
Imaging Evidence
Second Copy of Digital Evidence

Imaging Tools

- Use two or more imaging tools such as **FTK Imager** and **ProDiscover Pro** to create two images of the evidence
- If the investigator has access to only one tool, create two or more **images** of the drive using the same tool


ProDiscover Pro
First Copy of Digital Evidence
Digital Evidence
FTK Imager
Second Copy of Digital Evidence

Step 7: Plan for Contingency (Cont'd)

Hardware Acquisition Tools

- Use hardware acquisition tools such as **UFED Ultimate** or **IM Solo-4 PLUS IT Enterprise** that can access the drive at the **BIOS level** to copy data in the Host Protected Area (HPA)

Accessing Drive at BIOS Level
Hardware Acquisition Tool
Hard Disk

Drive Decryption

- Investigators must be prepared to deal with encrypted drives that require decryption keys from users
- Microsoft Windows has a full disk encryption feature such as **BitLocker** in some selected versions

Decrypting Key
Encrypted Drive
Decrypted Drive

Step 8: Validate Data Acquisition

- Validating data acquisition involves calculating the hash value of the target media and comparing it with its forensic counterpart to ensure that the data have been **completely acquired**

- Hash value calculation generates a **unique numeric value** for files, often considered as "digital footprint," which represents the uniqueness of a file or disk drive

- Some hashing algorithms that can be used to **validate** the data acquired include CRC-32, MD5, SHA-1, SHA-3, SHA-256, SHA-512, and BLAKE2

Acquiring RAID Disks

- Forensic examiners may encounter challenges when acquiring data from RAID disks, primarily owing to the intricate design, complex configurations, and considerable storage capacities associated with RAID systems

Consider the following factors before acquiring data:

- The amount of data storage required to acquire all data
- The RAID format used (RAID 0, RAID 1, RAID 5, RAID 10, etc.)
- Check whether RAID is managed by hardware or software
- Forensic tools suitable for imaging RAID disks and reading these images
- Decide whether to perform disk imaging (bit-by-bit) or logical acquisition (file-level) based on the RAID configuration
- Ensure that the forensic hardware and software tools are compatible with the RAID controller in use
- Reconstruct the RAID array on a separate forensic workstation or by using specialized RAID recovery tools
- Calculate and compare checksums of the acquired data with the original RAID array to verify data integrity

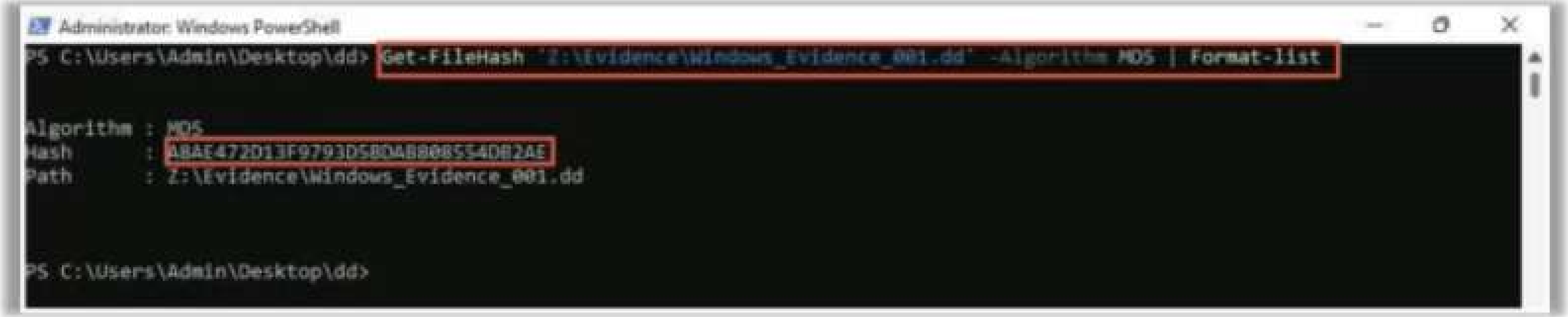
- Several computer forensics tools such as OpenText EnCase Forensic, X-Ways Forensics, and ProDiscover are built with capabilities to recover RAID disks

Step 8: Validate Data Acquisition – Windows Validation Methods

- The **Get-FileHash** cmdlet computes the hash value for an evidence file using the specified hash algorithm

- This hash value is used throughout the investigation for **validating** the integrity of the evidence

- Investigators can also use commercial computer forensic programs such as **ProDiscover**, which have built-in validation features that can be used to validate evidence files



```
PS C:\Users\Admin\Desktop\dd> Get-FileHash 'Z:\Evidence\Windows_Evidence_001.dd' -Algorithm MD5 | Format-list


Algorithm : MD5
Hash      : ABAE472D13F9793D58DA8808554D82AE
Path      : Z:\Evidence\Windows_Evidence_001.dd


PS C:\Users\Admin\Desktop\dd>
```

Step 8: Validate Data Acquisition – Linux/Mac Validation Methods

Validating Data Acquired with dd

- Run the following command to acquire an image in a single file:

  `dd if=/dev/sda of=/image_sda.dd`

- Now, you can use the **md5sum** utility to validate the image
- Run the following command to calculate the hash of the original drive: `md5sum /dev/sda > md5_hashes.txt`
- Run the `command cat image_sda.dd| md5sum << md5_hashes.txt` to calculate the MD5 hash for the image file and generate the output to the `md5_hashes.txt` file

Validating dcfldd Acquired Data

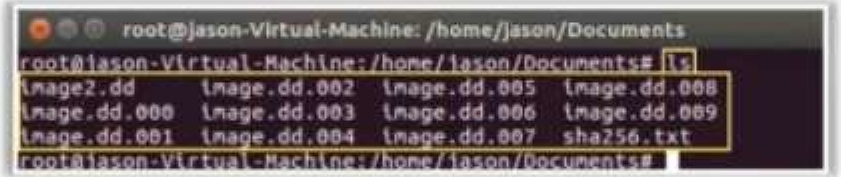
- Enter the following command in the terminal to create an image and calculate **sha256 hash post-data acquisition**:

  ```
  dcfldd if=/dev/sda split=100M
  of=/media/image.dd hash=sha256
  ```

- Run the following command in the terminal to create an image and store its **sha256** hash value in a text file:

  ```
  dcfldd if=/dev/sda split=100M
  of=/media/image.dd hash=sha256
  hashlog=/media/sha256.txt
  ```

- Navigate to the directory and enter the `ls` command to view the **files generated**



root@jason-Virtual-Machine: /home/jason/Documents

```
root@jason-Virtual-Machine:/home/jason/Documents# ls
image2.dd      image.dd.002  image.dd.005  image.dd.008
image.dd.000   image.dd.003  image.dd.006  image.dd.009
image.dd.001   image.dd.004  image.dd.007  sha256.txt
root@jason-Virtual-Machine:/home/jason/Documents#
```

Data Acquisition Guidelines and Best Practices

1. Define the **purpose** and data **requirements** of data acquisition
2. Identify **data sources** to obtain evidence
3. Devise a suitable data acquisition **strategy**
4. Select appropriate **data acquisition tools** considering the requirements
5. Gather only **relevant** data to mitigate the risk of violating an individual's privacy rights
6. Capture **volatile evidence first** and then proceed to non-volatile data
7. Minimize data **duplication** and examine the **sensitivity** of fresh datasets
8. Comply with **legal** and **ethical** data protection laws and regulations
9. Document the entire data **acquisition process**
10. Establish access controls, read-only mode, and encryption techniques to maintain **data security**
11. Use the data for their **original purpose** only and avoid misinterpretation
12. Evaluate the **quality** and **efficiency** of the adopted data collection plan regularly

LO#04: Prepare an Image File for Examination

- Preparing an Image for Examination
- Scenario 1: Examining Images on Linux Forensic Workstation
- Scenario 2: Examining Images on Windows Forensic Workstation
- Scenario 3: Examining Images on Mac Forensic Workstation
- Digital Forensic Imaging Tools

Preparing an Image for Examination

- After collecting image files, the investigator should ensure that the **image files are ready for examination**

- Investigators might encounter challenging situations when the file format of the acquired image file is incompatible with the OS used in the forensic workstation

- An investigator might encounter the following **scenarios during investigation**:
  - **Scenario 1:** Examining Images on Linux Forensic Workstation
  - **Scenario 2:** Examining Images on Windows Forensic Workstation
  - **Scenario 3:** Examining Images on Mac Forensic Workstation

Scenario 1: Examining Images on Linux Forensic Workstation

- Linux workstations support many file systems and **contain advanced tools** for conducting forensic investigations

- An investigator might encounter the following **scenarios during investigation** on a Linux forensic workstation:
  - Scenario 1.1: Converting E01 image file to dd image file
  - Scenario 1.2: Converting E01 image file to raw image file
  - Scenario 1.3: Converting dd image file to VHDX file
  - Scenario 1.4: Examining a dd image file
  - Scenario 1.5: Examining physical hard disk
  - Scenario 1.6: Examining Mac APFS image file
  - Scenario 1.7: Examining disk image using PyTSK
  - Scenario 1.8: Examining EWF-formatted disk image using libewf
  - Scenario 1.9: Examining disk image using dfvfs library

