

## Lab 2: Data Acquisition and Basic Forensic Analysis Objectives

### Objectives

- Perform Data Acquisition using FTK imager and Linux dd, dcfldd commands
- Learn to use Windows version Autopsy
- Locate deleted/hidden files
- Perform a dirty word search
- Create a case report with any evidence you find
- Understand the difference between disk formatting in Windows and zero-out Tasks Task 1. Software Preparation.

### Tasks

- Task 1: Software Preparation.
- Task 2: Prepare Suspect Drive.
- Task 3: Perform a data acquisition with FTK Imager.
- Task 4: Perform a data acquisition with Linux dd/dcfldd command.
- Task 5: Analyze the acquired data.
- Task 6: Task 6. Perform a dirty word search.
- Task 7: Zero-out the suspect USB drive.
- Task 8: Repeat Task 4-6.
- Task 9: Answer the questions.

### Task 1: Software Preparation.

The first task requires us to download and install the following forensics tools: FTK Imager and Autopsy, which are to be downloaded on to our windows machine, and CAINE being already installed in our Virtual Machine which follows the similar procedures found in our previous labs.

Installation of FTK Imager version 4.2.0:

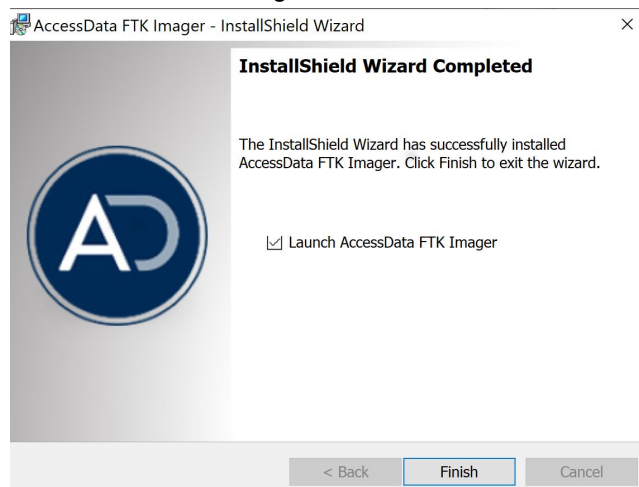
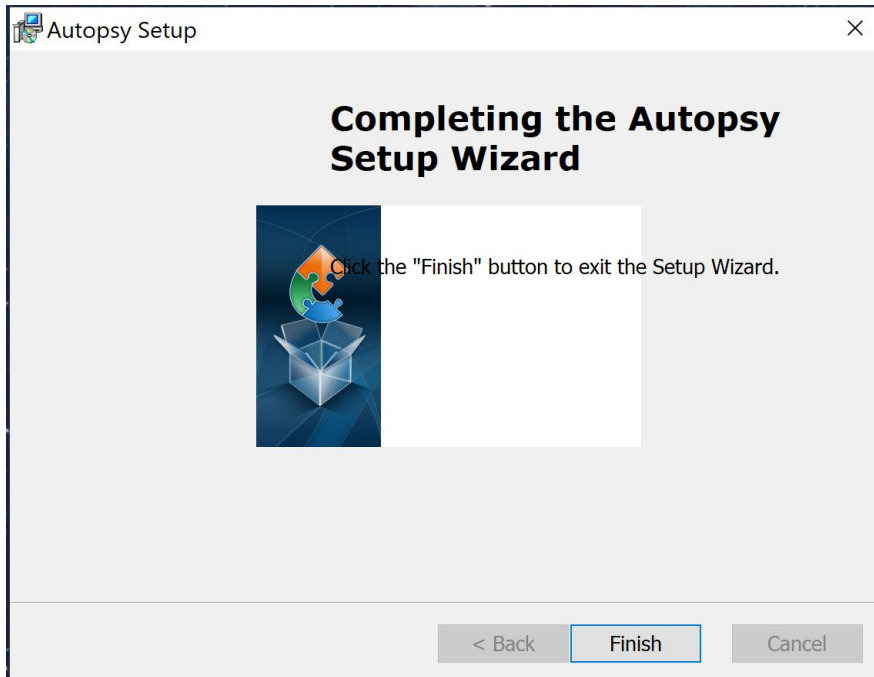


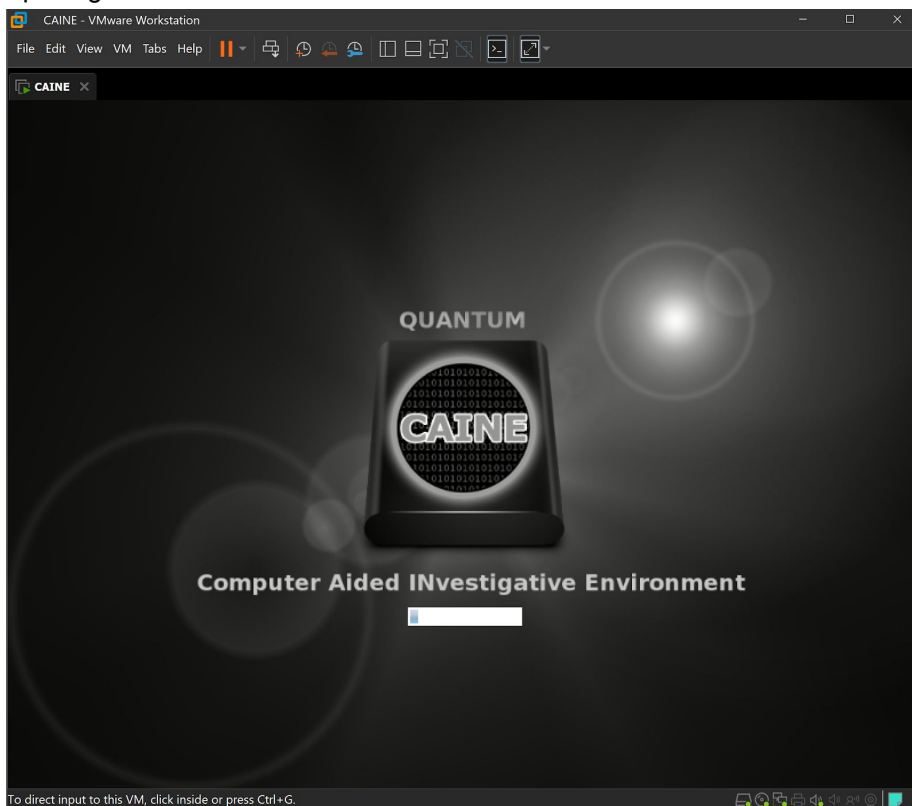
Figure 1: Installation of FTK Imager.

Installation of Sleuth Kit Autopsy version 4.16.0 windows:



**Figure 2: Installation of Autopsy.**

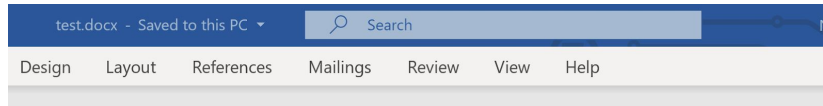
Opening CAINE on our Virtual Machine:



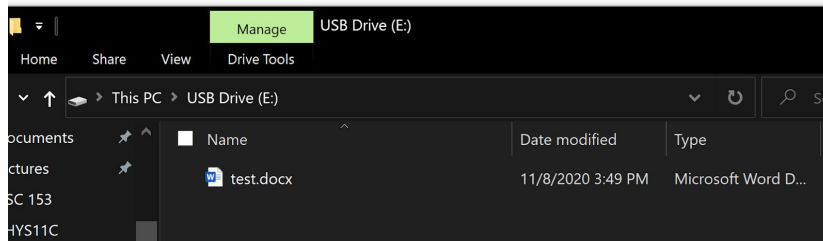
**Figure 3: Prepare Virtual Machine CAINE**

## Task 2: Prepare Suspect Drive.

The next task specifies that we need a 500mb-2gb USB. In this case, I only have a 2GB USB available so I am using it for this lab. I have zeroed out this drive before and what I need to do is create a test.doc file in the USB. In the document, I need to include the following text “Life does not provide Warranties and Guarantees it only provides possibilities and opportunities for those who there to make the best use of it!”

