
CSC153: Activity 3 - Linux Data Acquisition

Ryan Kozak



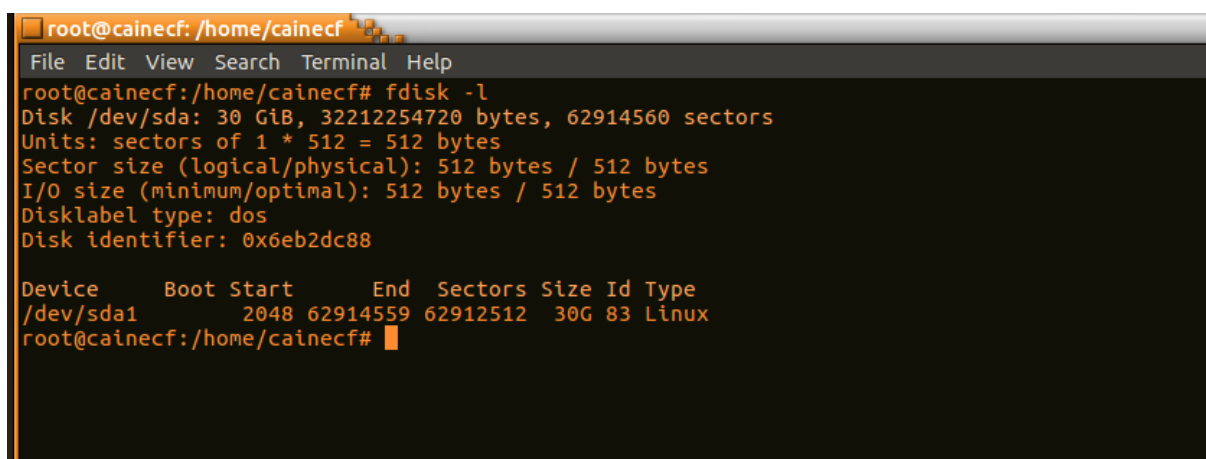
2019-09-22

Activity 3: Linux Data Acquisition

CSC 153 - Computer Forensics Principles and Practice

Part 1: Preparing The Target Drive

First we open up the terminal and issue the `su` command to login as root. We then issue the `fdisk -l` command to show the current disks.