Scenario 1.1: Converting E01 Image File to dd Image File

- When an investigator is presented with an **E01 file**, they cannot directly examine it on a Linux workstation
- The E01 file must be **converted to the dd file** format using "**xmount**" to access the mounted volume's files or directory structure

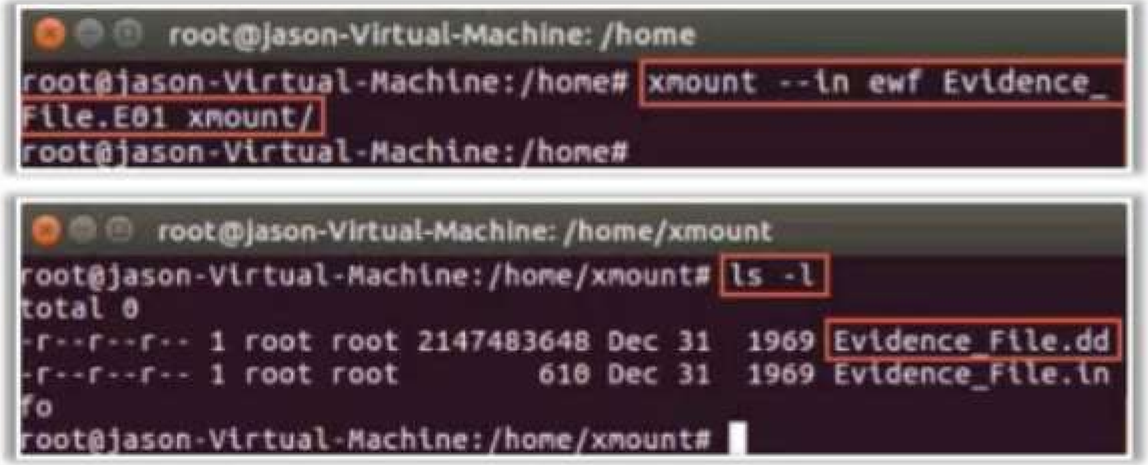
To Convert E01 to dd on Linux

- Use the **xmount** command to convert the E01 image to dd image

  Command:

  ```
  xmount --in [input_image_format]
  [file_name.E01][mount_directory]
  ```

- The converted image file can be viewed in the **xmount directory**, as shown in the screenshot


root@jason-Virtual-Machine: /home
root@jason-Virtual-Machine: /home/xmount

```
root@jason-Virtual-Machine:/home# xmount --in ewf Evidence_
File.E01 xmount/
root@jason-Virtual-Machine:/home#
```

```
root@jason-Virtual-Machine:/home/xmount# ls -l
total 0
-r--r--r-- 1 root root 2147483648 Dec 31  1969 Evidence_File.dd
-r--r--r-- 1 root root        610 Dec 31  1969 Evidence_File.in
fo
root@jason-Virtual-Machine:/home/xmount#
```

E01 file is converted to dd image file

Scenario 1.2: Converting E01 Image File to Raw Image File

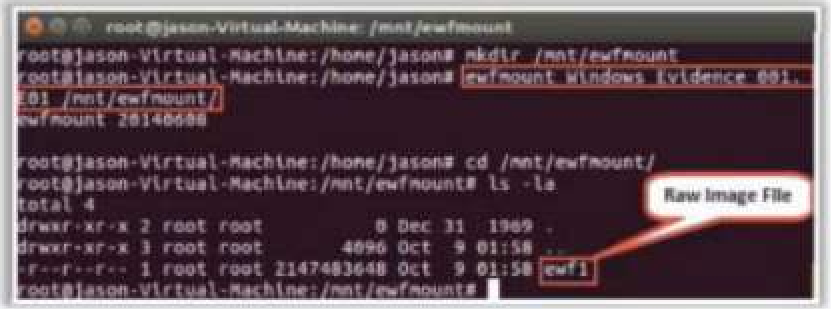
- The E01 file must be **converted to raw image file** format using "**ewfmount**" to access the mounted volume's files or directory structure

Generate Raw Image Using "ewfmount"

- Use **ewfmount** command to generate raw image from E01 image file

Command:

```
ewfmount [file_name.E01]
[mount_directory]
```



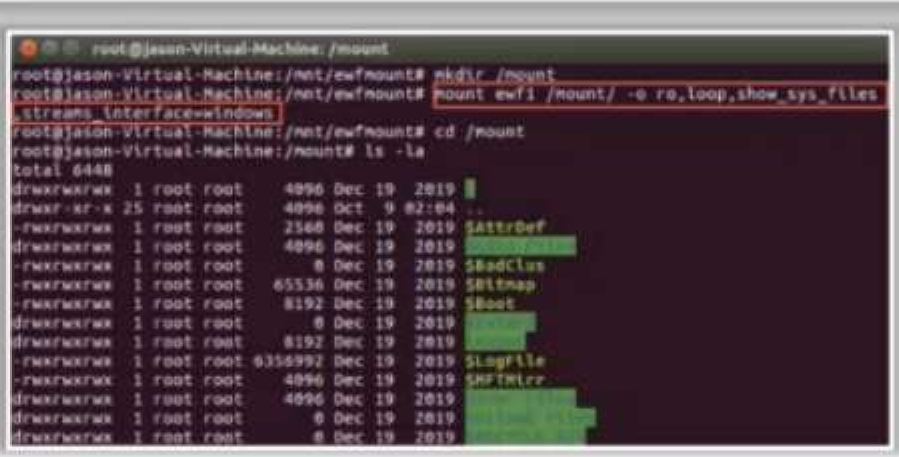
```
root@jason-Virtual-Machine: /mnt/ewfmount
root@jason-Virtual-Machine:/home/jason# mkdir /mnt/ewfmount
root@jason-Virtual-Machine:/home/jason# ewfmount Windows Evidence 001.
E01 /mnt/ewfmount/
ewfmount 20140608

root@jason-Virtual-Machine:/home/jason# cd /mnt/ewfmount/
root@jason-Virtual-Machine:/mnt/ewfmount# ls -la
total 4
drwxr-xr-x 2 root root          0 Dec 31  1969 .
drwxr-xr-x 3 root root       4096 Oct  9 01:58 ..
-r--r--r-- 1 root root 2147483648 Oct  9 01:58 ewf1
root@jason-Virtual-Machine:/mnt/ewfmount#
```

Mount Raw Image Using "mount" command

- Use **mount** command to mount the image file

Command: `mount [raw_image_filename] [mount_directory] -o ro, loop,show_sys_files,streams_interface=windows`



```
root@jason-Virtual-Machine: /mount
root@jason-Virtual-Machine:/mnt/ewfmount# mkdir /mount
root@jason-Virtual-Machine:/mnt/ewfmount# mount ewf1 /mount/ -o ro,loop,show_sys_files
,streams_interface=windows
root@jason-Virtual-Machine:/mnt/ewfmount# cd /mount
root@jason-Virtual-Machine:/mount# ls -la
total 6448
drwxrwxrwx  1 root root    4096 Dec 19  2019 .
drwxr-xr-x 25 root root    4096 Oct  9 02:04 ..
-rwxrwxrwx  1 root root    2560 Dec 19  2019 $AttrDef
drwxrwxrwx  1 root root    4096 Dec 19  2019 [illegible]
-rwxrwxrwx  1 root root       0 Dec 19  2019 $BadClus
-rwxrwxrwx  1 root root   65536 Dec 19  2019 $Bitmap
-rwxrwxrwx  1 root root    8192 Dec 19  2019 $Boot
drwxrwxrwx  1 root root       0 Dec 19  2019 [illegible]
drwxrwxrwx  1 root root    8192 Dec 19  2019 [illegible]
-rwxrwxrwx  1 root root 6356992 Dec 19  2019 $LogFile
-rwxrwxrwx  1 root root    4096 Dec 19  2019 $MFTMirr
drwxrwxrwx  1 root root    4096 Dec 19  2019 [illegible]
drwxrwxrwx  1 root root       0 Dec 19  2019 [illegible]
drwxrwxrwx  1 root root       0 Dec 19  2019 [illegible]
```

Scenario 1.3: Converting dd Image File to VHDX File

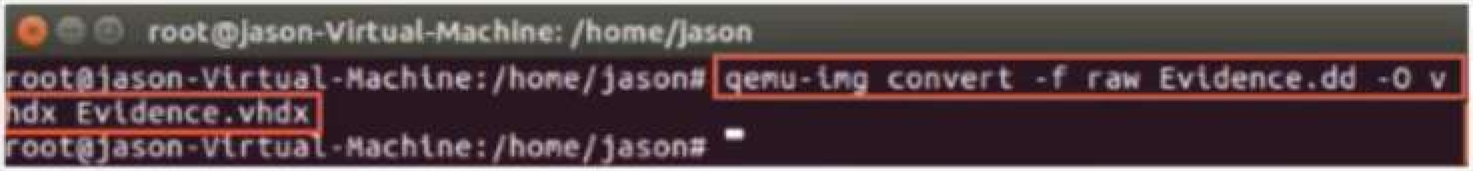
- While performing forensic examination on an image of a system drive, investigators might need to **create a live environment** of the machine to extract additional artifacts that may not be discovered in static analysis
- To do this, the investigator needs to boot the forensically acquired image file as a virtual machine

