



## **FORENSICS V2 LAB SERIES**

### **Lab 11: Picture File Analysis**

**Document Version: 2021-01-14**

Copyright © 2021 Network Development Group, Inc.  
[www.netdevgroup.com](http://www.netdevgroup.com)

NETLAB+ is a registered trademark of Network Development Group, Inc.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and other countries. Google is a registered trademark of Google, LLC. Amazon is a registered trademark of Amazon in the United States and other countries.

## Contents

Introduction .....	3
Objectives.....	3
Lab Topology .....	4
Lab Settings .....	5
1    Extracting EXIF Data from Picture Files.....	6
2    Carving Pictures from the Thumbs.db File.....	13

## Introduction

With smartphones becoming so popular, many photos are embedded with additional information that can provide location information as well as attribute evidence items to persons. This module will cover the types of data that can be ascertained from picture files and how to do it.

## Objectives

- ) Learn what EXIF data is
- ) Learn how to extract EXIF data and interpret it
- ) Learn how to corroborate camera specs with other evidence items
- ) Learn how to extract data from Thumbs.db files

## Lab Topology



## Lab Settings

The information in the table below will be needed to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address / Subnet Mask	Account (if needed)	Password (if needed)
Caine	172.16.16.30	caine	Train1ng\$
CSI-Linux	172.16.16.40	csi	csi
DEFT	172.16.16.20	deft	Train1ng\$
WinOS	172.16.16.10	Administrator	Train1ng\$

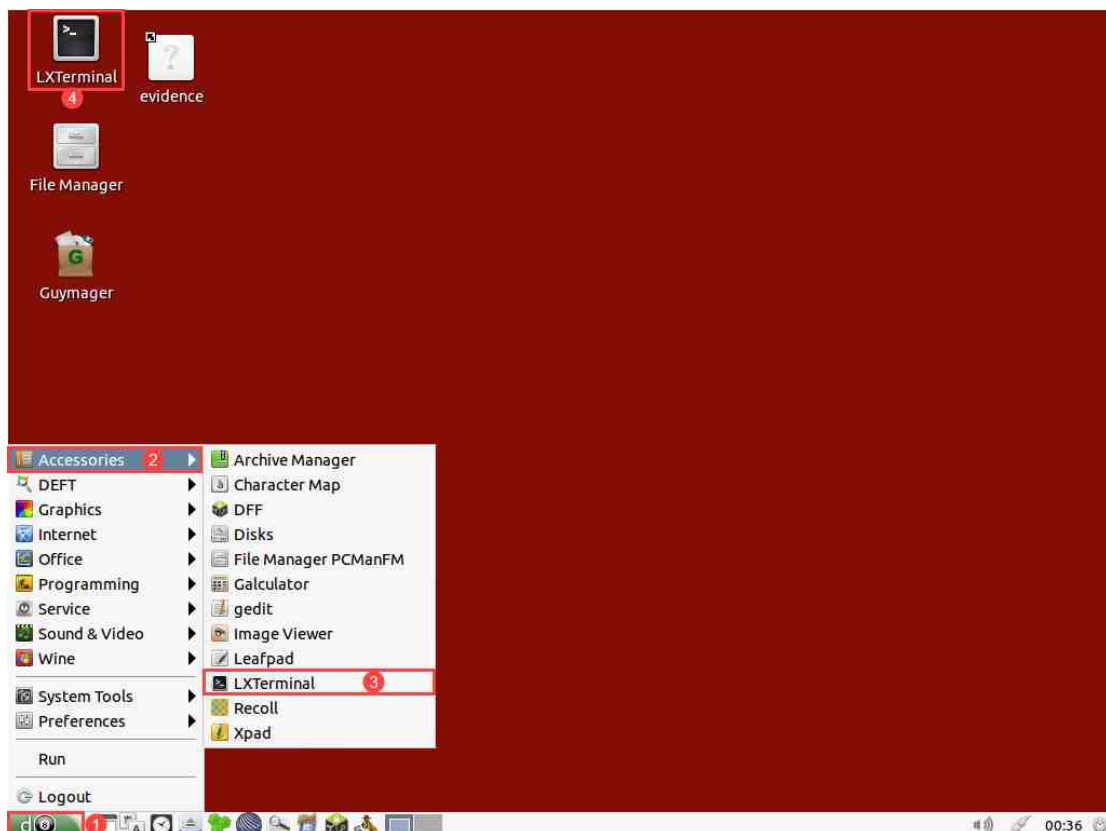
## 1 Extracting EXIF Data from Picture Files

Picture files are very common, especially now that almost every person owns a powerful handheld audio-visual recording device, the smartphone. In this exercise, we will delve into additional data stored within picture files and teach you how to extract and interpret this data.