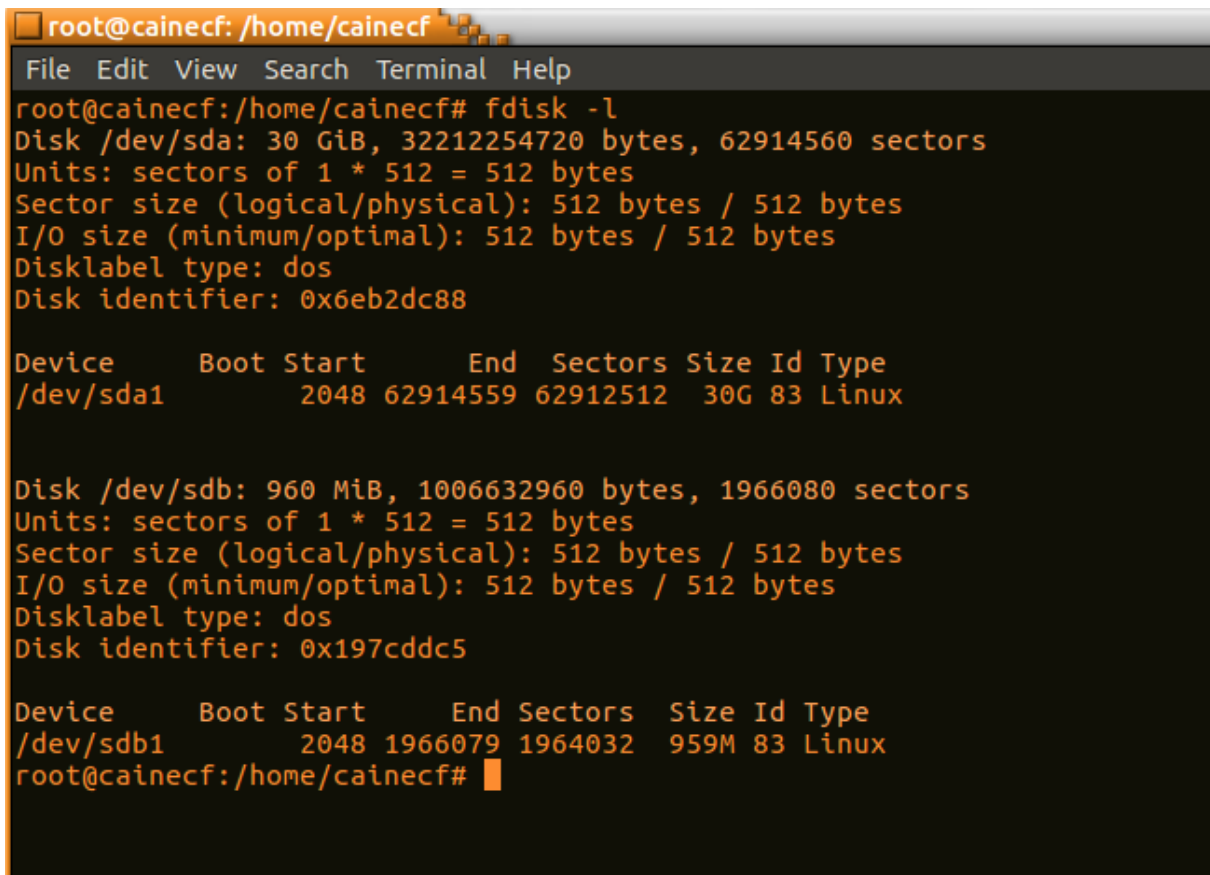


```
root@cainecf: /home/cainecf# fdisk -l
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6eb2dc88

Device      Boot Start        End    Sectors  Size Id Type
/dev/sda1   2048 62914559 62912512   30G 83 Linux
root@cainecf: /home/cainecf#
```

Figure 1: Current disks, no flash drives plugged in.

Now we plug the target USB drive into the system and issue `fdisk -l` once more. This time `/dev/sdb` appears, which is our target drive.

A terminal window titled 'root@cainecf: /home/cainecf' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the output of the 'fdisk -l' command. It lists details for two disks: /dev/sda (30 GiB) and /dev/sdb (960 MiB). For each disk, it shows units, sector size, I/O size, disklabel type, and disk identifier. Below the disk details, it shows a table of partitions for each disk. For /dev/sda, there is one