Step 1: Convert the acquired dd image file into a virtual machine file format using QEMU Disk Image Utility

- **qemu-img** is a command line tool used to create, convert, and modify image files offline
- Assuming Hyper-V to be the virtualization platform used for forensics, we shall convert the dd image to a **vhdx** file
- Use the following command to convert the dd image file to VHDX file

Command:

```
qemu-img convert -f <file format> <Source_Image_filename> -O vhdx <destination_filename.vhdx>
```



```
root@jason-Virtual-Machine: /home/jason
root@jason-Virtual-Machine:/home/jason# qemu-img convert -f raw Evidence.dd -O v
hdx Evidence.vhdx
root@jason-Virtual-Machine:/home/jason# 
```

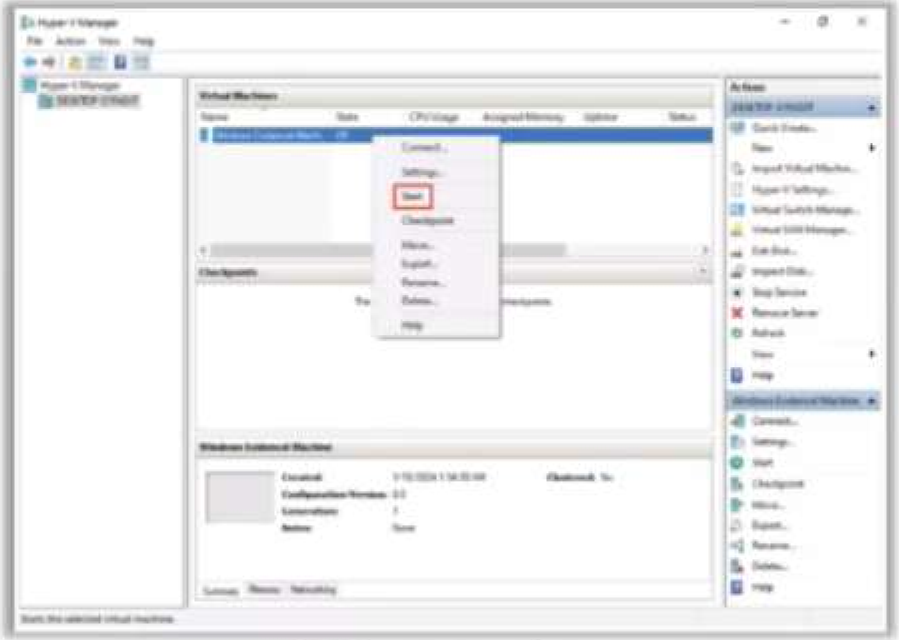
Converting DD image file to VHDX file format

Scenario 1.3: Converting dd Image File to VHDX File (Cont'd)

Step 2: Create a new virtual machine by connecting the vhdx file and start it

Step 3: Boot the virtual machine

- Now, the virtual machine **boots from the forensic image file**
- Upon successful login, the system runs in a live environment, allowing the investigator to perform live analysis



VM is ready to Start

Live environment of the system

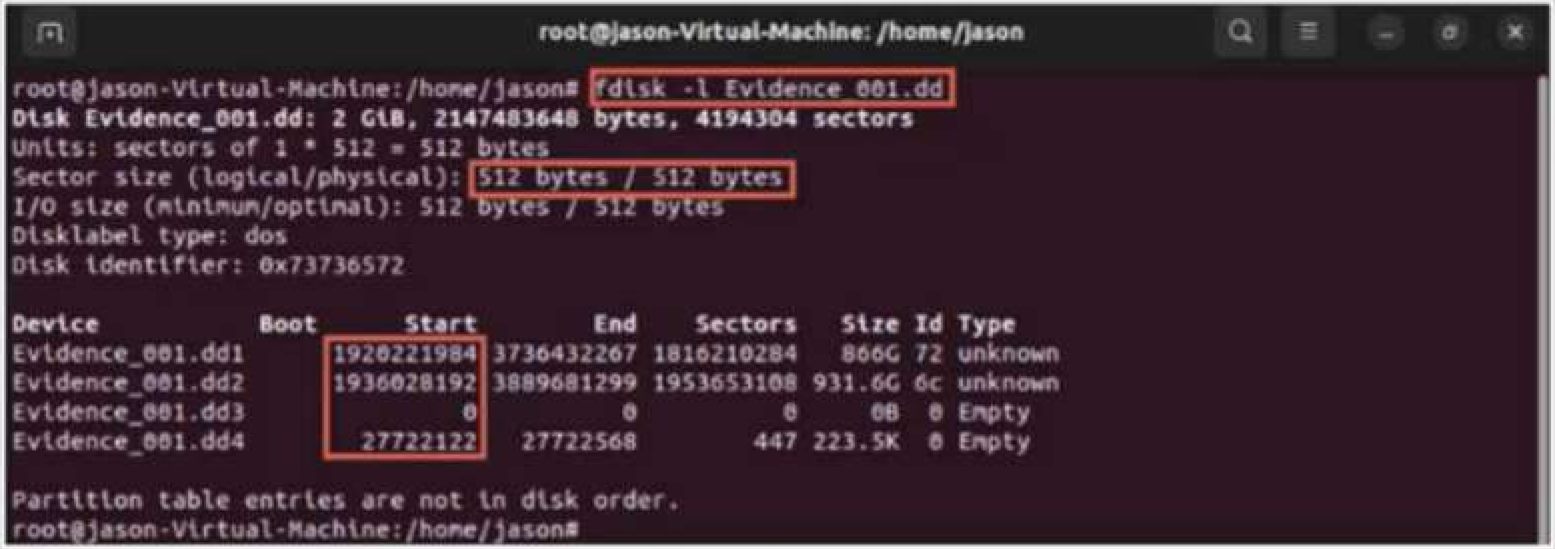
Scenario 1.4: Examining a dd Image File

Step 1

- Use the `fdisk` command to list information such as sector size, start sector, and type of evidence file

  Syntax: `fdisk -l <file_name>`

- The sector start point is required for calculating the offset value before mounting the image file



```
root@jason-Virtual-Machine:/home/jason# fdisk -l Evidence_001.dd
Disk Evidence_001.dd: 2 GiB, 2147483648 bytes, 4194304 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: 0x73736572

Device           Boot      Start        End    Sectors   Size Id Type
Evidence_001.dd1      1920221984 3736432267 1816210284   866G 72 unknown
Evidence_001.dd2      1936028192 3889681299 1953653108 931.6G 6c unknown
Evidence_001.dd3               0          0          0     0B  0 Empty
Evidence_001.dd4        27722122   27722568        447 223.5K  0 Empty

Partition table entries are not in disk order.
root@jason-Virtual-Machine:/home/jason#
```

Scenario 1.4: Examining a dd Image File (Cont'd)

Step 2

- Create a new directory to mount the image file

  Syntax:

  `mkdir [options] directory_name(s)`

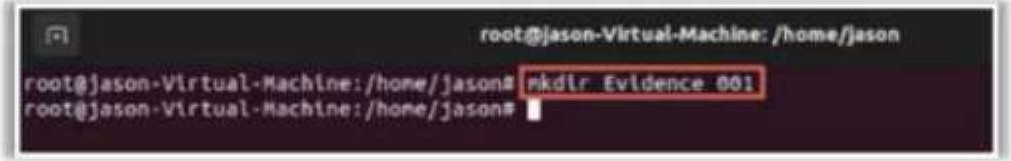
- If the mounted image contains multiple volumes, the user can mount one volume at a time by specifying an "**offset**" to the volume

  Syntax:

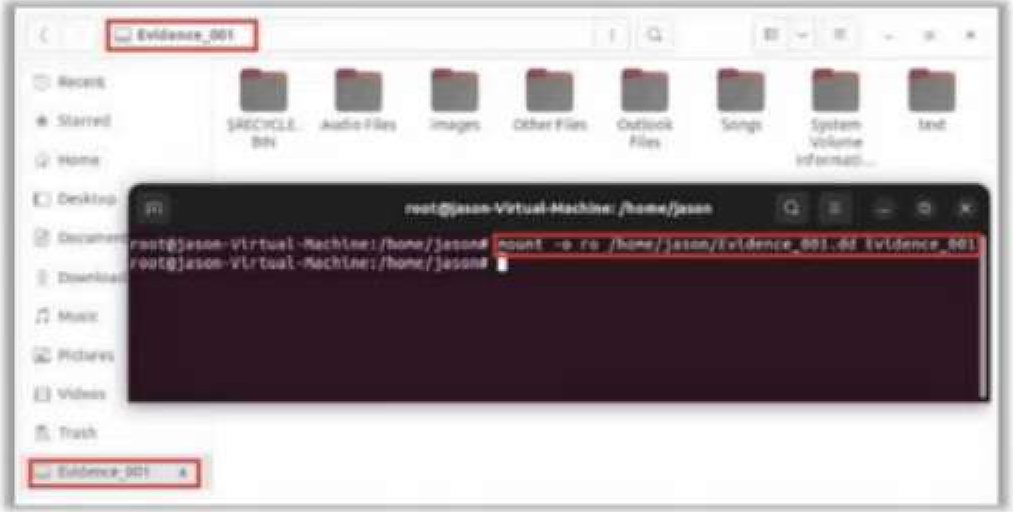
  `mount -t ntfs -o ro,offset=[value_in_bytes] [dd_image_file_name] [mount_directory]`