1. To begin, launch the DEFT Linux virtual machine to access the graphical login screen. Log in as `deft` using the password: `Training$`



2. Once you are logged into the VM, launch the LXTerminal shell by navigating to Application Menu > Accessories > LXTerminal as seen in items 1, 2, and 3 below. Alternatively, you can open LXTerminal from the Desktop by double-clicking the icon as seen in item 4 below.



- Once the terminal opens, let us navigate to the folder that contains the picture files. Do this by typing the following command and pressing Enter when you are done.

```
cd Evidence_Files/FOR_LAB_011/Pics
```



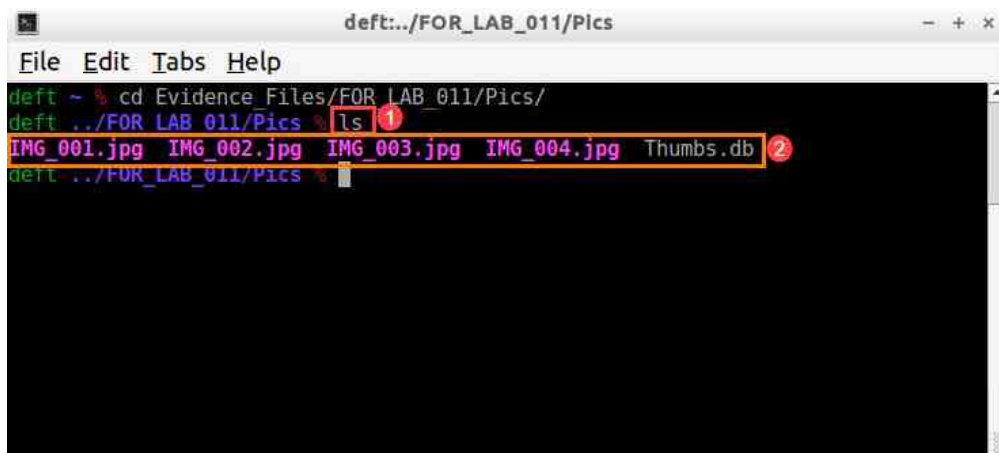
A terminal window titled 'deft:~/FOR\_LAB\_011/Pics' with a menu bar (File, Edit, Tabs, Help). The command 'cd Evidence\_Files/FOR\_LAB\_011/Pics/' is entered and highlighted with a red box. The prompt changes to 'deft ../FOR\_LAB\_011/Pics %'.



Typically, all commands are lowercase. However, Linux file names are case-sensitive. Ensure to type the commands as seen.

- Now that we are in the folder, let us look at the files within this directory. Type the following command and press Enter to list all the files within the directory. As you can see in item 2 below, there are 5 files in the directory; 4 of these files are JPG picture files, and 1 is a database file. Let us focus on the picture files for now.