partition: /dev/sda1, starting at sector 2048 and ending at 62914559, with a size of 30G, type 83 (Linux). For /dev/sdb, there is one partition: /dev/sdb1, starting at sector 2048 and ending at 1966079, with a size of 959M, type 83 (Linux). The prompt 'root@cainecf: /home/cainecf#' is shown at the bottom.

```
root@cainecf: /home/cainecf# fdisk -l
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 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: 0x6eb2dc88

Device      Boot Start        End  Sectors  Size Id Type
/dev/sda1                2048 62914559 62912512   30G 83 Linux

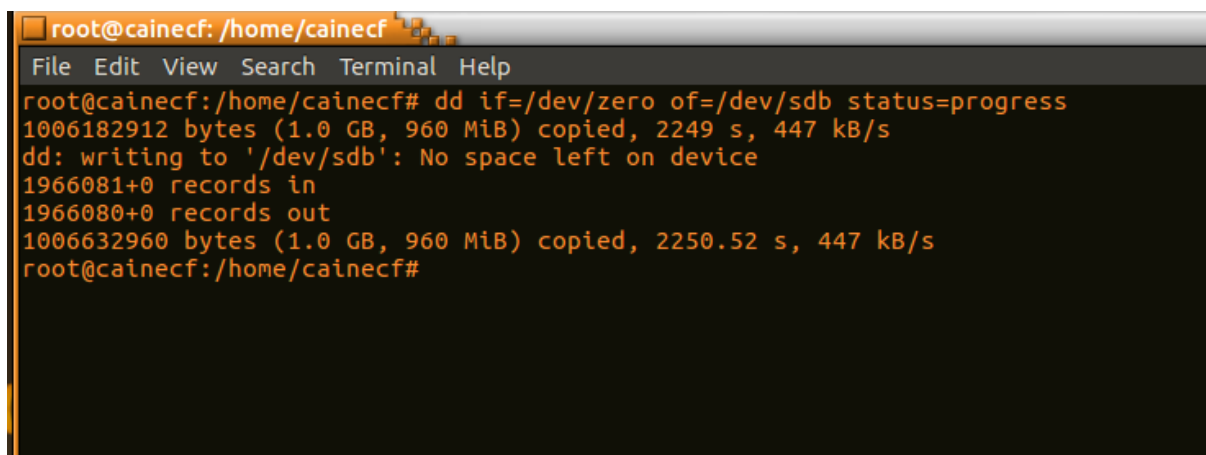
Disk /dev/sdb: 960 MiB, 1006632960 bytes, 1966080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x197cddc5

Device      Boot Start        End  Sectors  Size Id Type
/dev/sdb1                2048 1966079 1964032   959M 83 Linux
root@cainecf: /home/cainecf#
```