- Run the `ls -l` command to navigate to the mount point directory and view the files/folders in the mounted volume

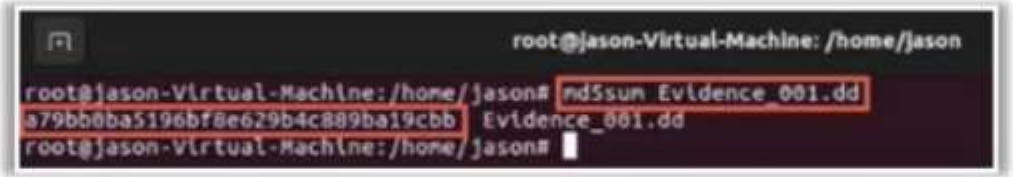
- Now, unmount the volume and **calculate the MD5** hash of the image file and compare it with the computed MD5 hash value of the image file before it was mounted



```
root@jason-Virtual-Machine: /home/jason
root@jason-Virtual-Machine:/home/jason# mkdir Evidence_001
root@jason-Virtual-Machine:/home/jason#
```



```
root@jason-Virtual-Machine: /home/jason
root@jason-Virtual-Machine:/home/jason# mount -o ro /home/jason/Evidence_001.dd Evidence_001
root@jason-Virtual-Machine:/home/jason#
```



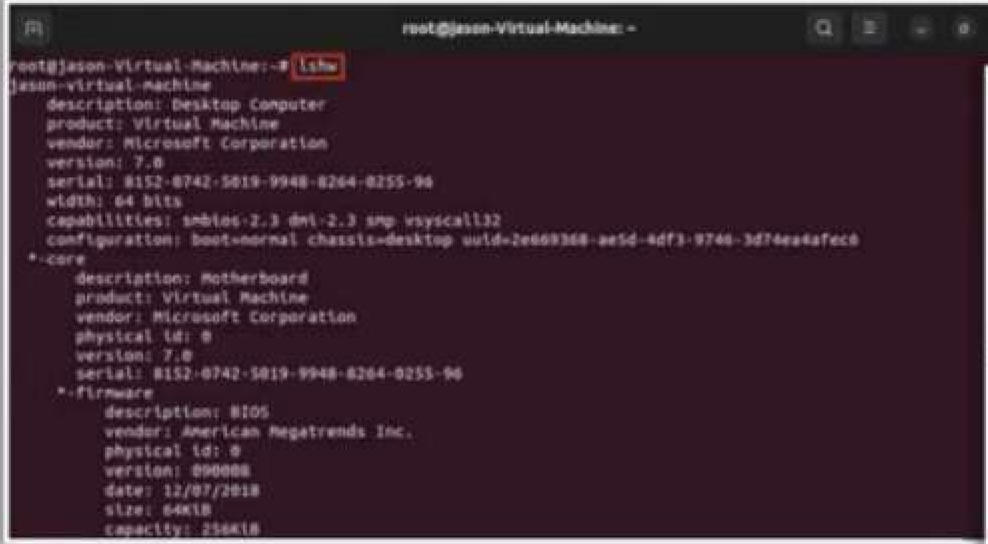
```
root@jason-Virtual-Machine: /home/jason
root@jason-Virtual-Machine:/home/jason# md5sum Evidence_001.dd
a79bb0ba5196bf8e629b4c889ba19cbb  Evidence_001.dd
root@jason-Virtual-Machine:/home/jason#
```

Scenario 1.5: Examining Physical Hard Disk

Step 1: Determine the file system type

- Run the `lshw` command in the command line terminal to view the attached hard disk and file system used in it
- "lshw" is a command-line utility that lists detailed information about various hardware devices available on the machine

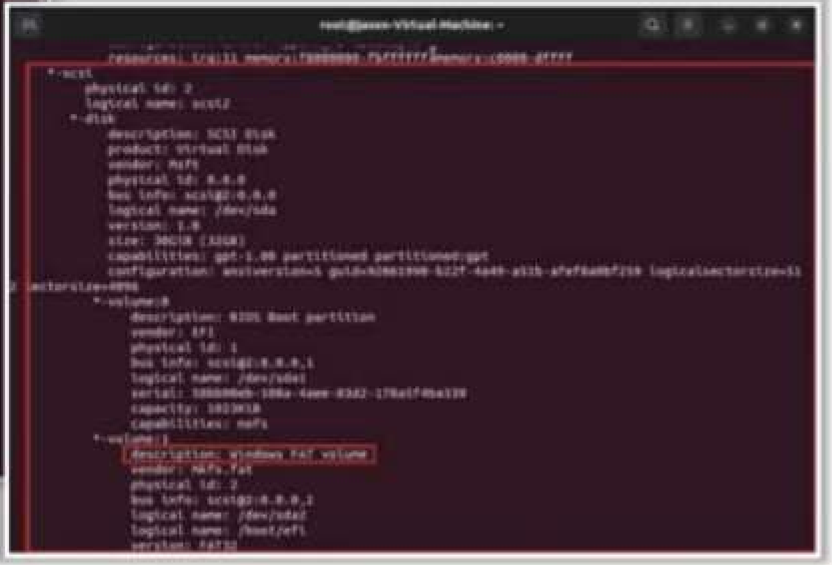
Note: Running the "lshw" command without any "options" generates information about all detected hardware on the machine



```
root@jason-Virtual-Machine:~# lshw
jason-virtual-machine
    description: Desktop Computer
    product: Virtual Machine
    vendor: Microsoft Corporation
    version: 7.0
    serial: 8152-0742-5019-9948-8264-0255-96
    width: 64 bits
    capabilities: smbios-2.3 dmi-2.3 smp vsyscall32
    configuration: boot=normal chassis=desktop uuid=2e669368-ae5d-4df3-9746-3d74ea4afec6
  *-core
       description: Motherboard
       product: Virtual Machine
       vendor: Microsoft Corporation
       physical id: 0
       version: 7.0
       serial: 8152-0742-5019-9948-8264-0255-96
     *-firmware
          description: BIOS
          vendor: American Megatrends Inc.
          physical id: 0
          version: 090008
          date: 12/07/2018
          size: 64KiB
          capacity: 256KiB
```

"lshw" command extracting hardware information

NTFS is mounted

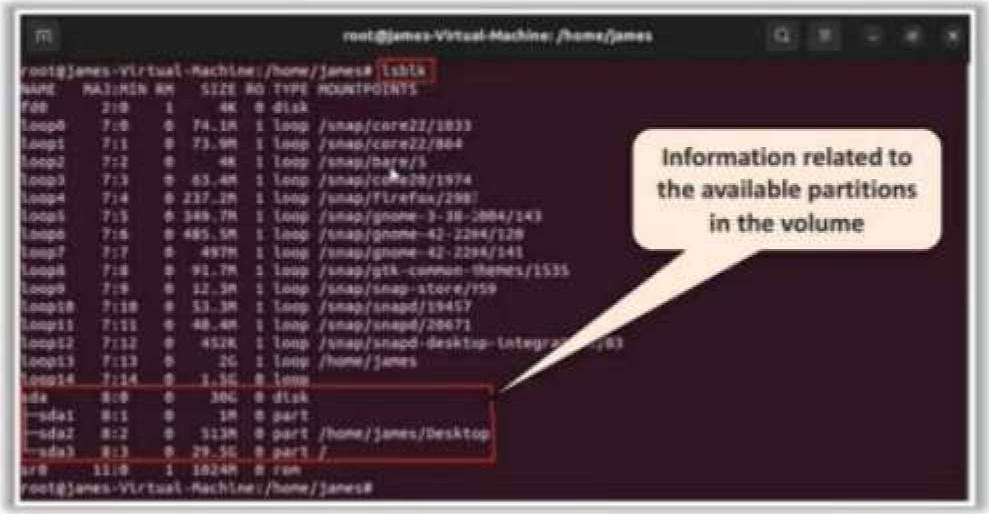


```
             resources: irq:11 memory:f8000000-fbffffff memory:c0000-dffff
     *-scsi
          physical id: 2
          logical name: scsi2
        *-disk
             description: SCSI Disk
             product: Virtual Disk
             vendor: Msft
             physical id: 0.0.0
             bus info: scsi@2:0.0.0
             logical name: /dev/sda
             version: 1.0
             size: 30GiB (32GB)
             capabilities: gpt-1.00 partitioned partitioned:gpt
             configuration: ansiversion=5 guid=[illegible] logicalsectorsize=51
2 sectorsize=4096
           *-volume:0
                description: BIOS Boot partition
                vendor: EFI
                physical id: 1
                bus info: scsi@2:0.0.0,1
                logical name: /dev/sda1
                serial: [illegible]
                capacity: 1023KiB
                capabilities: nofs
           *-volume:1
                description: Windows FAT volume
                vendor: mkfs.fat
                physical id: 2
                bus info: scsi@2:0.0.0,2
                logical name: /dev/sda2
                logical name: /boot/efi
                version: FAT32
```