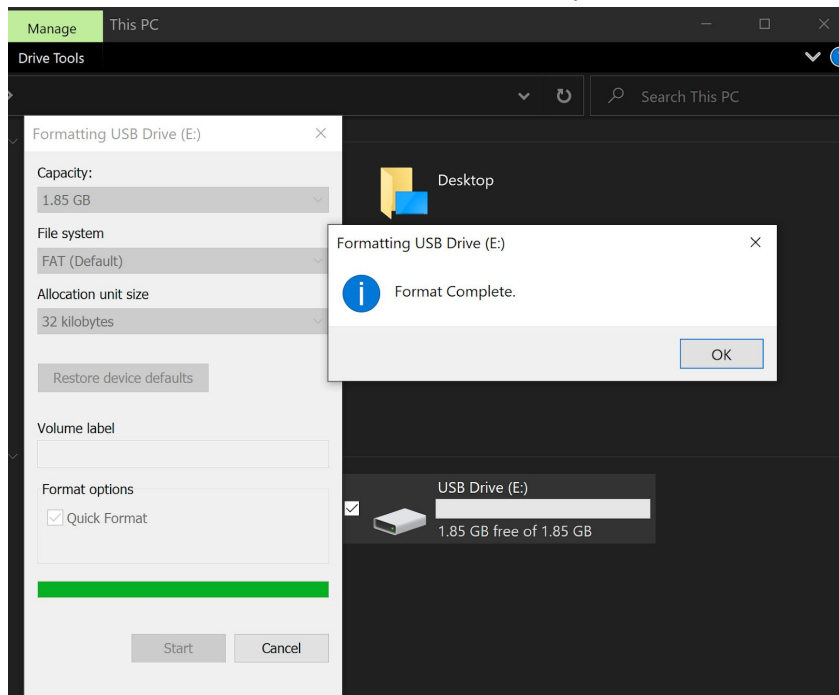


Life does not provide Warranties and Guarantees it only provides possibilities and opportunities for those who there to make best use of it!



**Figure 4: Saving test.docx onto the suspect drive.**

After the file is saved onto the drive, we need to perform a disk format towards the drive



**Figure 5: Formatting the drive was successful.**

### Task 3: Perform a data acquisition with FTK Imager.

For task 3, we start by opening FTK Imager and follow this “File->Create Disk Image.” Then in the select source dialog box, we need the click on the “Physical Drive” option before clicking next. That will take you the “Source Drive selection”, once here select the suspect drive that is still connected to windows before clicking “Finish”.

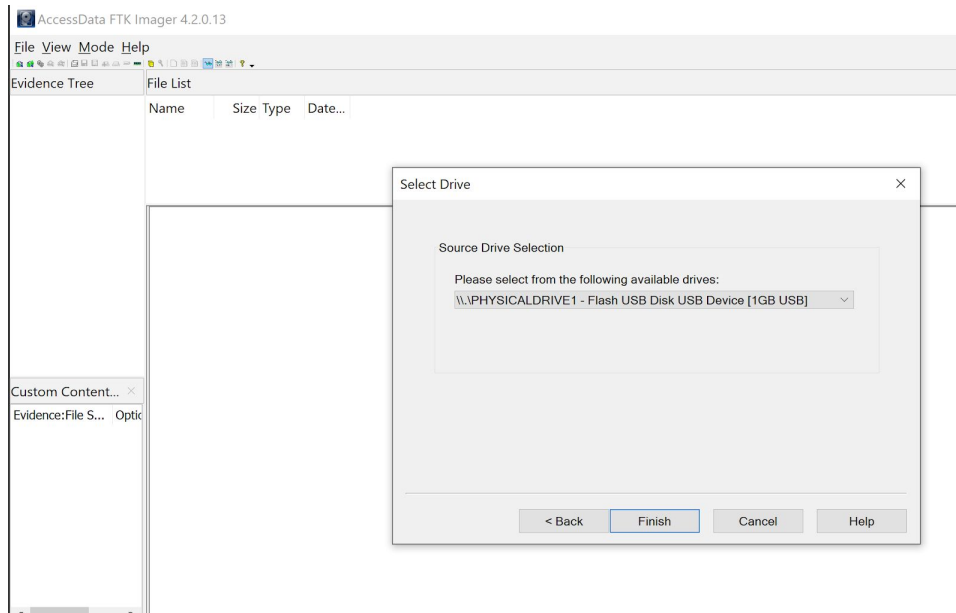


Figure 6: Selecting our Suspect Drive as the Source Drive Selection.

After clicking the “Finish”, it will take you to the “Create Image Dialog Box”, here we are to check the “Verify images after they are created” box. Then click on “Add..”, Then select “Raw” as the Destination Image Type. Then complete the case information before clicking “Next”.

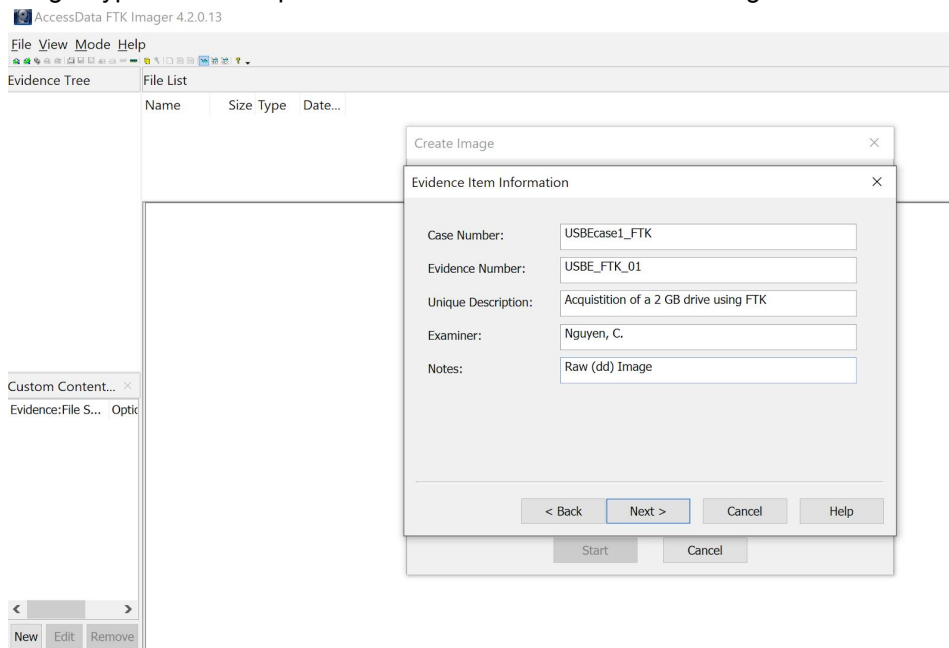
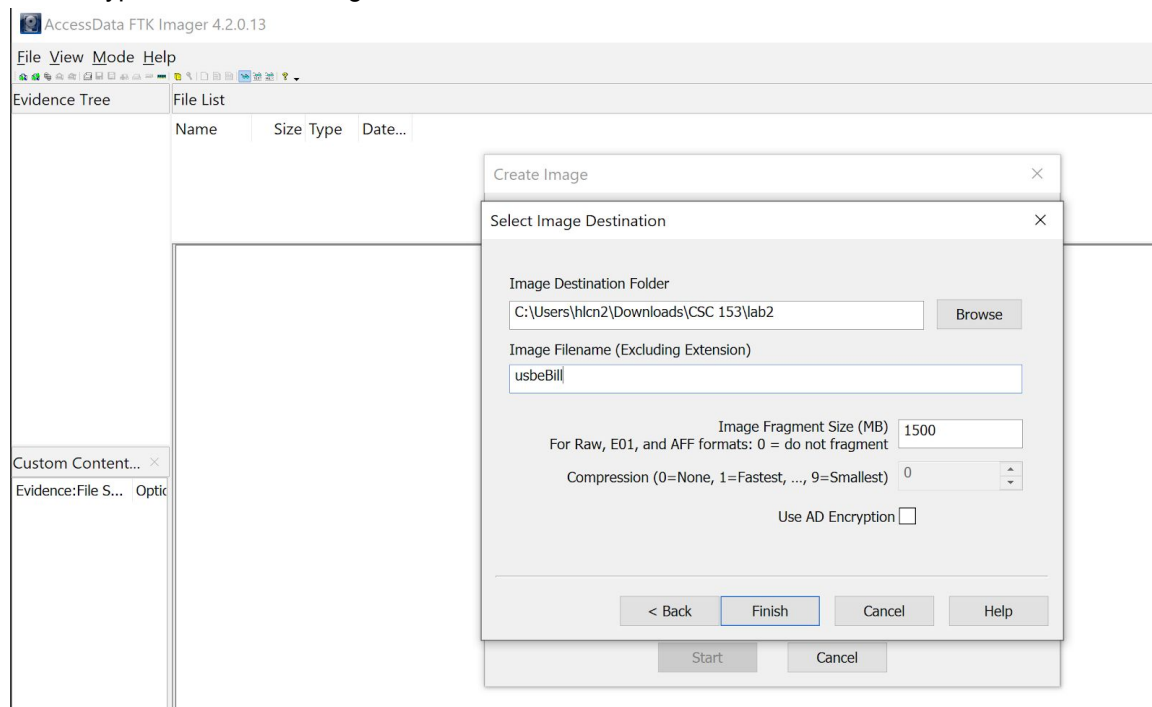