Figure 2: Current disks, with target drive plugged in.

It is now time we zero out the target drive to ensure that absolutely no data is on it when we use it to make a copy of our evidence drive. The target drive is zeroed out via `dd if=/dev/zero of=/dev/sdb`.

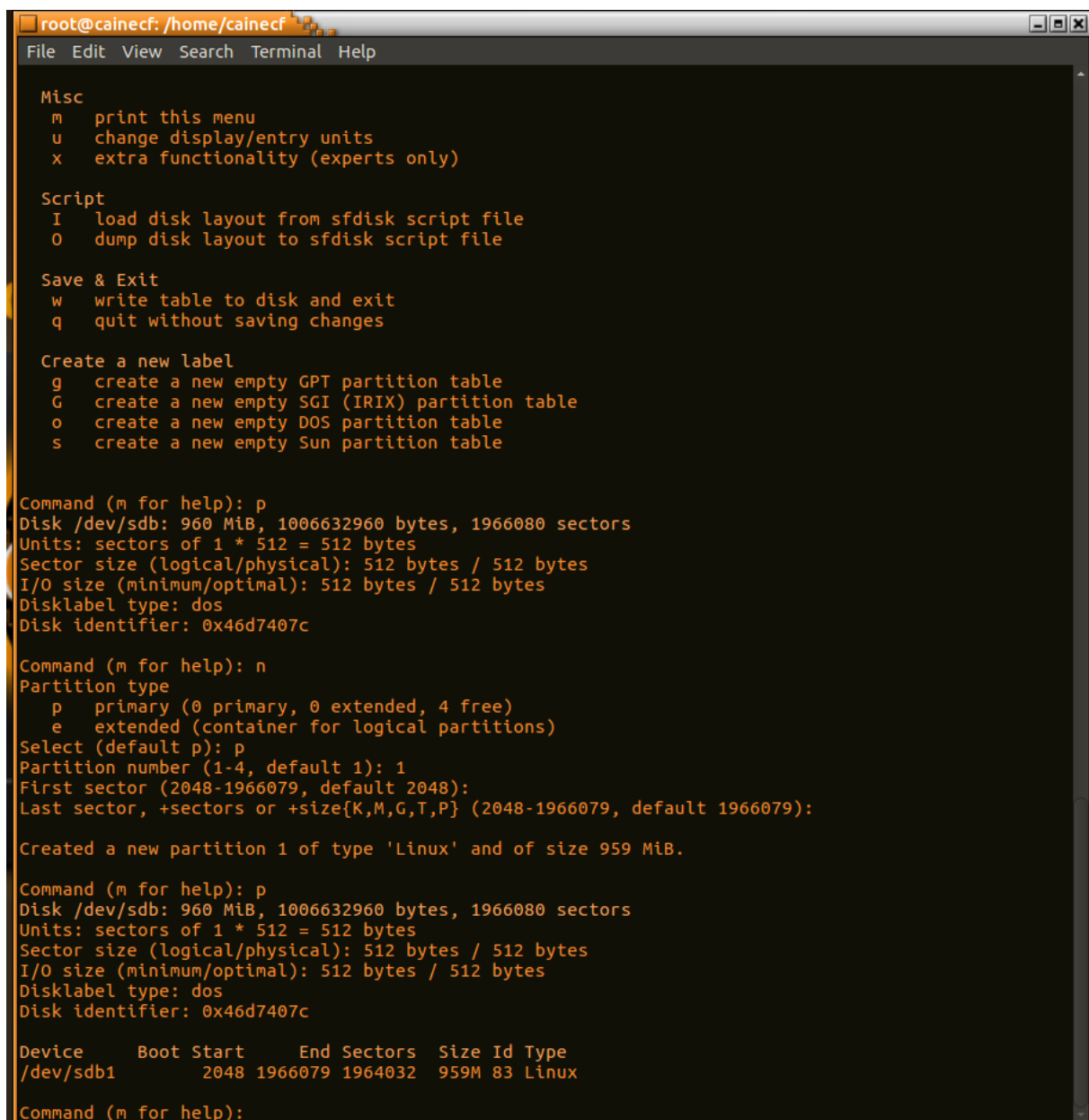
Note: Because it was taking so long to zero out a drive of only 1Gb, I decided to add the `status=progress` option to the command. Knowing the progress prevented me from thinking things were hanging.



```
root@cainecf: /home/cainecf
File Edit View Search Terminal Help
root@cainecf: /home/cainecf# dd if=/dev/zero of=/dev/sdb status=progress
1006182912 bytes (1.0 GB, 960 MiB) copied, 2249 s, 447 kB/s
dd: writing to '/dev/sdb': No space left on device
1966081+0 records in
1966080+0 records out
1006632960 bytes (1.0 GB, 960 MiB) copied, 2250.52 s, 447 kB/s
root@cainecf: /home/cainecf#
```