Scenario 1.5: Examining Physical Hard Disk (Cont'd)

Step 2: List the partitions available on the evidence hard disk

- The lsblk command lists information about all blocked devices connected to the system

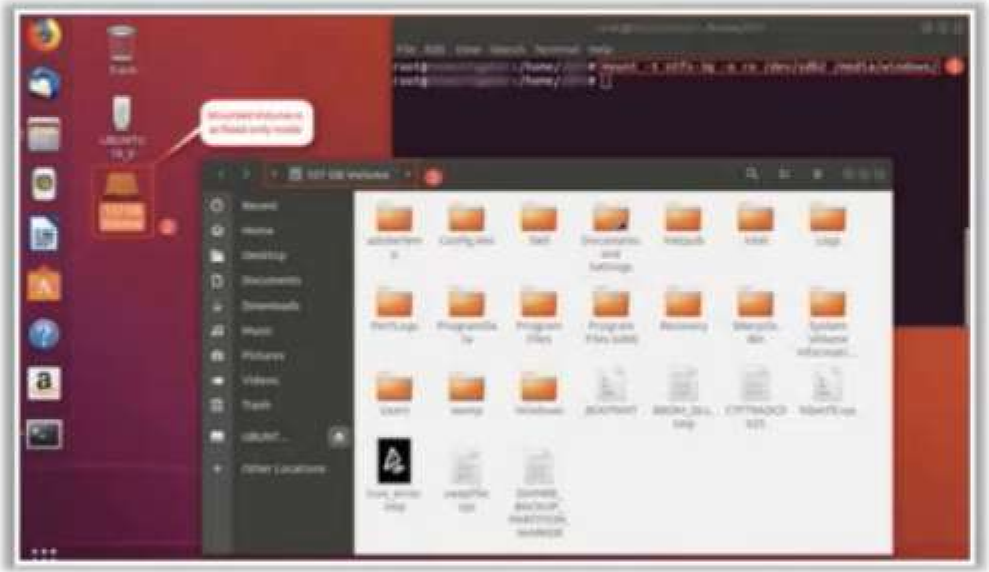

root@james-Virtual-Machine: /home/james
Information related to the available partitions in the volume

```
root@james-Virtual-Machine:/home/james# lsblk
NAME     MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
fd0        2:0    1     4K  0 disk
loop0      7:0    0  74.1M  1 loop /snap/core22/1033
loop1      7:1    0  73.9M  1 loop /snap/core22/864
loop2      7:2    0     4K  1 loop /snap/bare/5
loop3      7:3    0  63.4M  1 loop /snap/core20/1974
loop4      7:4    0 237.2M  1 loop /snap/firefox/2987
loop5      7:5    0 349.7M  1 loop /snap/gnome-3-38-2004/143
loop6      7:6    0 485.5M  1 loop /snap/gnome-42-2204/120
loop7      7:7    0   497M  1 loop /snap/gnome-42-2204/141
loop8      7:8    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop9      7:9    0  12.3M  1 loop /snap/snap-store/959
loop10     7:10   0  53.3M  1 loop /snap/snapd/19457
loop11     7:11   0  40.4M  1 loop /snap/snapd/20671
loop12     7:12   0   452K  1 loop /snap/snapd-desktop-integration/83
loop13     7:13   0     2G  1 loop /home/james
loop14     7:14   0   1.5G  0 loop
sda        8:0    0    30G  0 disk
├─sda1     8:1    0     1M  0 part
├─sda2     8:2    0   513M  0 part /home/james/Desktop
└─sda3     8:3    0  29.5G  0 part /
sr0       11:0    1  1024M  0 rom
root@james-Virtual-Machine:/home/james#
```

Note: Use write blockers before connecting the physical hard disk to the Linux forensic workstation

Step 3: Mount the Windows file system on Linux using "mount" command

- **Command**: `mount -t ntfs-3g -o ro [partition_number] [mount_directory]`

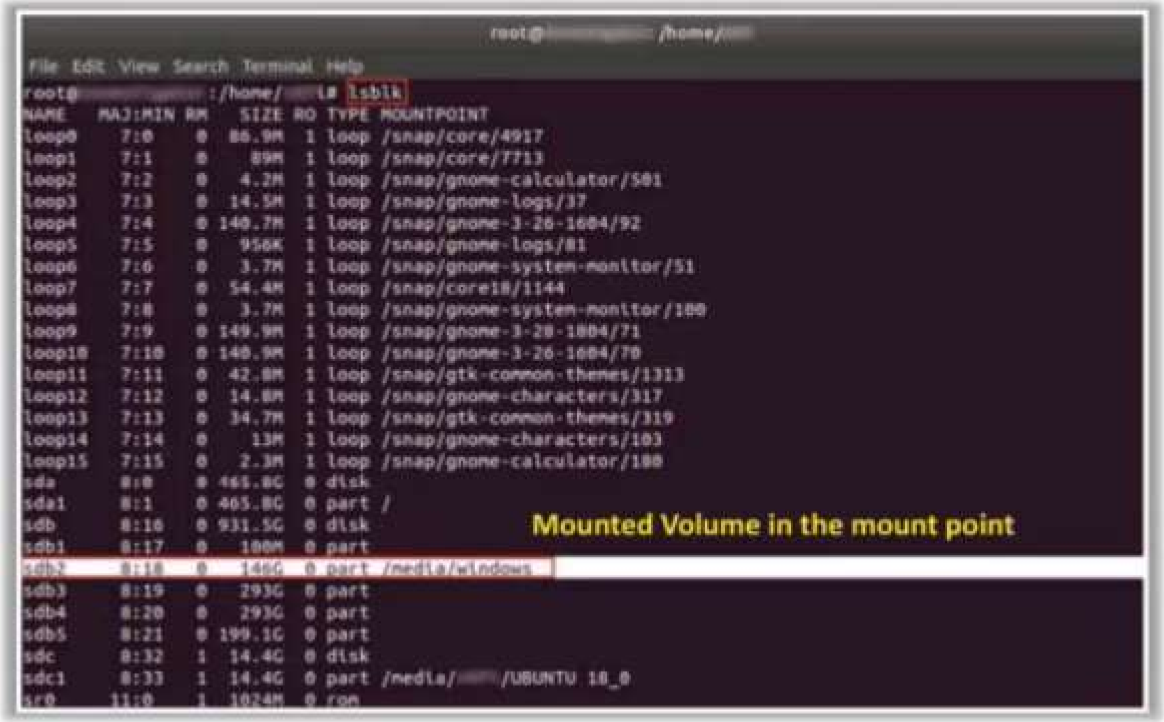


```
root@[illegible]:/home/[illegible]# mount -t ntfs-3g -o ro /dev/sdb1 /media/windows/
root@[illegible]:/home/[illegible]# 
```

1. Mount Command, 2. Mounted Volume in read-only mode, 3. Files/Folders present in the Mounted Volume

Scenario 1.5: Examining Physical Hard Disk (Cont'd)

- After the partition is mounted, the **MOUNTPOINT** for the partition is updated to **/media/windows**


root@[redacted] /home/[redacted]
File Edit View Search Terminal Help

```
root@[redacted]:/home/[redacted]# lsblk
NAME     MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
loop0      7:0    0  86.9M  1 loop /snap/core/4917
loop1      7:1    0    89M  1 loop /snap/core/7713
loop2      7:2    0   4.2M  1 loop /snap/gnome-calculator/501
loop3      7:3    0  14.5M  1 loop /snap/gnome-logs/37
loop4      7:4    0 140.7M  1 loop /snap/gnome-3-26-1604/92
loop5      7:5    0   956K  1 loop /snap/gnome-logs/81
loop6      7:6    0   3.7M  1 loop /snap/gnome-system-monitor/51
loop7      7:7    0  54.4M  1 loop /snap/core18/1144
loop8      7:8    0   3.7M  1 loop /snap/gnome-system-monitor/100
loop9      7:9    0 149.9M  1 loop /snap/gnome-3-28-1804/71
loop10     7:10   0 140.9M  1 loop /snap/gnome-3-26-1604/70
loop11     7:11   0  42.8M  1 loop /snap/gtk-common-themes/1313
loop12     7:12   0  14.8M  1 loop /snap/gnome-characters/317
loop13     7:13   0  34.7M  1 loop /snap/gtk-common-themes/319
loop14     7:14   0    13M  1 loop /snap/gnome-characters/103
loop15     7:15   0   2.3M  1 loop /snap/gnome-calculator/180
sda        8:0    0 465.8G  0 disk
sda1       8:1    0 465.8G  0 part /
sdb        8:16   0 931.5G  0 disk
sdb1       8:17   0   100M  0 part
sdb2       8:18   0   146G  0 part /media/windows
sdb3       8:19   0   293G  0 part
sdb4       8:20   0   293G  0 part
sdb5       8:21   0 199.1G  0 part
sdc        8:32   1  14.4G  0 disk
sdc1       8:33   1  14.4G  0 part /media/[redacted]/UBUNTU 18_0
sr0       11:0    1  1024M  0 rom
```