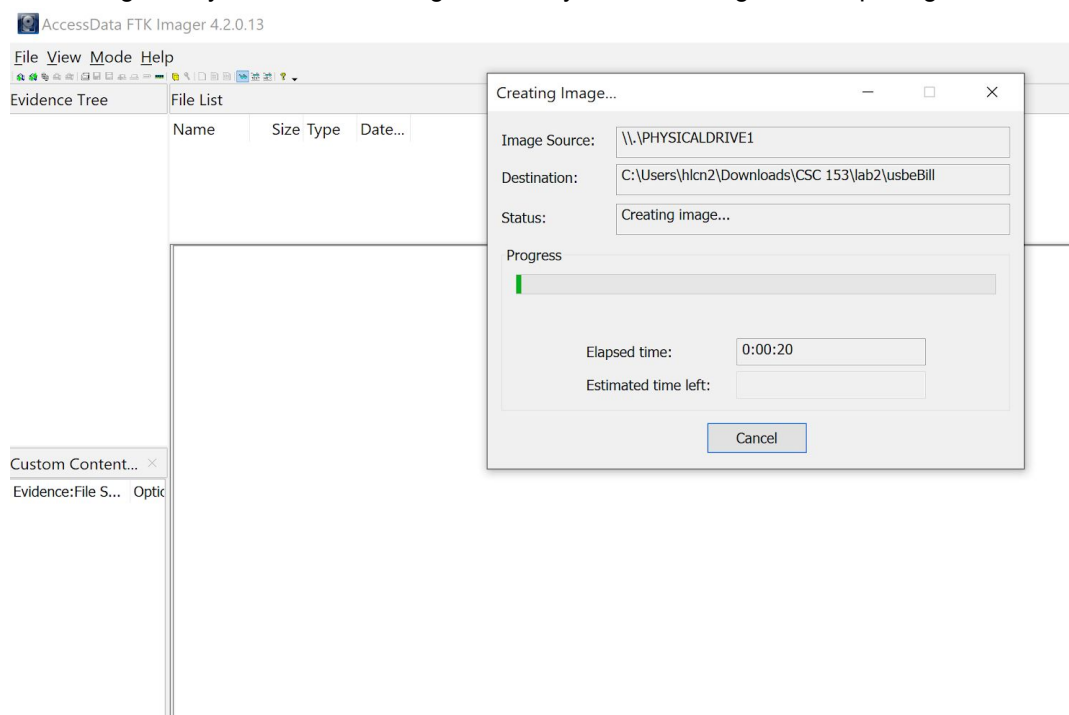
Figure 7: Filling in the Case Information: name-USBCase1\_FTK, evidence number-USBE\_FTK\_01.

After clicking “Next”, it will take you to the “Select Image Destination” dialogue box, click “Browse” and specify the location you want to store the image. Then fill in the Image Filename. Also, uncheck the “Use AD Encryption” before clicking on “Finish”.



**Figure 8: Selecting Image Destination.**

Once you’ve gotten this far, click “start” this will initiate the inquiry. Review the information in the Drive/Image verify results in the image summary before closing and completing this task.



**Figure 9: Process of creating the FTK Image.**

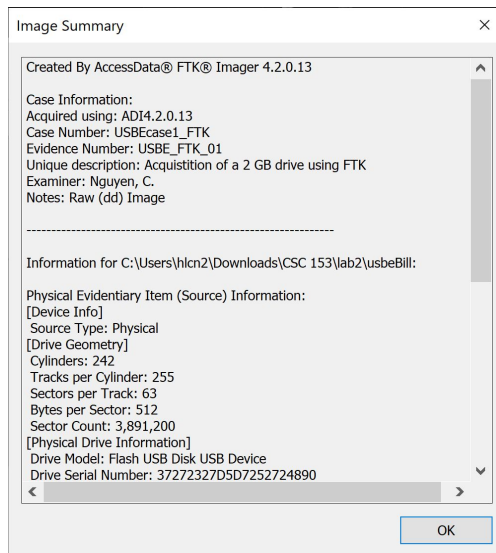


Figure 10: Image Summary after completion.

#### Task 4: Perform a data acquisition with Linux dd/dcfldd command.

This task asks us to follow the same instructions from activity 3 in order for us to perform data acquisition of the USB drive using the Linux commands.dd/dcfldd. First, we go onto CAINE and locate our drives. In this case, sdb1 is the suspect drive and sdc1 is our to target drive.

```
root@cainecf:/home/cainecf# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 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: 0x0c063046

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1                2048 41943039 41940992   20G 83 Linux

Disk /dev/sdb: 1.9 GiB, 1992294400 bytes, 3891200 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: 0x028e887c

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdb1   *          32 3891199 3891168   1.9G  e W95 FAT16 (LBA)

Disk /dev/sdc: 1.9 GiB, 1992294400 bytes, 3891200 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: 0x5a2f562e

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdc1                2048 3891199 3889152   1.9G  c W95 FAT32 (LBA)
root@cainecf:/home/cainecf#
```

Figure 11: /dev/sdb1 is our evidence drive, /dev/sdc1/ is our target drive.

Now we need to zero out our target drive to prevent any data corruption.

```
root@cainecf:/home/cainecf# dd if=/dev/zero of=/dev/sdc status=progress
1991873024 bytes (2.0 GB, 1.9 GiB) copied, 1974 s, 1.0 MB/s
dd: writing to '/dev/sdc': No space left on device
3891201+0 records in
3891200+0 records out
1992294400 bytes (2.0 GB, 1.9 GiB) copied, 1975.79 s, 1.0 MB/s
root@cainecf:/home/cainecf#
```

**Figure 12: Zero out the target drive before the acquisition.**

After that, we need to create a partition. To do this follow these commands: “fdisk /dev/sdc”, “p” to see if there is a partition, if there is none then we need to create another one using “n” then “p” for primary, then “1” for the first partition.

```
root@cainecf:/home/cainecf# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognised partition table.
Created a new DOS disklabel with disk identifier 0xf199c47c.

Command (m for help): p
Disk /dev/sdc: 1.9 GiB, 1992294400 bytes, 3891200 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: 0xf199c47c

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 (2048-3891199, default 2048):
Last sector, +sectors or +size[K,M,G,T,P] (2048-3891199, default 3891199):
Created a new partition 1 of type 'Linux' and of size 1.9 GiB.

Command (m for help):
```

**Figure 13: Creation of Partition.**

After, we need to change into 95 FAT16 by following these commands: m, t, l, e, w.,

```
1e Hidden W95 FAT1 80 Old Minix      be Solaris boot  ff

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/sdc: 1.9 GiB, 1992294400 bytes, 3891200 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: 0xf199c47c

Device      Boot Start      End Sectors  Size Id Type
/dev/sdc1   2048 3891199 3889152  1.9G  c W95 FAT32 (LBA)

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

root@cainecf:/home/cainecf#
```

**Figure 14: Write W95 FAT16 File system.**



Now we are ready to perform data acquisition. need to mount, to store a copy of our evidence, and evidence hashes: We then validate the data and they match.

```
root@cainecf:/home/cainecf# dcflddd if=/dev/sdb of=/mnt/sdc1/case1/image1.dd conv=noerror, sync  
hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-imagesource.md5.txt  
60672 blocks (1896Mb) written.  
60758+0 records in  
60757+0 records out  
root@cainecf:/home/cainecf#
```

Figure 15: Acquisition of data to case1 with dcfldd.

## Task 5: Analyze the acquired data.

We open autopsy on windows that we downloaded and create a new case.