Figure 3: Zeroing out target drive with dd.

We then create a new partition table on the target drive by issuing `fdisk /dev/sdb`, selecting `n` for new partition, and `p` for primary. This partition is to be the first partition on the drive, so 1 is entered.



```

root@cainecf: /home/cainecf
File Edit View Search Terminal Help

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

Script
I  load disk layout from sfdisk script file
O  dump disk layout to sfdisk script file

Save & Exit
w  write table to disk and exit
q  quit without saving changes

Create a new label
g  create a new empty GPT partition table
G  create a new empty SGI (IRIX) partition table
o  create a new empty DOS partition table
s  create a new empty Sun partition table

Command (m for help): p
Disk /dev/sdb: 960 MiB, 1006632960 bytes, 1966080 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: 0x46d7407c

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1966079, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-1966079, default 1966079):

Created a new partition 1 of type 'Linux' and of size 959 MiB.

Command (m for help): p
Disk /dev/sdb: 960 MiB, 1006632960 bytes, 1966080 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: 0x46d7407c

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1             2048 1966079 1964032  959M 83 Linux

Command (m for help):

```

Figure 4: Creating new partition on target drive.

The next step is changing the partition to Windows 95 FAT32. To do so we navigate to the menu, select **t** to change the partition type, and view the available file systems via **l**. We'll select **c** for Windows 95 FAT32(LBA). Changes are written to the drive via **w**.

```

root@cainecf: /home/cainecf
File Edit View Search Terminal Help
s create a new empty Sun partition table

Command (m for help): t
Selected partition 1
Partition type (type L to list all types): l

 0 Empty                24 NEC DOS                81 Minix / old Lin  bf Solaris
 1 FAT12                 27 Hidden NTFS Win  82 Linux swap / So c1 DRDOS/sec (FAT-
 2 XENIX root            39 Plan 9             83 Linux            c4 DRDOS/sec (FAT-
 3 XENIX usr             3c PartitionMagic    84 OS/2 hidden or  c6 DRDOS/sec (FAT-
 4 FAT16 <32M            40 Venix 80286        85 Linux extended  c7 Syrix
 5 Extended              41 PPC PReP Boot     86 NTFS volume set da Non-FS data
 6 FAT16                 42 SFS                87 NTFS volume set db CP/M / CTOS / .
 7 HPFS/NTFS/exFAT       4d QNX4.x             88 Linux plaintext de Dell Utility
 8 AIX                   4e QNX4.x 2nd part   8e Linux LVM        df BootIt
 9 AIX bootable          4f QNX4.x 3rd part   93 Amoebe           e1 DOS access
a OS/2 Boot Manag       50 OnTrack DM        94 Amoebe BBT       e3 DOS R/O
b W95 FAT32             51 OnTrack DM6 Aux  9f BSD/OS          e4 SpeedStor
c W95 FAT32 (LBA)       52 CP/M              a0 IBM Thinkpad hi ea Rufus alignment
e W95 FAT16 (LBA)       53 OnTrack DM6 Aux  a5 FreeBSD         eb BeOS fs
f W95 Ext'd (LBA)       54 OnTrackDM6        a6 OpenBSD         ee GPT
10 OPUS                 55 EZ-Drive          a7 NeXTSTEP        ef EFI (FAT-12/16/
11 Hidden FAT12         56 Golden Bow        a8 Darwin UFS       f0 Linux/PA-RISC b
12 Compaq diagnost      5c Priam Edisk       a9 NetBSD           f1 SpeedStor
14 Hidden FAT16 <3      61 SpeedStor         ab Darwin boot     f4 SpeedStor
16 Hidden FAT16         63 GNU HURD or Sys  af HFS / HFS+      f2 DOS secondary
17 Hidden HPFS/NTF      64 Novell Netware    b7 BSDI fs         fb VMware VMFS
18 AST SmartSleep       65 Novell Netware    b8 BSDI swap        fc VMware VMKCORE
1b Hidden W95 FAT3      70 DiskSecure Mult  bb Boot Wizard hid fd Linux RAID auto
1c Hidden W95 FAT3      75 PC/IX             bc Acronis FAT32 L fe LANstep
1e Hidden W95 FAT1      80 Old Minix         be Solaris boot    ff BBT

Partition type (type L to list all types): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help): p
Disk /dev/sdb: 960 MiB, 1006632960 bytes, 1966080 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: 0x46d7407c

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1                2048 1966079 1964032  959M  c W95 FAT32 (LBA)

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@cainecf: /home/cainecf#

```

Figure 5: Changing the partition to Windows 95 FAT32.

Lastly, we format a FAT file system from Linux by issuing `mkfs.msdos -vF32 /dev/sdb1`.

```
Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1   2048 1966079 1964032  959M  c W95 FAT32 (LBA)
root@cainecf:/home/cainecf# mkfs.msdos -vF32 /dev/sdb1
mkfs.fat 3.0.28 (2015-05-16)
/dev/sdb1 has 31 heads and 62 sectors per track,
hidden sectors 0x0800;
logical sector size is 512,
using 0xf8 media descriptor, with 1964032 sectors;
drive number 0x80;
filesystem has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 1915 sectors, and provides 245021 clusters.
There are 32 reserved sectors.
Volume ID is 757298a4, no volume label.
root@cainecf:/home/cainecf#
```

Figure 6: Formatting a FAT file system.