Mounted Volume in the mount point

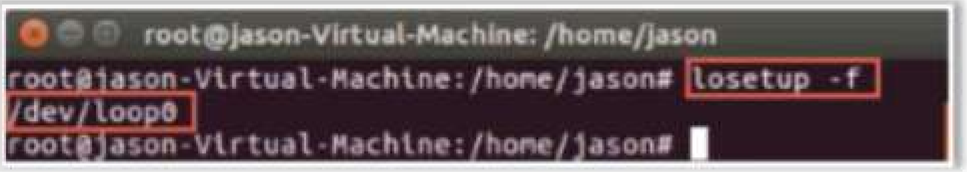
Scenario 1.6: Examining Mac APFS Image File

- When the acquired evidence contains APFS and the forensic workstation is Linux, an investigator can use either mount, losetup, (mount and losetup) together, or (apfs-fuse) or (apfs-fuse and losetup) together to mount the image and view its contents
- In this scenario, we shall be mounting the evidence using losetup and apfs-fuse together

Step 1: Identify Unused Loopback Device

- To mount an image, first identify the unused loopback device
- Accordingly, issue the following command:

```
losetup -f
```

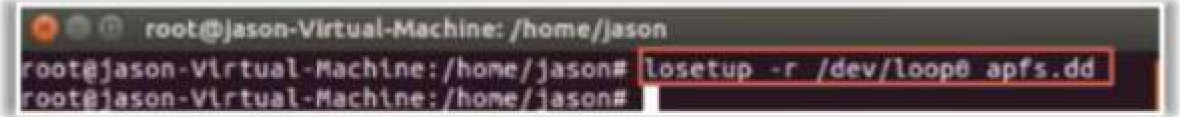


```
root@jason-Virtual-Machine:/home/jason# losetup -f
/dev/loop0
root@jason-Virtual-Machine:/home/jason#
```

Step 2: Mount the Image File onto the Unused Loopback Device

- Now, mount the APFS image file to the unused loopback device using the following command:

```
losetup -r /dev/loop[number]
[evidence.dd]
```



```
root@jason-Virtual-Machine:/home/jason# losetup -r /dev/loop0 apfs.dd
root@jason-Virtual-Machine:/home/jason#
```

Note: Here, /dev/loop17 is the unused loopback device, which might vary from one system to another

- This creates a mount point on the machine. You may either view the image contents through the mount point or further mount this loop device using mount or apfs-fuse

Scenario 1.6: Examining Mac APFS Image File (Cont'd)

Step 3: Mount the APFS

- Create a mount directory (named apfs) and mount the APFS on it using the following command:

  `mkdir /mnt/apfs`

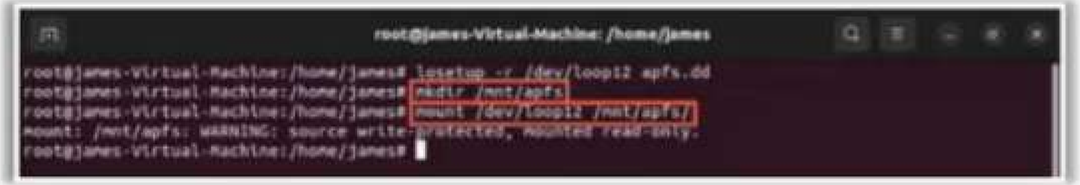
- Mount the APFS onto the loopback device by issuing the following command:

  `mount /dev/loop[number] /mnt/apfs`

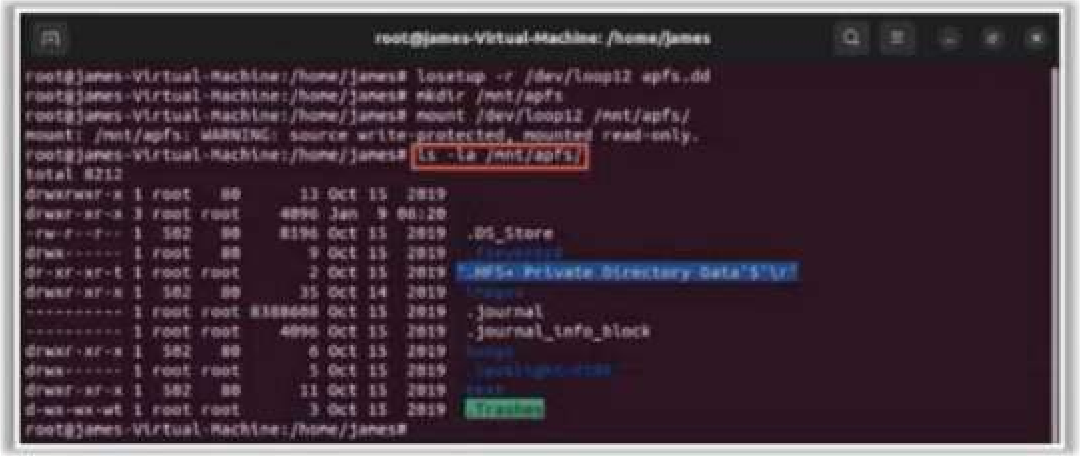
  (or)

  `apfs-fuse /dev/loop[number] /mnt/apfs`

- If no error is generated, the image file is considered successfully mounted
- Upon successfully mounting the file system, you can **view the contents of the image file** as shown in the screenshot



```
root@james-Virtual-Machine:/home/james# losetup -r /dev/loop12 apfs.dd
root@james-Virtual-Machine:/home/james# mkdir /mnt/apfs
root@james-Virtual-Machine:/home/james# mount /dev/loop12 /mnt/apfs/
mount: /mnt/apfs: WARNING: source write-protected, mounted read-only.
root@james-Virtual-Machine:/home/james# 
```

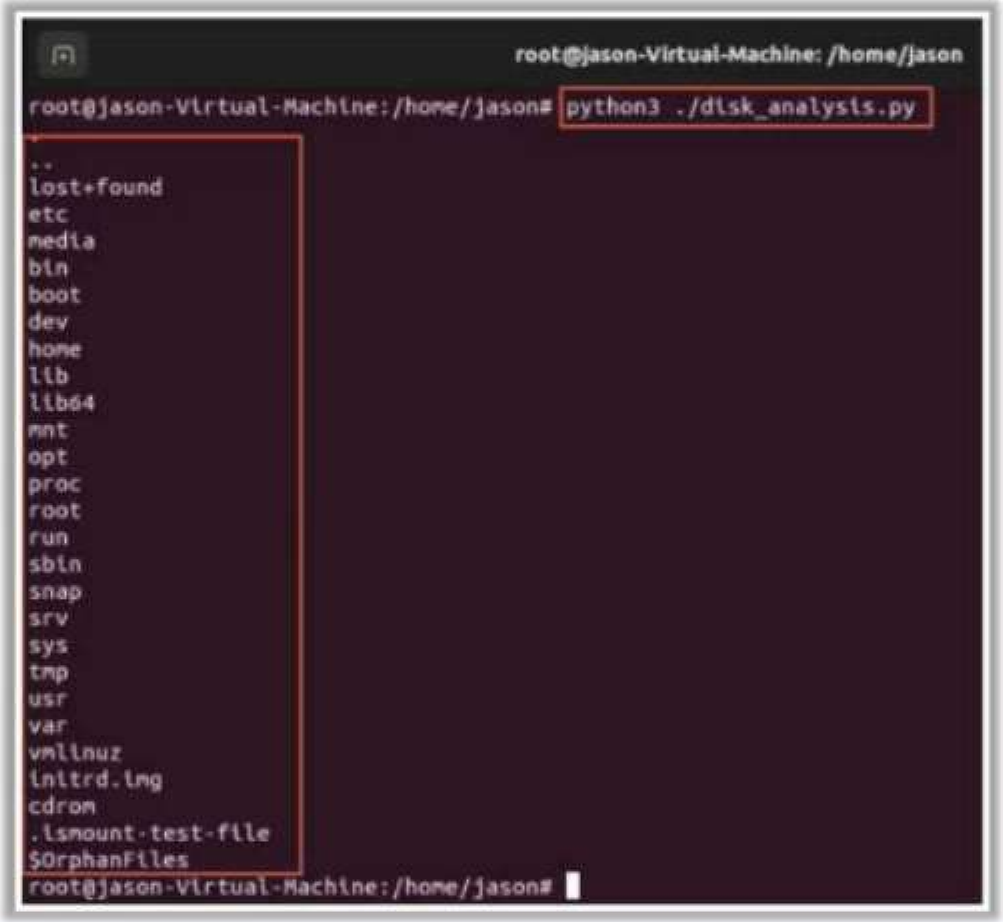


```
root@james-Virtual-Machine:/home/james# losetup -r /dev/loop12 apfs.dd
root@james-Virtual-Machine:/home/james# mkdir /mnt/apfs
root@james-Virtual-Machine:/home/james# mount /dev/loop12 /mnt/apfs/
mount: /mnt/apfs: WARNING: source write-protected, mounted read-only.
root@james-Virtual-Machine:/home/james# ls -la /mnt/apfs/
total 8212
drwxrwxr-x 1 root   80       13 Oct 15  2019 
drwxr-xr-x 3 root root     4096 Jan  9 06:20 
-rw-r--r-- 1  502   80     8196 Oct 15  2019  .DS_Store
drwx------ 1 root   80        9 Oct 15  2019  .fseventsd
dr-xr-xr-t 1 root root        2 Oct 15  2019 '.HFS+ Private Directory Data'$'\r'
drwxr-xr-x 1  502   80       35 Oct 14  2019  [illegible]
---------- 1 root root  8388608 Oct 15  2019  .journal
---------- 1 root root     4096 Oct 15  2019  .journal_info_block
drwxr-xr-x 1  502   80        6 Oct 15  2019  [illegible]
drwx------ 1 root root        5 Oct 15  2019  .Spotlight-V100
drwxr-xr-x 1  502   80       11 Oct 15  2019  [illegible]
d-wx-wx-wt 1 root root        3 Oct 15  2019  .Trashes
root@james-Virtual-Machine:/home/james# 
```