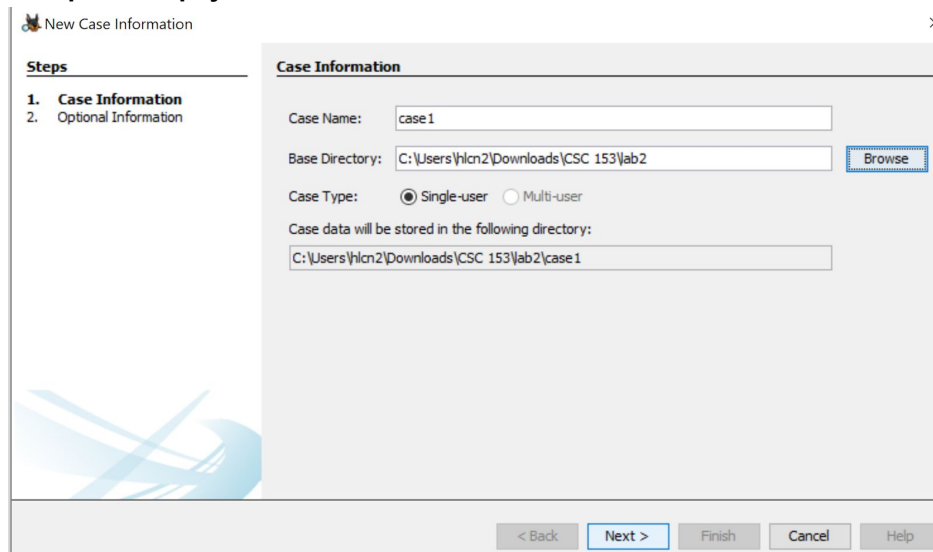


Figure 17: Creating a new case

Then we need to select the correct image we just acquisition earlier,

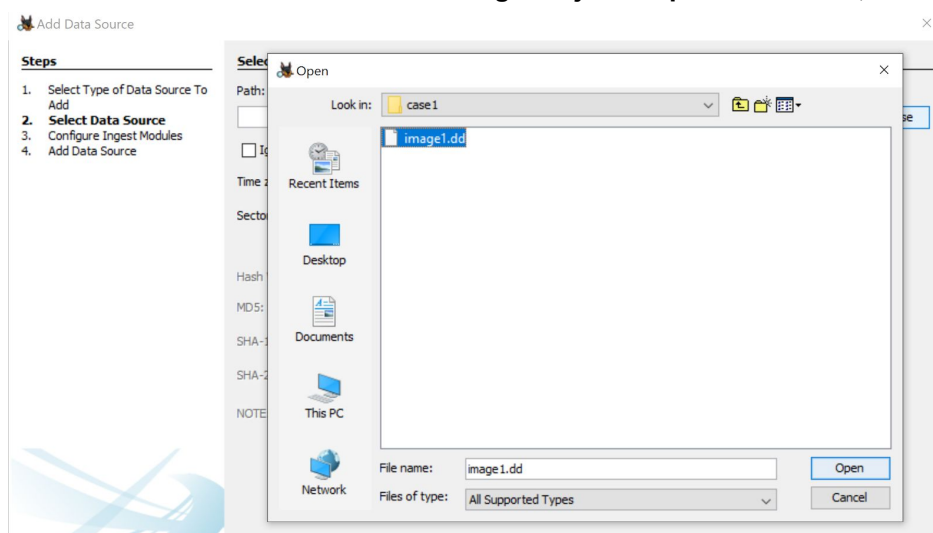
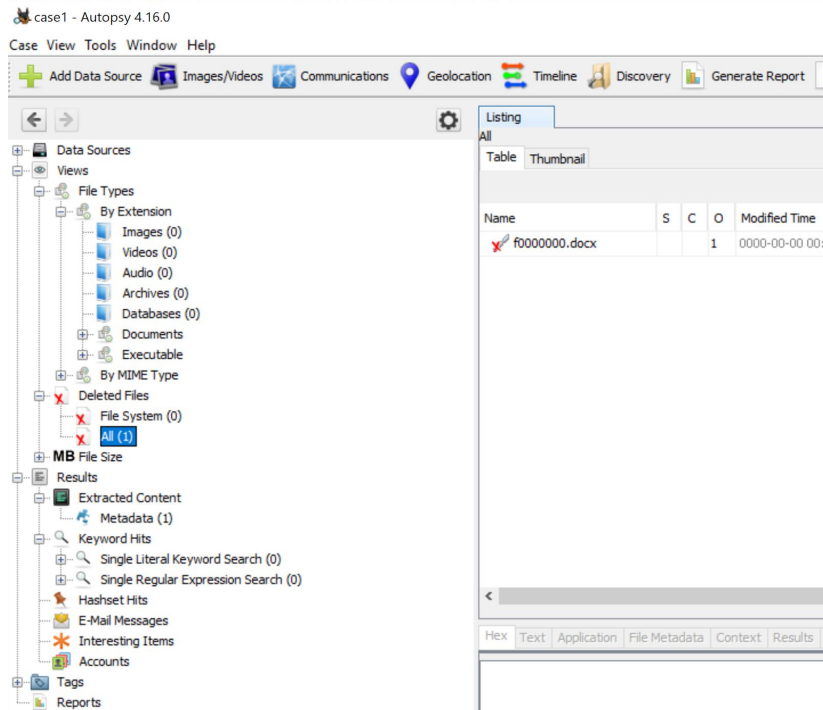


Figure 18: Selected image1.dd from our target drive.

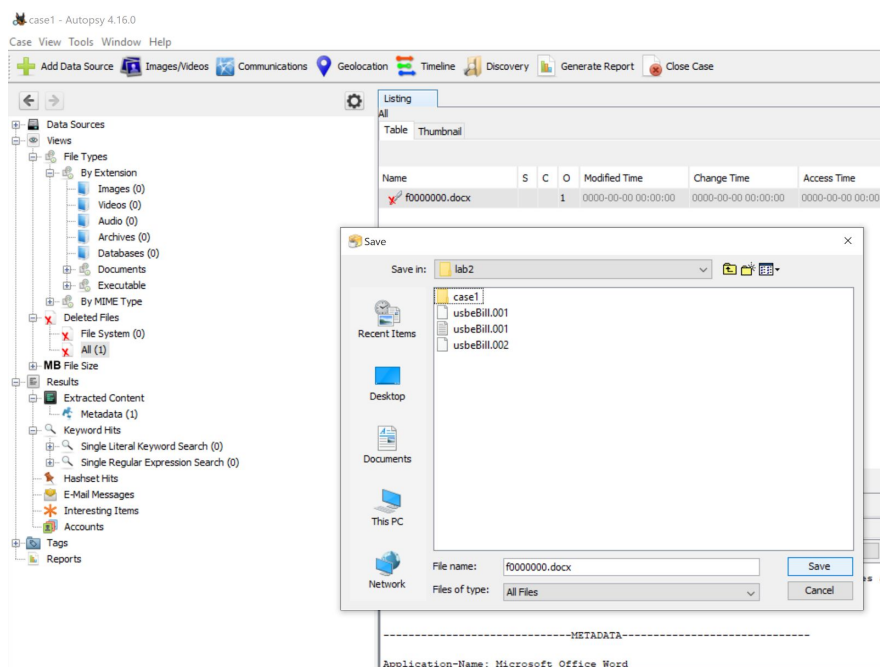


After selecting your image, press open then press “Select All” in step 3 of Autopsy. Then press “next” followed by “Finish” to complete the creation of a new case. Once the data source has completed its analysis. Now we can start analyzing the image. Since this is a new USB and has barely been used only for this class there isn’t much on this drive, which is not surprising after looking into the “View tab”.



**Figure 19: Deleted files, only one. It is a “.docx” file. Nothing else.**

Now we can examine this file by right-clicking the chosen file and extracting it to a certain directory to examine.



**Figure 20: Extracted Deleted file .docx**

Now that the file is extracted I can now open it and examine it using Microsoft word.

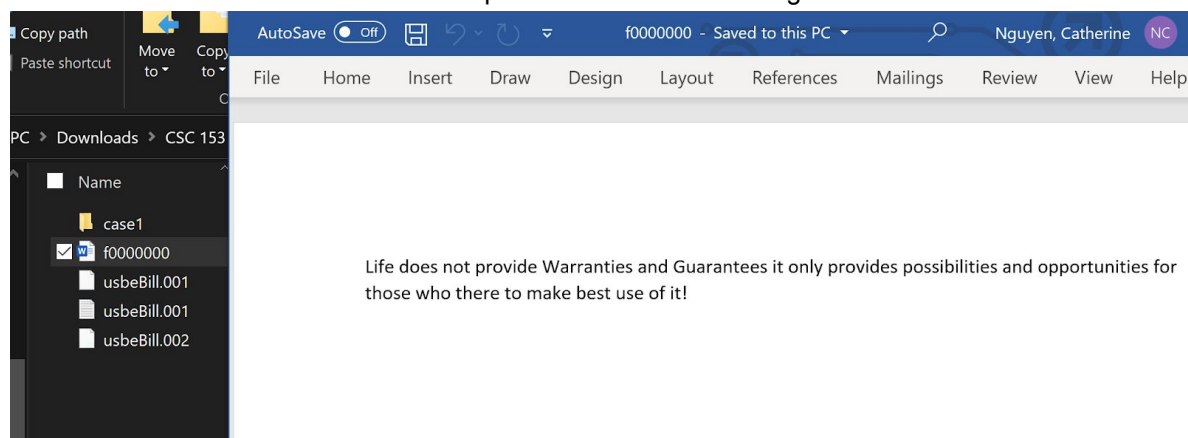


Figure 21: Contents of the extracted deleted .docx. Which matches the file we created at the start.