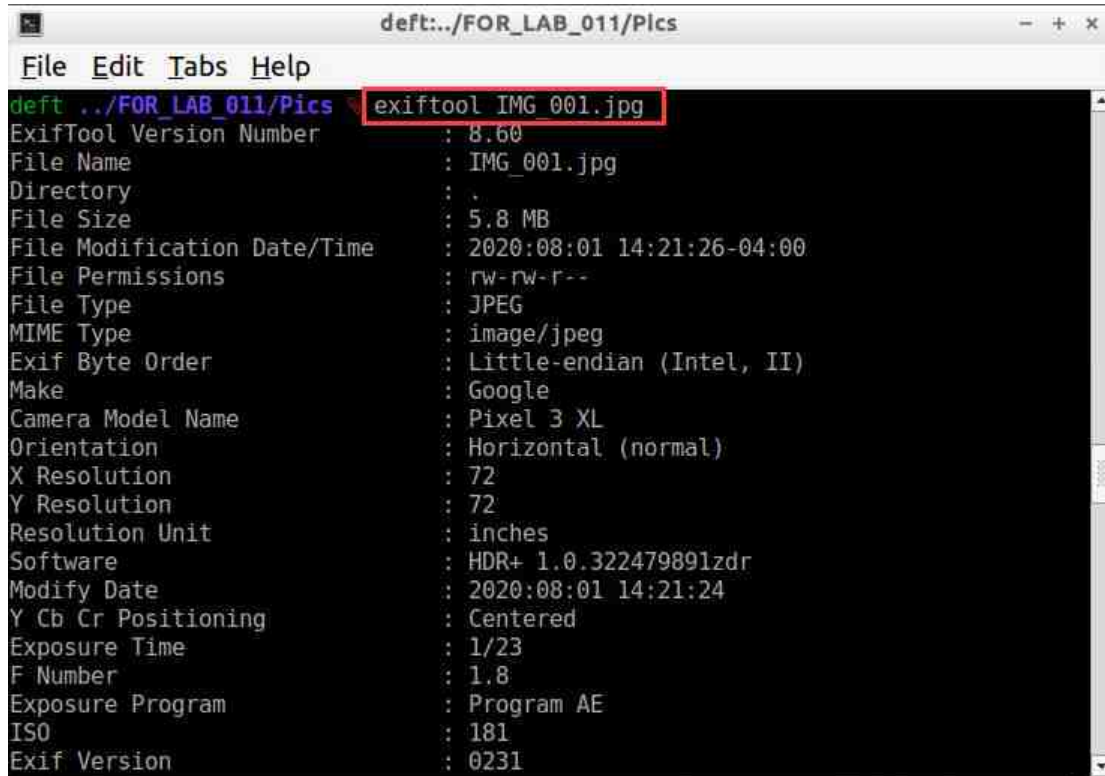
```
ls
```



A terminal window titled 'deft:~/FOR\_LAB\_011/Pics' with a menu bar (File, Edit, Tabs, Help). The command 'ls' is entered and highlighted with a red box and a red circle with the number 1. The output shows 'IMG\_001.jpg IMG\_002.jpg IMG\_003.jpg IMG\_004.jpg Thumbs.db' highlighted with a red box and a red circle with the number 2. The prompt changes to 'deft ../FOR\_LAB\_011/Pics %'.

5. We will be using a picture file analysis tool called Exiftool to extract EXIF data from these picture files. The command is very simple. Let us test it out by typing the following command and pressing Enter.

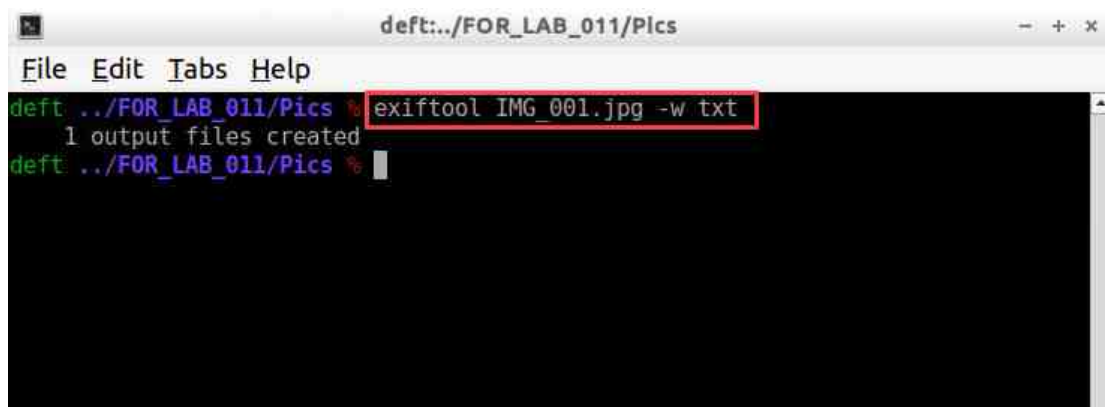
```
exiftool IMG_001.jpg
```



```
deft:~/FOR_LAB_011/Pics
File Edit Tabs Help
deft ~/FOR_LAB_011/Pics % exiftool IMG_001.jpg
ExifTool Version Number      : 8.60
File Name                    : IMG_001.jpg
Directory                    : .
File Size                    : 5.8 MB
File Modification Date/Time   : 2020:08:01 14:21:26-04:00
File Permissions              : rw-rw-r--
File Type                    : JPEG
MIME Type                    : image/jpeg
Exif Byte Order               : Little-endian (Intel, II)
Make                         : Google
Camera Model Name             : Pixel 3 XL
Orientation                   : Horizontal (normal)
X Resolution                  : 72
Y Resolution                  : 72
Resolution Unit               : inches
Software                      : HDR+ 1.0.322479891zdr
Modify Date                   : 2020:08:01 14:21:24
Y Cb Cr Positioning           : Centered
Exposure Time                 : 1/23
F Number                      : 1.8
Exposure Program              : Program AE
ISO                           : 181
Exif Version                  : 0231
```