Scenario 1.7: Examining Disk Image Using PyTSK

- PyTSK is a Python wrapper for SleuthKit that provides a **Python-based interface** to access the libraries of SleuthKit using various Python scripts for performing different analysis tasks during an investigation

- You can use **disk_analysis.py**, a Python script available in Module 06 of your CHFIv11 Student Resource Kit to access a disk image and view the associated files and directories



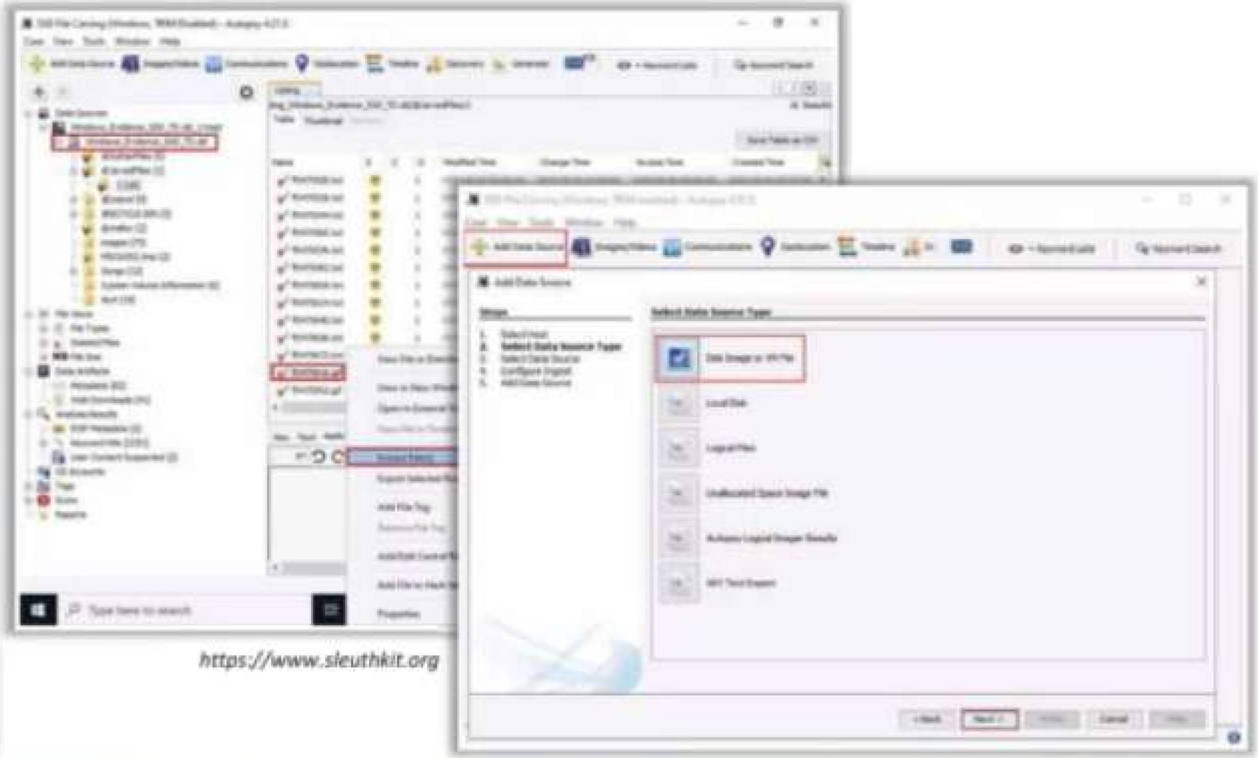
```
root@jason-Virtual-Machine:/home/jason# python3 ./disk_analysis.py
..
lost+found
etc
media
bin
boot
dev
home
lib
lib64
mnt
opt
proc
root
run
sbin
snap
srv
sys
tmp
usr
var
vmlinuz
initrd.img
cdrom
.lsmount-test-file
$OrphanFiles
root@jason-Virtual-Machine:/home/jason# 
```

Scenario 2: Examining Images on Windows Forensic Workstation

- Tools such as **Autopsy**, **FTK Imager**, and **Volatility** enable investigators to **examine image files** on Windows forensic workstations and identify other files and folders located on them

Steps to View an Image File Using the Autopsy Tool

- Click on the "**Add Data Source**" option; then, select the required type of data source to add and click on "**Next**"
- Provide the path of data source to be examined
- Select the required modules as per the investigation and click on "**Next**"
- After module selection, click on the "**Finish**" button



https://www.sleuthkit.org

Scenario 2: Examining Images on Windows Forensic Workstation (Cont'd)

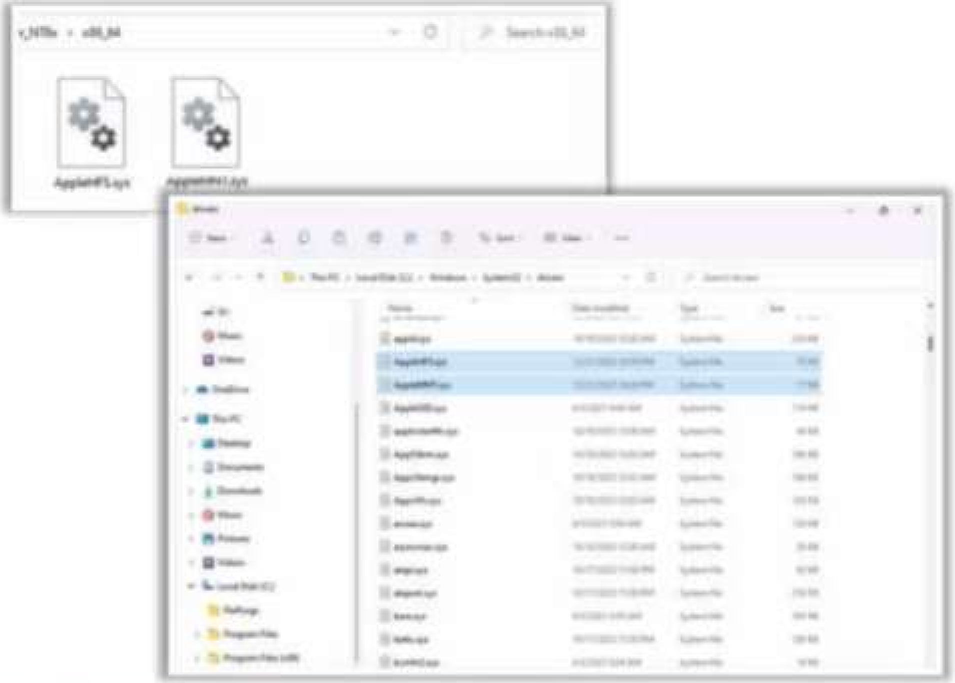
Examining Mac HFS+ Image File

Method 1: Using Apple HFS+ Drivers

Investigators can read Mac HFS+ through the Windows built-in File Explorer utility by installing **Apple HFS+ Drivers** on a Windows workstation that can provide insights about **stored caches**, **saved files**, etc. during an investigation

Steps to Read Mac HFS+ Using Apple HFS+ Driver

- **Step 1**: Install and extract the **Apple HFS+ Windows driver package** zip file on the Windows workstation
- **Step 2**: From the extracted folder, copy **AppleHFS.sys** and **AppleMNT.sys** files to **C:Windows/System32/drivers/**
- **Step 3**: Double-click on **Add AppleHFS.reg** in the extracted folder to merge the Apple HFS registry with the Windows registry. Next, Click on **Yes** and **OK** on the prompt window and restart the system