Part 2: Perform Data Acquisition

Now we plug our evidence drive into the system, and issue `fdisk -l` to determine where that is at as well.

```
root@cainecf:/home/cainecf# fdisk -l
Disk /dev/sda: 30 GiB, 32212254720 bytes, 62914560 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: 0x6eb2dc88

Device      Boot Start      End Sectors  Size Id Type
/dev/sda1   2048 62914559 62912512  30G 83 Linux

Disk /dev/sdb: 960 MiB, 1006632960 bytes, 1966080 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: 0x46d7407c

Device      Boot Start      End Sectors  Size Id Type
/dev/sdb1   2048 1966079 1964032  959M  c W95 FAT32 (LBA)

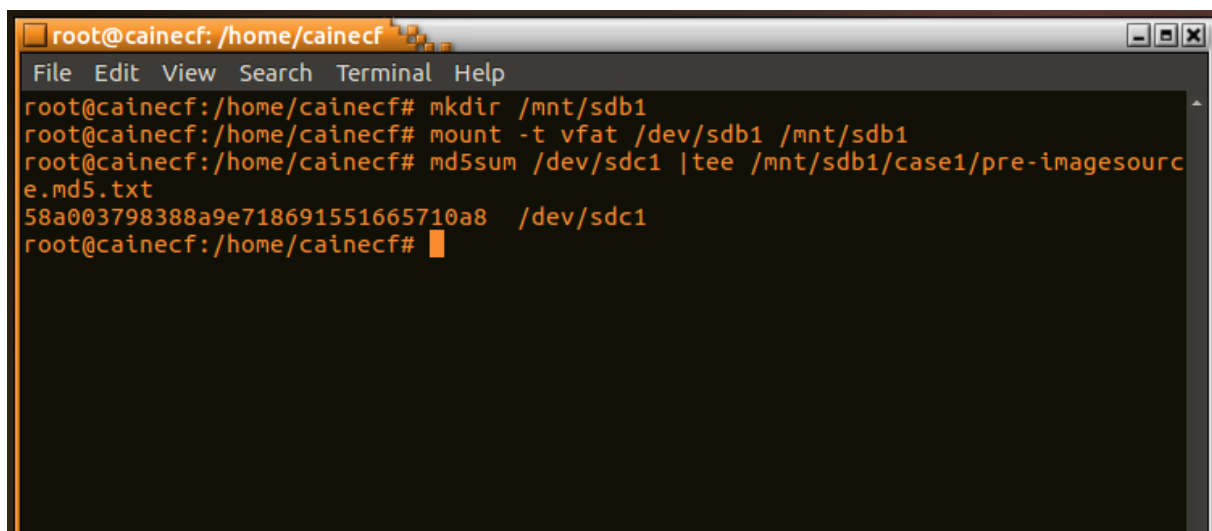
Disk /dev/sdc: 960 MiB, 1006632960 bytes, 1966080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x9b624fcc

Device      Boot Start      End Sectors  Size Id Type
/dev/sdc1   2048 1966079 1964032  959M  b W95 FAT32
root@cainecf:/home/cainecf#
```

Figure 7: Evidence drive is `/dev/sdc1` in this case.

The next step is to mount our target drive by creating a directory `/mnt/sdb1` and issuing the command `mount -t vfat /dev/sdb1 /mnt/sdb1`. We then create a directory `case1` and calculate the md5sum of the evidence drive, saving it into this new directory. The hash is calculated via `md5sum /dev/sdc1 | tee /mnt/sdb1/case1/pre-imagesource.md5.txt`.

We're ready to acquire data from the evidence drive. We do so via `dcfldd if=/dev/sdc1 of=/mnt/sdb1/case1/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdb1/case1/post-imagesource.md5.txt`.

A screenshot of a terminal window titled 'root@cainecf: /home/cainecf'. The terminal shows the following commands and output:

```
root@cainecf:/home/cainecf# mkdir /mnt/sdb1
root@cainecf:/home/cainecf# mount -t vfat /dev/sdb1 /mnt/sdb1
root@cainecf:/home/cainecf# md5sum /dev/sdc1 | tee /mnt/sdb1/case1/pre-imagesource.md5.txt
58a003798388a9e718691551665710a8 /dev/sdc1
root@cainecf:/home/cainecf#
```