6. As you can see, the command provides a long list of metadata. It is a little tricky to handle it in the shell so let us run the command again to write the data in a text file. Type the following command and then press Enter.

```
exiftool IMG_001.jpg -w txt
```

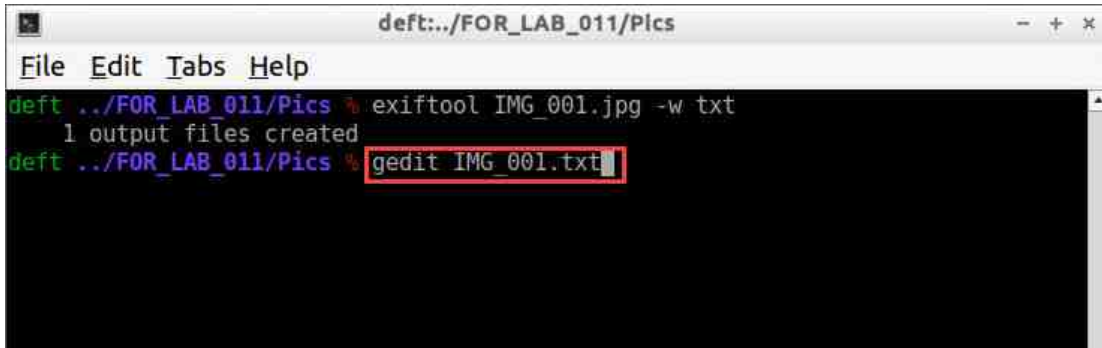


```
deft:~/FOR_LAB_011/Pics
File Edit Tabs Help
deft ~/FOR_LAB_011/Pics % exiftool IMG_001.jpg -w txt
1 output files created
deft ~/FOR_LAB_011/Pics %
```



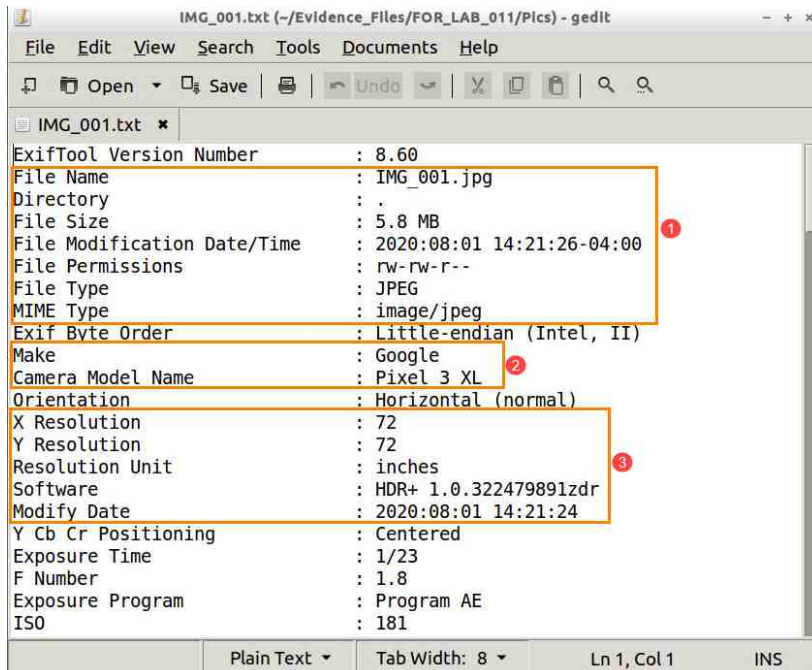
7. Now that the file is extracted, let us look at it using the Gedit text editor. To do this, type the following command and press Enter.

```
gedit IMG_001.txt
```