## Task 6: Perform a dirty word search.

Next is to perform a dirty word search, which searches bit for bit. In this lab, we will only use “Warranties” as the keyword because we know it’s mentioned in our deleted .docx file. To perform the search we click on “Keyword Search” button, and set the settings to ACSII and case insensitive then search “Warranties”.

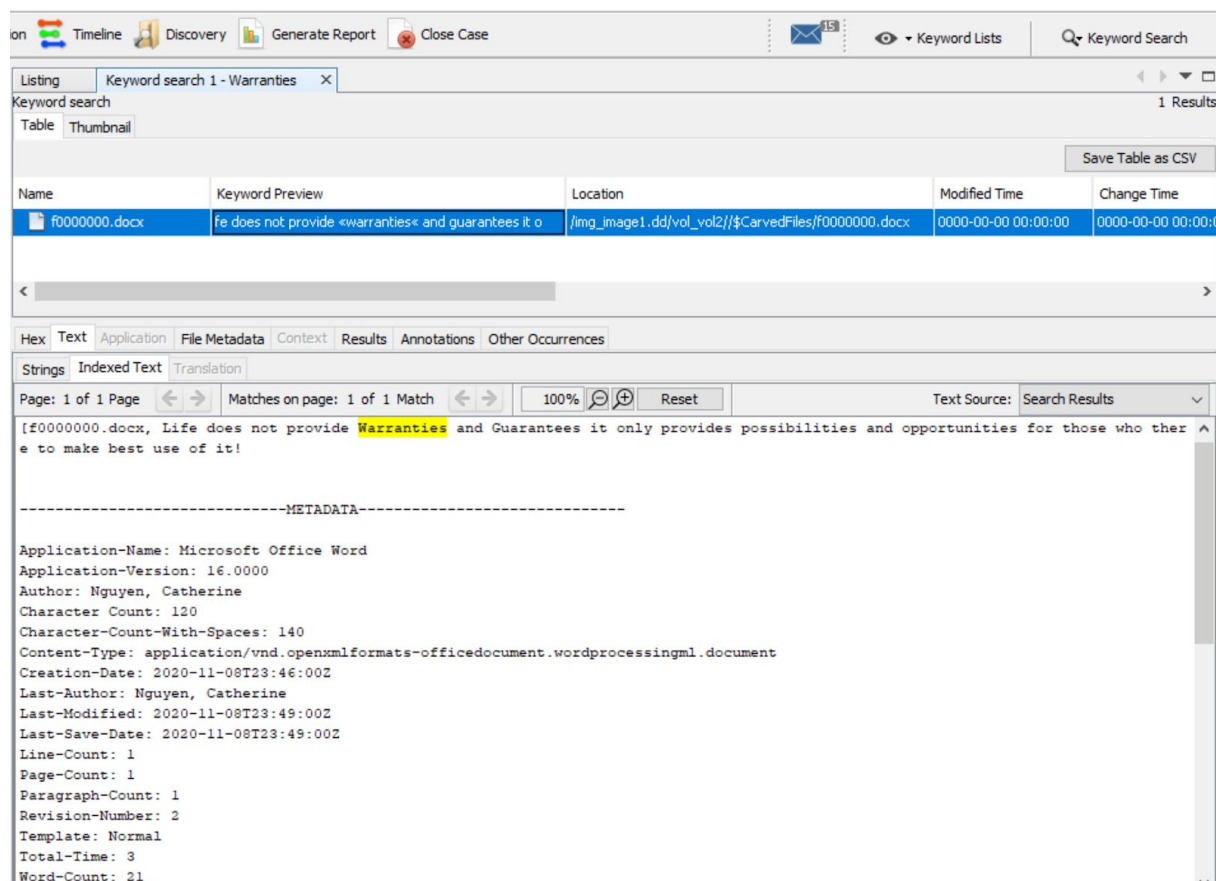


Figure 22: Results of Files that include the word “Warranties” plus hit and file info.

## Task 7: Zero-out the suspect USB drive.

This task is simple. All we need do is go back into CAINE and zero out our suspect/evidence drive.

```
root@cainecf:/home/cainecf# fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 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: 0x0c063046

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1                2048 41943039 41940992   20G 83 Linux

Disk /dev/sdb: 1.9 GiB, 1992294400 bytes, 3891200 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: 0x028e887c

Device      Boot Start      End  Sectors  Size Id Type
/dev/sdb1   *          32 3891199 3891168   1.9G  e W95 FAT16 (LBA)
root@cainecf:/home/cainecf# dd if=/dev/zero of=/dev/sdb1 status=progress
1991238144 bytes (2.0 GB, 1.9 GiB) copied, 2097 s, 950 kB/s
dd: writing to '/dev/sdb1': No space left on device
3891169+0 records in
3891168+0 records out
1992278016 bytes (2.0 GB, 1.9 GiB) copied, 2100.31 s, 949 kB/s
root@cainecf:/home/cainecf#
```

Figure 23: Zero out of Evidence/Suspect Drive “/dev/sdb1”

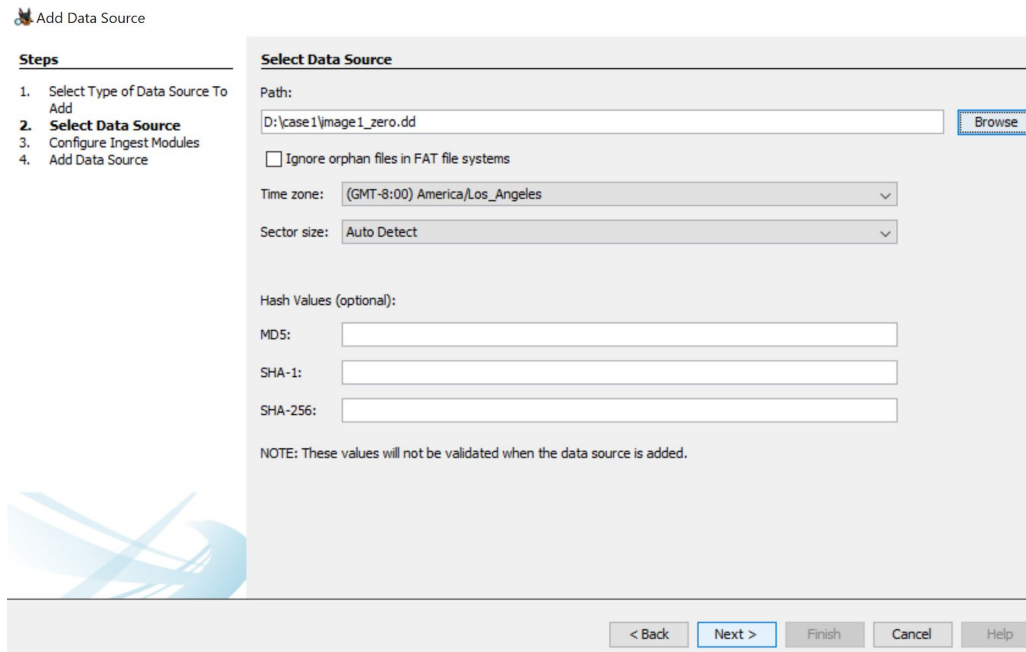
## Task 8: Repeat Task 4-6.

To begin we need to acquisition the data from our new zeroed-out suspect drive onto our target drive. Usually, I believe we could just input the data into the drive but since it's now, under “sdd1” I had to re-mount. Eventually, I was able to transfer the data as an image into the new case and validated with md5.

```
root@cainecf:/home/cainecf# dcfldd if=/dev/sdb of=/mnt/sdd1/case1/image1_zero.dd conv=noerror
, sync hash=md5 hashwindow=0 hashlog=/mnt/sdd1/case1/post-imagesource-zero.md5.txt
60672 blocks (1896Mb) written.
60758+0 records in
60757+0 records out
root@cainecf:/home/cainecf#
```

Figure 24: Acquiring image1\_zero.dd of the suspect drive after it has been zeroed out.

Now we need to create a new case on Autopsy using the new image1\_zero.dd.



**Add Data Source**

**Steps**

1. Select Type of Data Source To Add
2. **Select Data Source**
3. Configure Ingest Modules
4. Add Data Source

**Select Data Source**

Path:

☐ Ignore orphan files in FAT file systems

Time zone:

Sector size:

Hash Values (optional):

MD5:

SHA-1:

SHA-256:

NOTE: These values will not be validated when the data source is added.