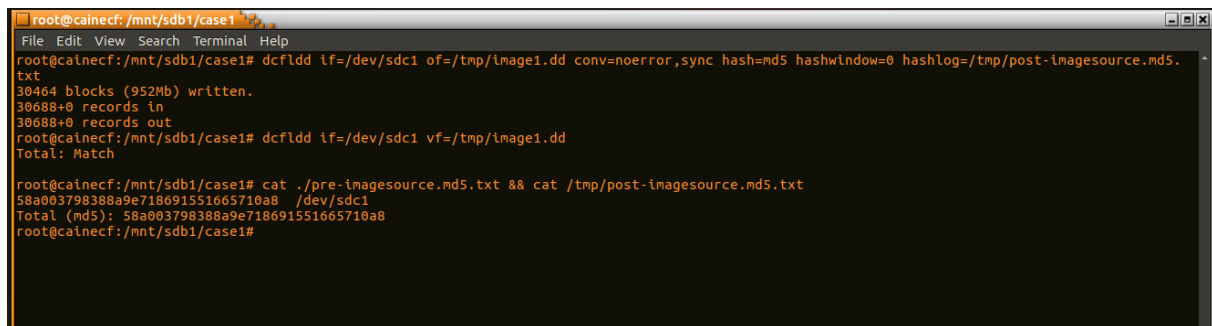
Figure 8: Verification of acquired data.

Note: The flash drive on which I setup to copy the evidence was the exact same size as the evidence drive. This created an issue for me, as there wasn't enough space. I used the `/tmp` folder to save `image1.dd` and its hash. This went fine. Next time I will bring a larger drive to copy my evidence.

See Figure 8 below for console output.

Part 3: Validate The Acquired Data

Now it's time to validate our acquired data. We can do this via `dcfldd` via the command `dcfldd if=/dev/sdc1 split=2M of=/tmp/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/tmp/post-imagesource.md5.txt`. We can also verify it using the md5 sums we've generated. Each method is used in figure 8 below.



```
root@cainecf: /mnt/sdb1/case1
File Edit View Search Terminal Help
root@cainecf: /mnt/sdb1/case1# dd if=/dev/sdc1 of=/tmp/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/tmp/post-imagesource.md5.txt
30464 blocks (952Mb) written.
30688+0 records in
30688+0 records out
root@cainecf: /mnt/sdb1/case1# dd if=/dev/sdc1 vf=/tmp/image1.dd
Total: Match

root@cainecf: /mnt/sdb1/case1# cat ./pre-imagesource.md5.txt && cat /tmp/post-imagesource.md5.txt
58a003798388a9e718691551665710a8 /dev/sdc1
Total (md5): 58a003798388a9e718691551665710a8
root@cainecf: /mnt/sdb1/case1#
```

Figure 9: Verification of acquired data.

Post-Activity Questions

1. What are the two broad categories of acquisition?
 - Static Acquisition.
 - Live Acquisition.
2. What is a live storage acquisition and when is it used?
 - Data is collected from the local computer or over a network while running. Not repeatable because data continually being altered by the OS.
 - Used when a computer cannot be shut down.
3. Which command should be used to check the disks available on the current system? You only need to state the command name, not the entire command string.
 - fdisk is used, `fdisk -l`.
4. The `mkfs -t` command does what?
 - Makes a file system of a certain type.
5. Which drive should be “zeroed out”, the source evidence drive or the target drive?
 - The target drive.
6. What is the purpose of “zeroing out” before a storage acquisition is performed?
 - To ensure there is actually absolutely nothing on the drive. Such as software/malware from the vendor that may effect evidence.
7. When you issue the command the command `dd if=/dev/zero of=/dev/sdb`, What does the string `/dev/sdb` represent?
 - This command zeroes out the target drive, before we copy evidence to it. So, `/dev/sdb` represents the target drive.

8. The `md5sum /dev/sda` command does what? Why is it used?
 - This command would generate a hash of the drive on which CAINE is installed. I think this question intended to say `/dev/sdc`? We do this to create a hash of the evidence before we copy it, to compare with the hash of our copy to validate that they're the same.
9. How many times should the `md5sum` command be used at least in one acquisition?
 - Once for the pre-image source when we hash the evidence drive.
 - Once for the post-image source when we hash our image after acquisition.
10. Instead of using “`dd`”, what other commands can you use to perform data acquisition in Linux?
 - You can use `dcfldd`, if it's installed, which is the DoD's enhanced version of `dd`.