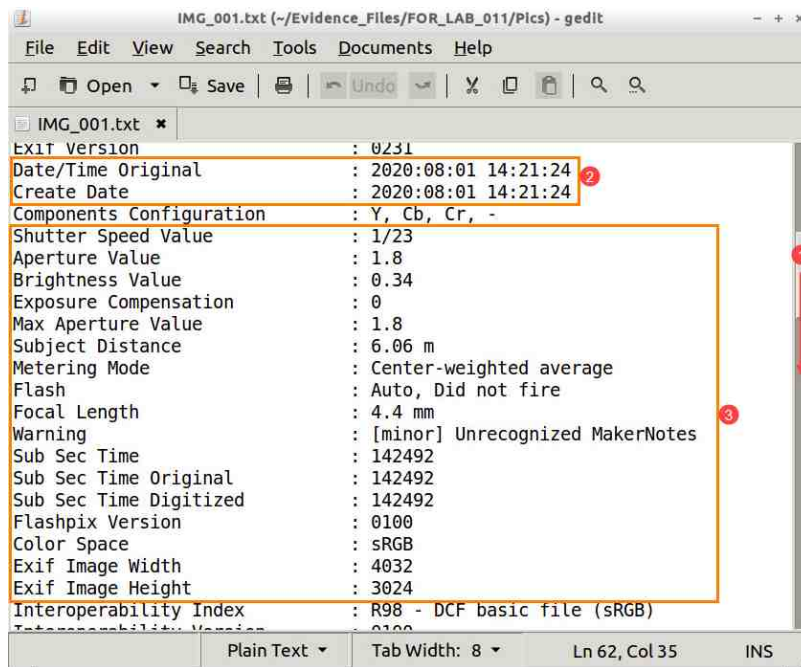
```
deft:~/FOR_LAB_011/Pics
File Edit Tabs Help
deft ../FOR_LAB_011/Pics % exiftool IMG_001.jpg -w txt
1 output files created
deft ../FOR_LAB_011/Pics % gedit IMG_001.txt
```

8. The text file we generated will now open and reveal the same data we saw earlier in the shell. Now let us look at the parsed data and see what we can learn from it.
  - a. The first set of metadata, seen in item 1 below, lists the name of the file, its permissions, when it was last modified, the file size, and the file type. This is data that is typically found in the file system as well.
  - b. The data highlighted as item 2 provides information about the camera. It lists the make and model of the device that captured the picture. In the picture below, the metadata indicates that the photo was taken by a Google Pixel 3 XL smartphone.
  - c. The data highlighted as item 3 provides more information about the image and camera. It lists things such as the software version for the camera and the resolution of the photo.