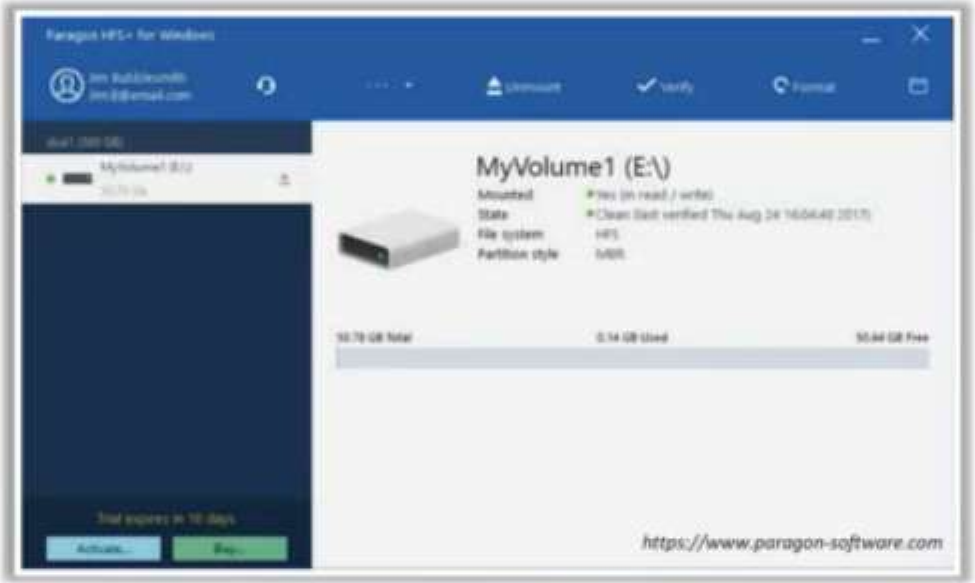


Scenario 2: Examining Images on Windows Forensic Workstation (Cont'd)

Examining Mac HFS+ Image File

Method 2: Using Paragon HFS+ for Windows

- Paragon HFS+ for Windows is a tool that enables investigators to read, write, and modify HFS+ formatted files on Windows, and can support HDD, SSD, and flash drives
- This tool helps to access HFS+ formatted drives collected from the evidence Mac system



MyVolume1 (E:\)

Mounted ● Yes (in read / write)
State ● Clean (last verified Thu Aug 24 16:04:40 2017)
File system HFS
Partition style MBR

50.78 GB Total 0.14 GB Used 50.64 GB Free

https://www.paragon-software.com

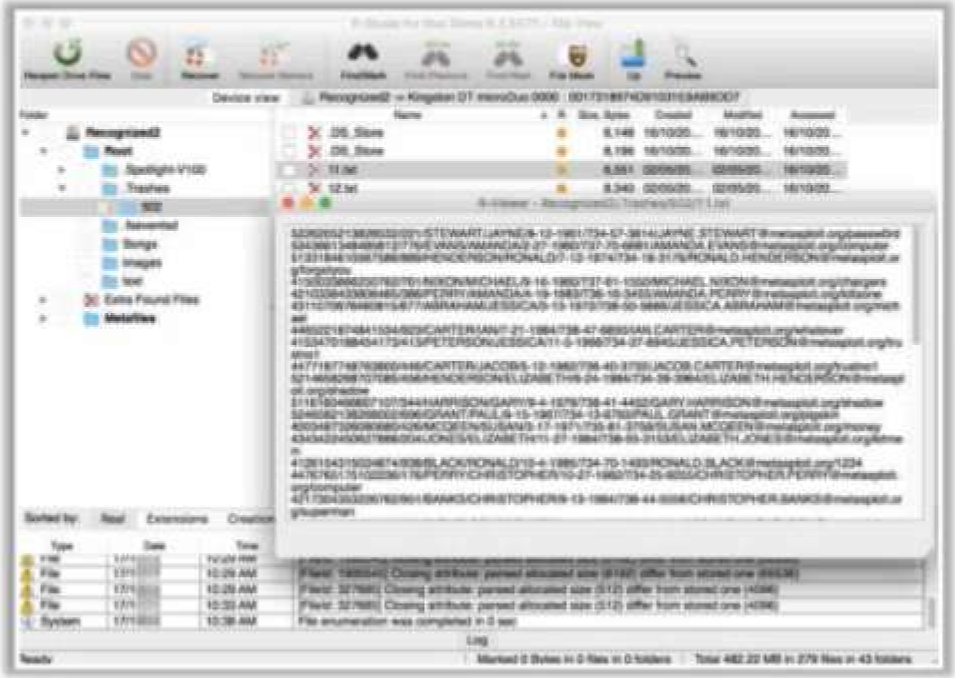
MyVolume1 (E:) 50.78 GB Total
0.14 GB Used 50.64 GB Free
Verified: ●Clean File system: HFS

Scenario 3: Examining Images on Mac Forensic Workstation

- Tools such as **R-Studio**, **Digital Collector**, and **OSForensics** can help forensic investigators to examine image files on macOS forensic workstations and identify files and folders located in them

R-Studio

- R-Studio is a data recovery solution for recovering files from **HFS/HFS+** and **APFS** (macOS) partitions along with Windows and Linux
- Investigators can use R-Studio to recover, view, and examine data from an image file



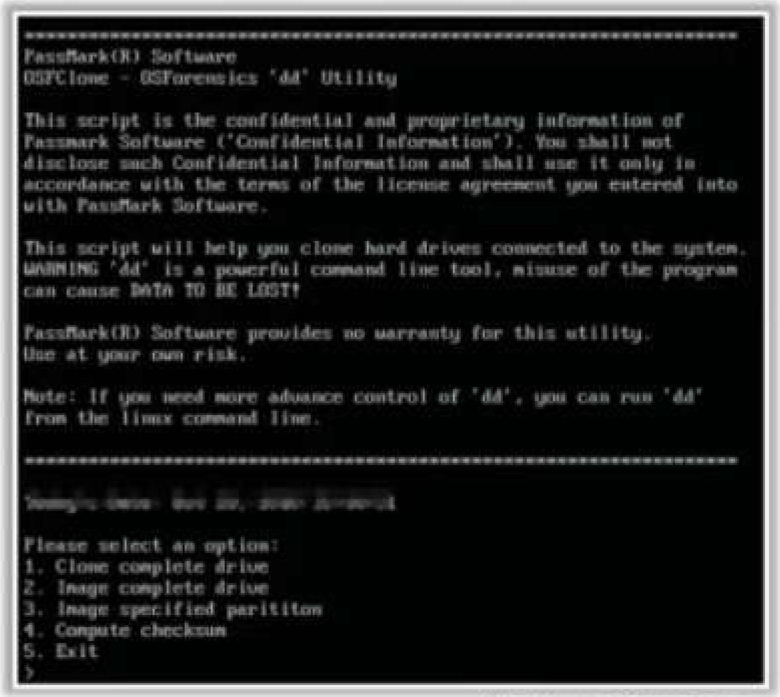
```
File    10:29 AM   [FileId: 1665044] Closing attribute: parsed allocated size (8192) differ from stored one (65536)
File    10:29 AM   [FileId: 327665] Closing attribute: parsed allocated size (512) differ from stored one (4096)
File    10:33 AM   [FileId: 327665] Closing attribute: parsed allocated size (512) differ from stored one (4096)
System  10:38 AM   File enumeration was completed in 0 sec
```

Ready | Marked 0 Bytes in 0 files in 0 folders | Total 482.22 MB in 279 files in 43 folders

https://www.r-studio.com

Digital Forensic Imaging Tools

OSFClone | OSFClone is a self-booting solution that allows investigators to **create** or **clone** exact **raw disk images** rapidly and independently of the installed OS



```
**************************************************************************
PassMark(R) Software
OSFClone - OSForensics 'dd' Utility

This script is the confidential and proprietary information of
Passmark Software ('Confidential Information'). You shall not
disclose such Confidential Information and shall use it only in
accordance with the terms of the license agreement you entered into
with PassMark Software.

This script will help you clone hard drives connected to the system.
WARNING 'dd' is a powerful command line tool, misuse of the program
can cause DATA TO BE LOST!

PassMark(R) Software provides no warranty for this utility.
Use at your own risk.

Note: If you need more advance control of 'dd', you can run 'dd'
from the linux command line.

**************************************************************************

[redacted]

Please select an option:
1. Clone complete drive
2. Image complete drive
3. Image specified peritition
4. Compute checksum
5. Exit
>
```

https://www.osforensics.com

Talon® Ultimate
https://www.logicube.com

Falcon®-NEO
https://www.logicube.com

PALADIN
https://sumuri.com

OpenText EnCase Forensic
https://www.opentext.com

Belkasoft X
https://belkasoft.com

Module Summary

- □ In this module, we discussed various data acquisition methods, types, and formats
- □ This module explained the acquisition of volatile and non-volatile information from different OSes
- □ It also elaborated on the various steps involved in the data acquisition methodology
- □ Furthermore, this module discussed various possible scenarios that one might encounter while preparing an acquired image file for forensic examination
- □ Finally, this module concluded with an illustration on various digital forensic imaging tools
- □ In the next module, we will discuss how to defeat anti-forensic techniques in detail