< Back Next > Finish Cancel Help

Figure 25: Creating a new case on Autopsy using the image of the zeroed-out suspect drive.

After creating the case and Autopsy finished analyzing the image we find ourselves with an image of absolutely nothing. Unlike before we were able to see the deleted .docx file.

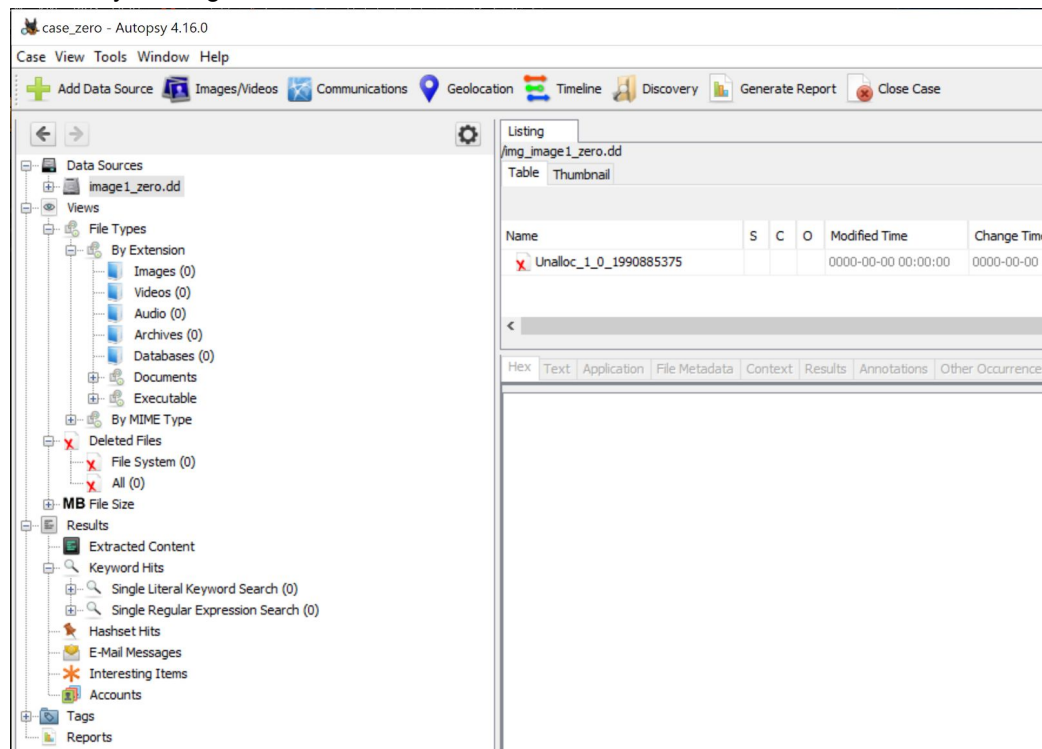


Figure 26: There are no files found in the image of the zeroed-out USB.

The next task was to perform the dirty word search using “Warranties” as our keyword, but since there are no files on the image I do not expect to get a hit.

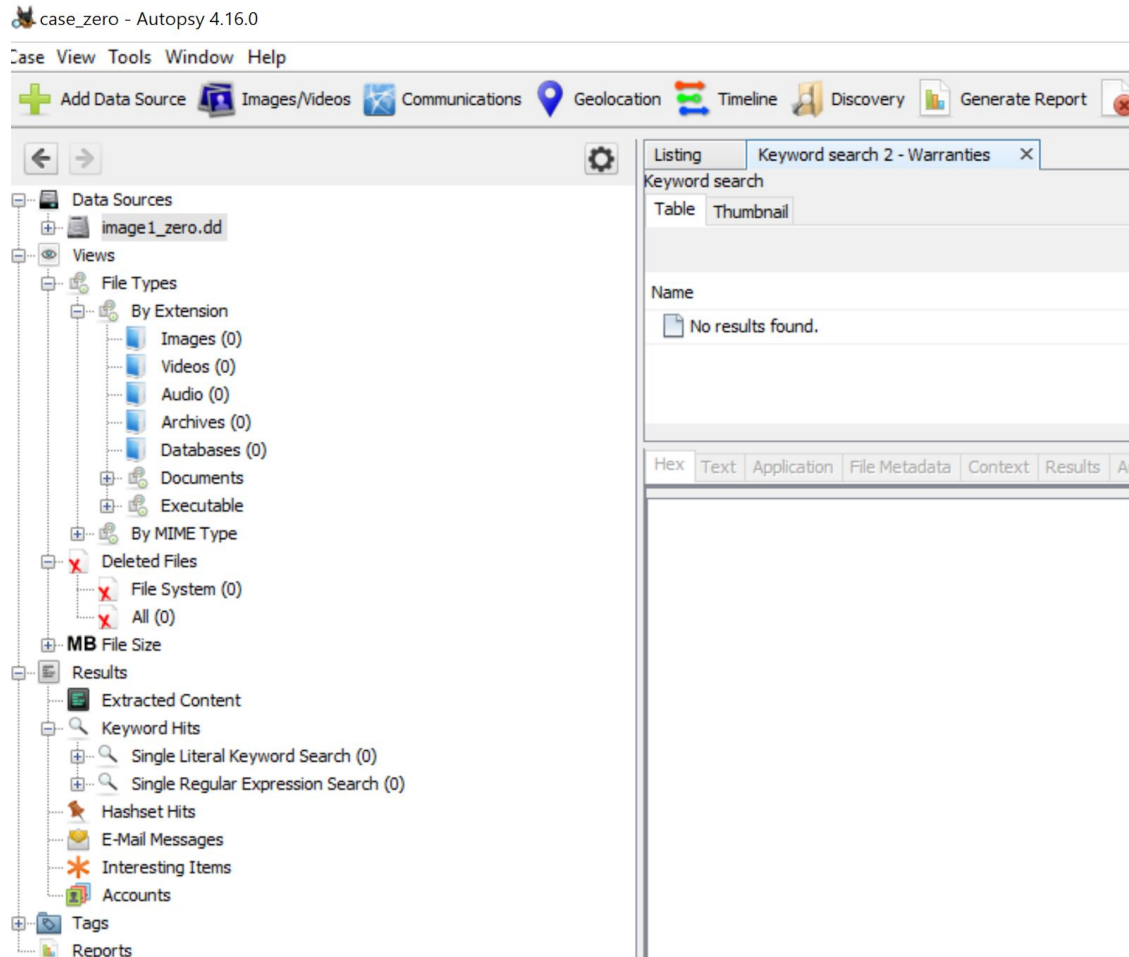


Figure 27: Failed to hit Keyword Search using “Warranties”

## Task 9: Answer the questions.

1. In Task 4, the acquired image has an extension of “.dd”. In Task 3, what is the extension for the image file?
  - a. FTK Imager uses an extension of .001. We can find this by looking at the output of FTK Imager, and we can see the file it left behind in the directory I set for it. “usbeBill.001”.

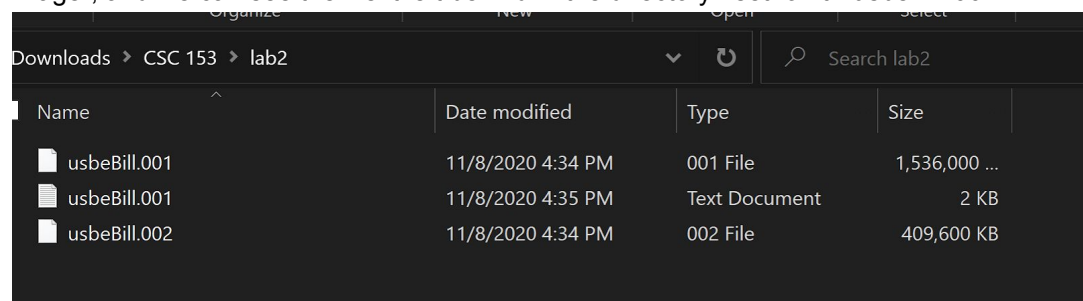


Figure 28: FTK Imager uses .001 extension not .dd.