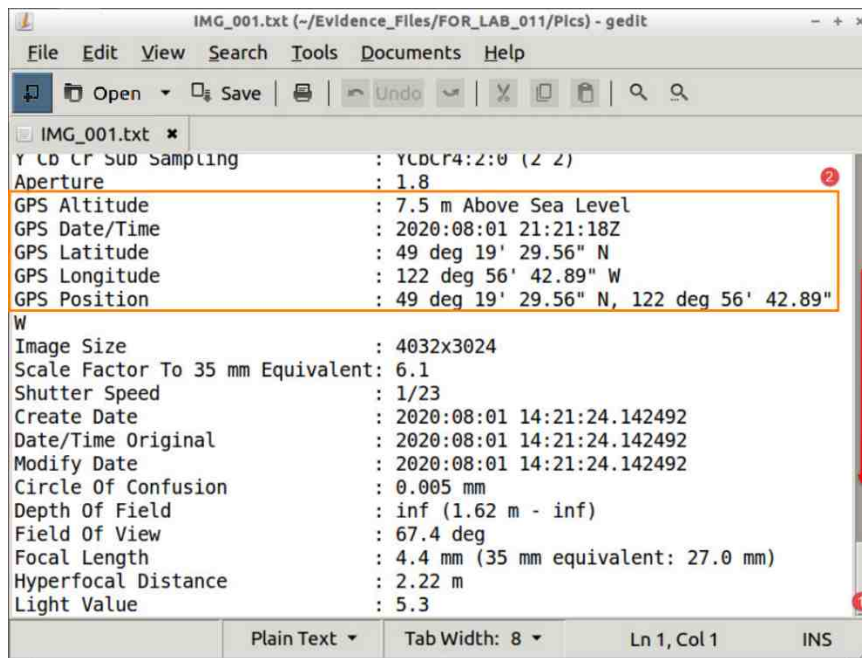


```
IMG_001.txt (~/.Evidence_Files/FOR_LAB_011/Pics) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
IMG_001.txt x
ExifTool Version Number      : 8.60
File Name                    : IMG_001.jpg
Directory                    : .
File Size                    : 5.8 MB
File Modification Date/Time   : 2020:08:01 14:21:26-04:00
File Permissions              : rw-rw-r--
File Type                    : JPEG
MIME Type                    : image/jpeg
Exif Byte Order               : Little-endian (Intel, II)
Make                         : Google
Camera Model Name             : Pixel 3 XL
Orientation                  : Horizontal (normal)
X Resolution                  : 72
Y Resolution                  : 72
Resolution Unit               : inches
Software                     : HDR+ 1.0.322479891zdr
Modify Date                   : 2020:08:01 14:21:24
Y Cb Cr Positioning           : Centered
Exposure Time                 : 1/23
F Number                     : 1.8
Exposure Program              : Program AE
ISO                           : 181
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

9. Let us scroll down a bit and look at some more data. Do this by using the mouse wheel or dragging the scroll bar down as seen in item 1 below.
  - a. Scroll until you get to Date/Time Original as seen in item 2 below. As the name suggests, this is the date and time that the photo was taken and can be a great way of confirming the correct date that the photo was taken. This metadata often conflicts with the file system date and times for computers and other digital devices and is normally more reliable.
  - b. The data highlighted in item 3 provides more details about the photo and the camera. It contains details such as whether the camera's flash was used when taking the photo, the size of the image in pixels, etc.

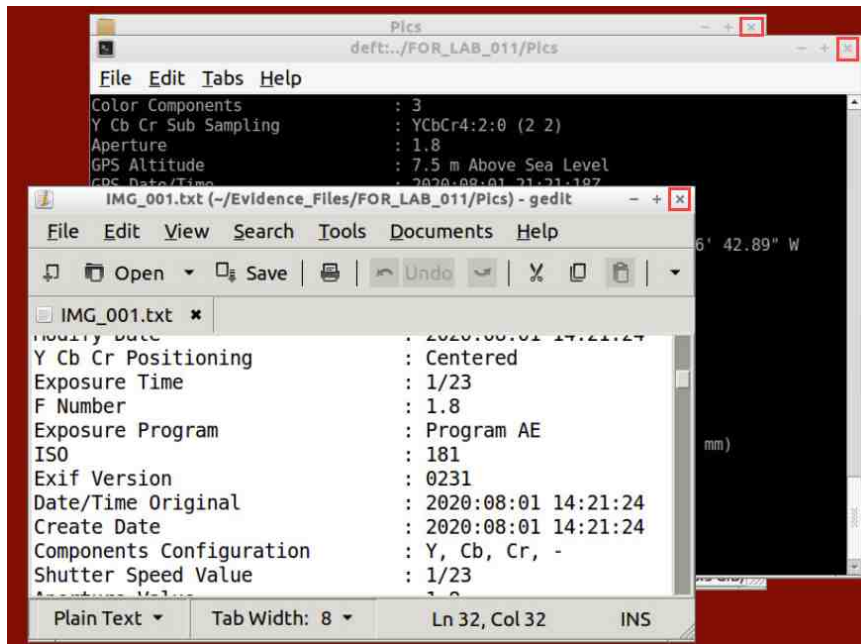


10. Let us look at one other very important and useful piece of data found in media files
- GPS data. Find it by scrolling all the way to the bottom of the text file, as seen in item 1 below. Now, look at the data highlighted as item 2 below. This is GPS data that provides the approximate location that the photo was taken. As you can see, it contains the same date and time as before. It also provides the altitude and GPS coordinates that can pinpoint the specific location. You can note this position and search for it on your own map as the lab workstations are not connected to the internet.



11. Now that you are done reviewing this file, repeat the above steps to review all the metadata for the JPG files called IMG002.jpg, IMG003.jpg, and IMG004.jpg. Once you are done reviewing the data from each picture file, note the different creation dates and times and GPS coordinates.
12. You are now at the end of this exercise. As you learned by using Exiftool, tons of data exist within picture files, and this data can be used to associate people with devices. It can also associate device users with locations, activities, and other devices. Now let us move on to another aspect of picture file analysis - the Thumbs.db file.

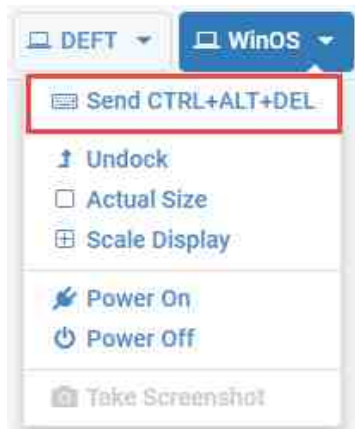
13. We will be switching to the Windows VM, so close all the open programs by clicking the X at the top-right corner as highlighted below.



## 2 Carving Pictures from the Thumbs.db File

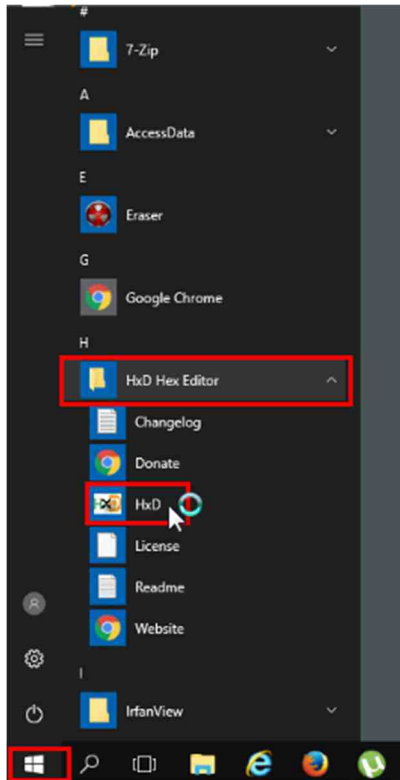
Thumbs.db is a hidden database file that contains smaller images of picture files within a folder. This file is created when the user views the pictures as thumbnails inside certain folders. The Thumbs.db file can be a goldmine, especially if the picture files within the folder are already deleted. Let us carve the Thumbs.db file we saw inside the Pics folder earlier. We will use HxD to do the carving, so let us log into the Windows VM.