## 2. In Task 5, how many files are there on the USB drive? What are they?

- a. There were no more files found on the drive besides those that have been deleted. So, the files can be found in the “Deleted Files” section. Which can then be extracted and analyzed. There is only one .doc file shown,

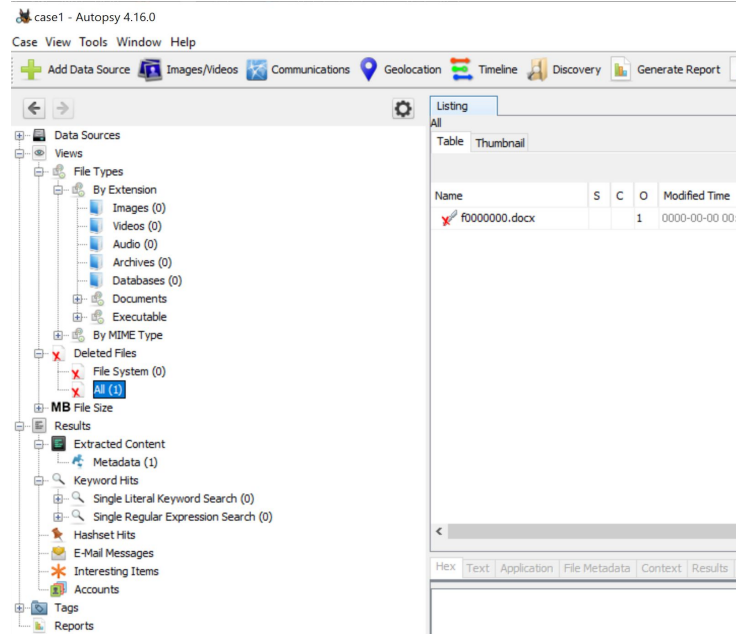


Figure 29: No existing files but the ones .docx that have been deleted.

## 3. In Task 5, which file/files are deleted?

- a. There is only one file that has been deleted. This makes sense because I had bought this USB just for class and it has been zeroed out a couple of times before. However, the one that is visible is the one test.docx we created at the beginning of this lab.

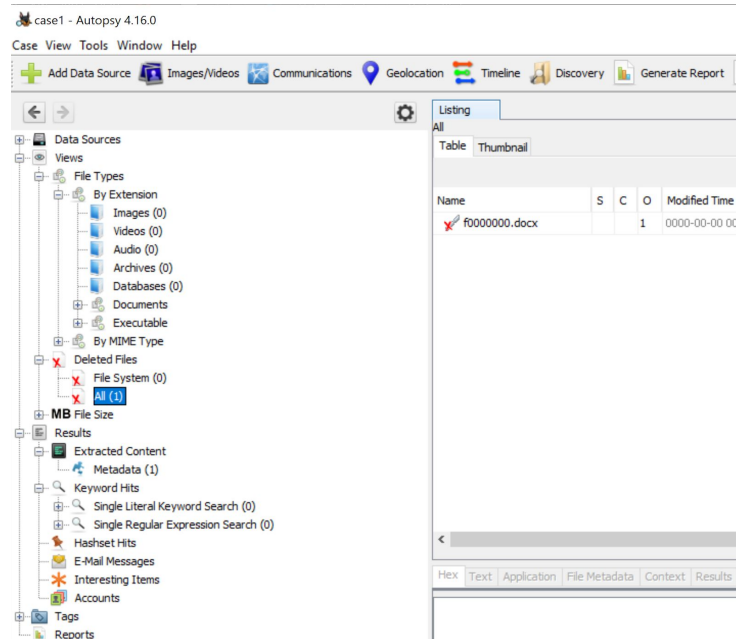


Figure 30: Shown are the deleted files in the image1.dd.



4. In Task 6, are you able to find any hit when you search “Warranties” as the keyword? In which file is the keyword located
- a. We were able to find one hit during our keyword search for Warranties which was found in the f0000000.docx formally known as the test.docx we created before formatting the USB at the beginning of this lab.

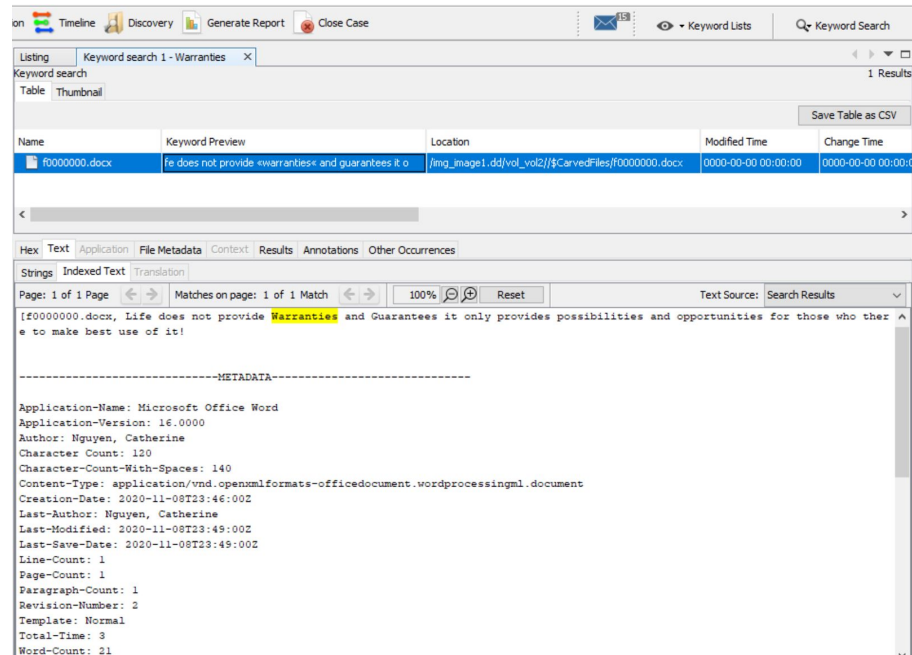


Figure 31: Results for the Keyword search of “Warranties”

5. In Task 8, how many files are there on the USB drive? What are they?
- a. There are no files found in the zeroed out image\_zero.dd, not even deleted files.

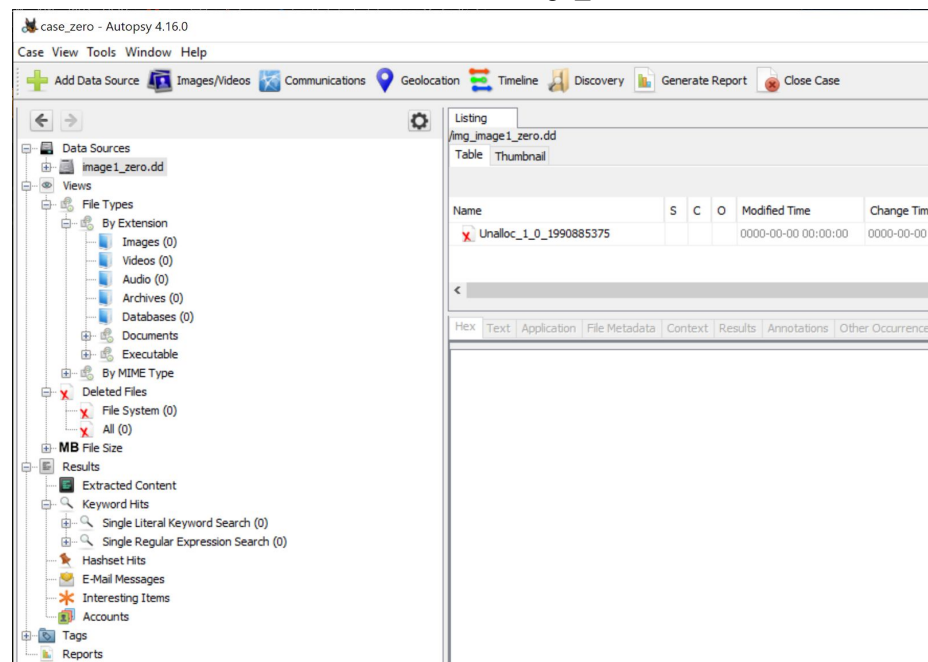


Figure 32: No existing files found after zero out.



6. In Task 2, you performed a “disk format” operation towards the USB drive. Did this operation completely erase the “test.docx” file in the USB drive? How do you know?
- a. No, it did not completely erase the test.docx file in the USB. We know this because we were able to see it as a deleted file in Autopsy and we were able to extract it on to our device and open it up for analysis. Even after it has been formatted.

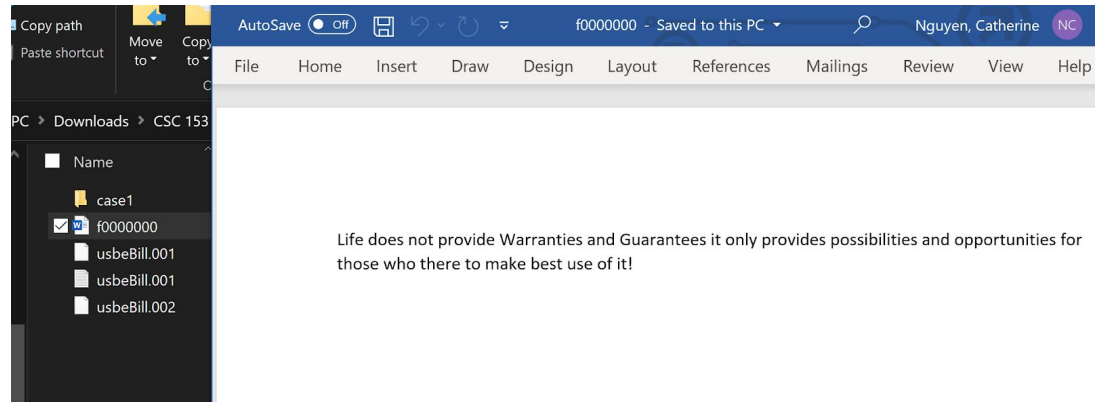


Figure 33: The contents of f0000000.docx aka test.docx

7. In Task 7, you performed a “zero out” operation towards the USB drive. Did this operation completely erase the “test.docx” file in the USB drive? How do you know?
- a. Yes, because no files were to be found after we zeroed out the suspect file. We know this because before zeroing out the drive we were still able to recover the test.docx file, now we can’t and the keyword search for “Warranties” has no hits.

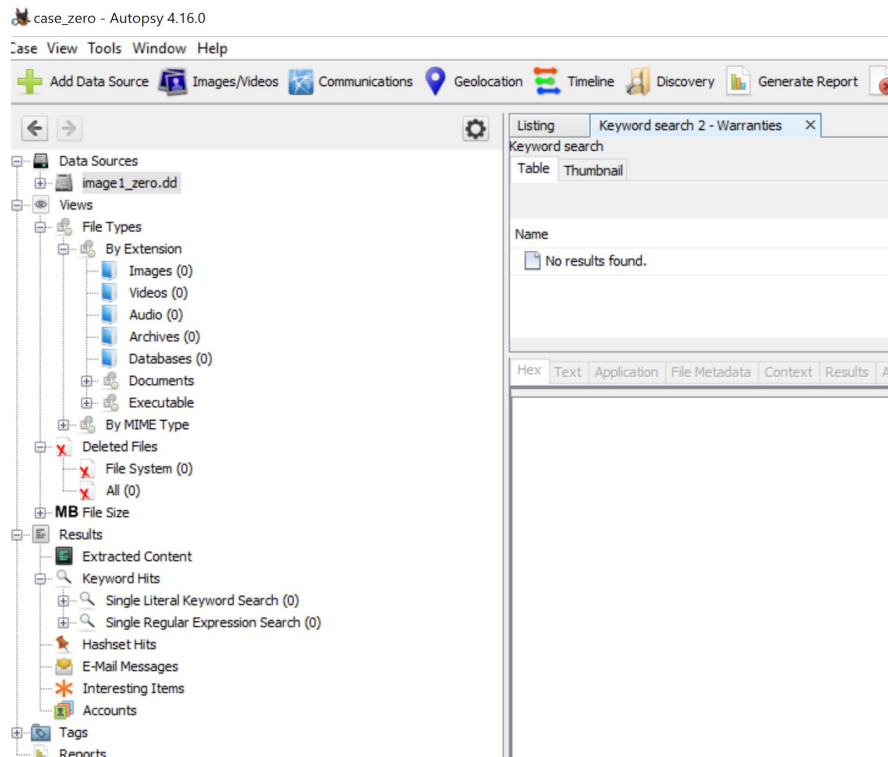


Figure 34: No Files found after being zeroed out. Zero hits for “Warranties” search

8. Did have any surprise in Task 5? Did you see any other files other than “test.doc” or “test.docx”?

- a. I was not surprised at all. This is a brand new drive and I bought it for this class so I expected it to have nothing, but the test.docx we created at the beginning.

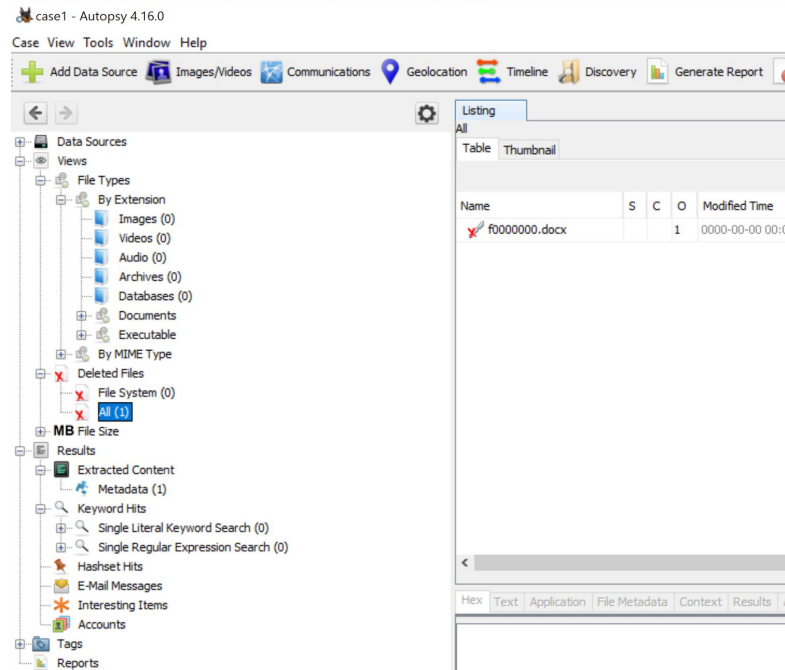


Figure 35: No other files found but “test.docx”

9. In Task 8, did you see any other files other than “test.doc” or “test.docx”?

- a. No other files could be seen. Not even the “test.docx”

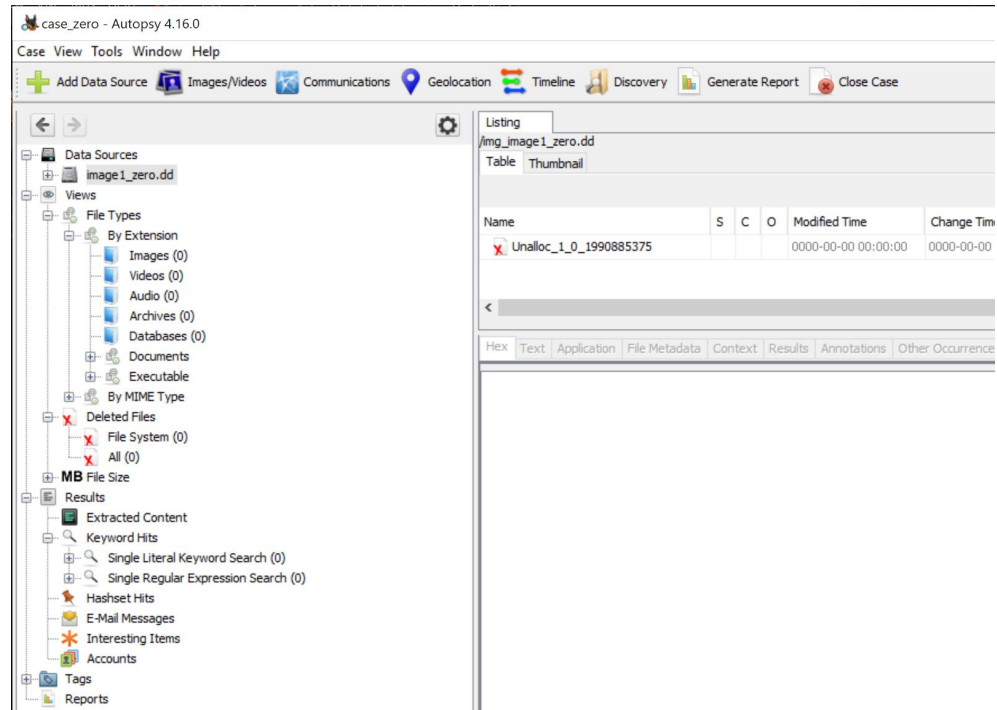


Figure 36: No other files to be found on the zero out image\_zero.dd.

**10. To summarize the questions above, what are the difference between disk formatting in Windows and the zero-out operation?**

- a. Formatting files only deletes file systems found on the disk, meaning that it will no longer reference the data, but the sectors where the data can be found remains unchanged. So it can be extracted.
- b. The zero-out operation goes into each sector of the drive and sets each one of its bits to zero. This causes everything stored in the disk to be destroyed. Meaning all the data previously stored will be lost. It can no longer be extracted.