1. To begin, launch the WinOS virtual machine to access the graphical login screen.
  - a. Select Send CTRL+ALT+DEL from the dropdown menu to be prompted with the login screen.

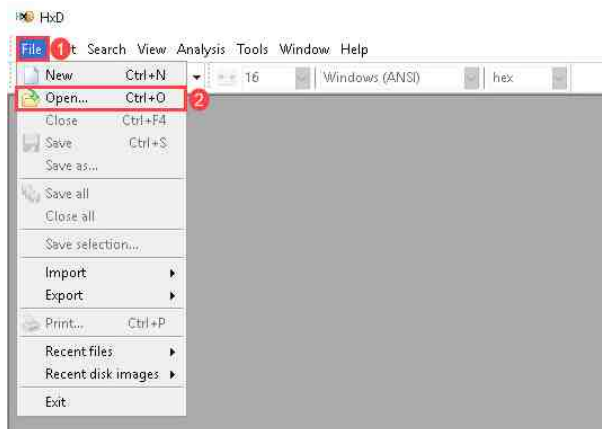


- b. Log in as Administrator using the password: Training\$

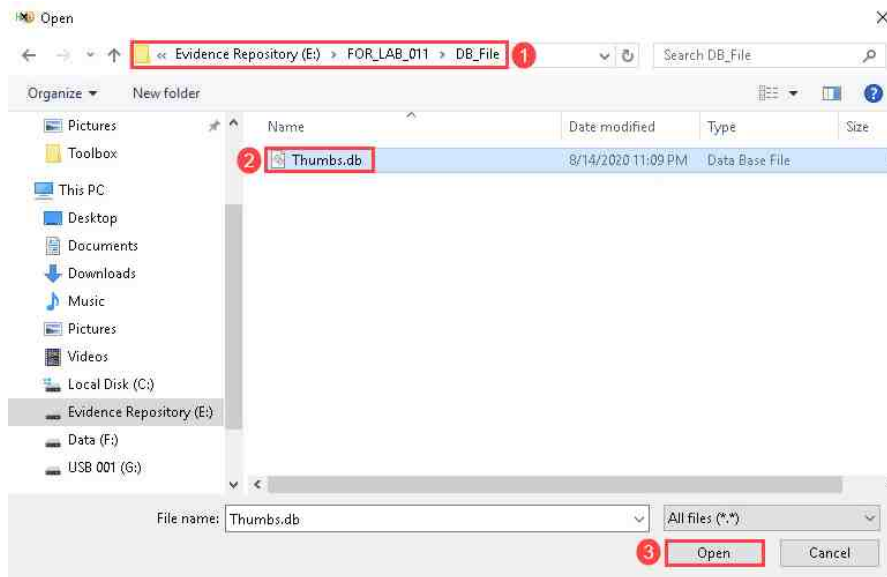
2. Once you are logged into the VM, launch the HxD program from the Windows menu by navigating to Start Menu > HxD Hex Editor > HxD. Alternatively, you can open HxD Hex Editor from the desktop by double-clicking the icon called HxD.



3. HxD will open and since you are familiar with it from the previous labs, let us dive right in by loading the Thumbs.db file. Begin by navigating to File > Open as highlighted in items 1 and 2 below.



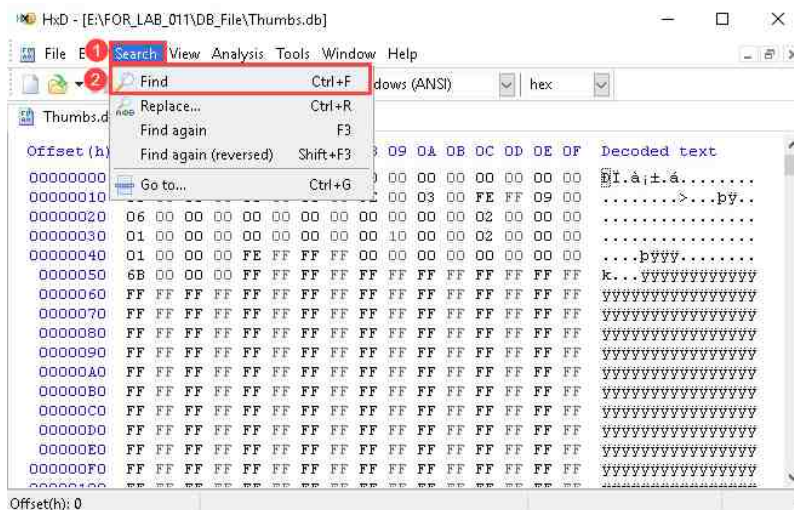
- The Open window will appear. Browse to Evidence Repository (E:) > FOR\_LAB\_011 > DB\_File as seen in item 1 below. Once there, click the file called Thumbs.db and then click Open as seen in items 2 and 3 below.



**Please  
Note**

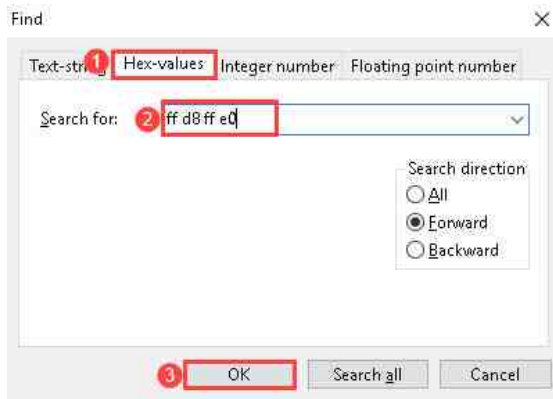
The Thumbs.db file is a hidden system file which means it cannot be seen if the hidden operating system files and hidden files options are enabled or if it is being viewed using third party viewers.

- The file called Thumbs.db will now open in HxD, and you will see the hex and text representation of the file populate the main GUI window. Now that the file is open, let us search for the JPG signature. To do this, open the Find window by navigating to Edit > Find as seen in items 1 and 2 below. Alternatively, you can use Ctrl+F.

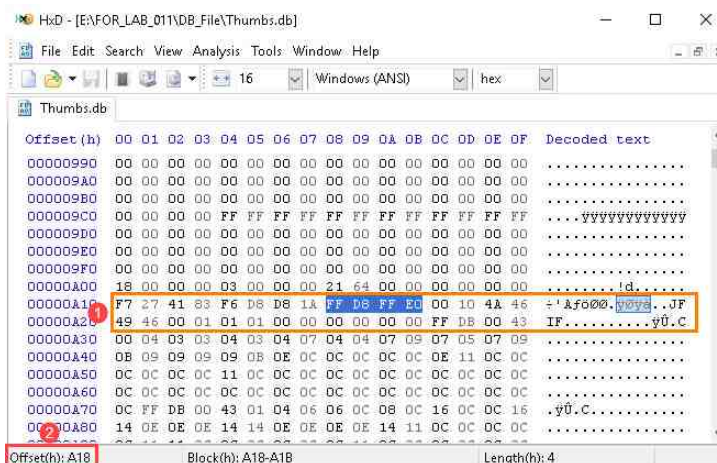




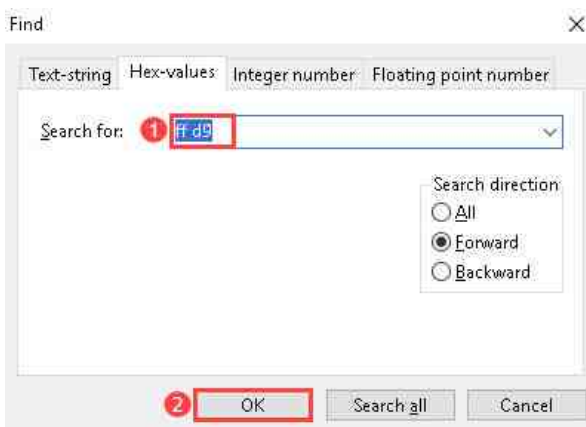
- When the Find window opens, click the Hex-values tab as highlighted in item 1 below and type FF D8 FF E0 in the Search for field. Then click OK as seen in items 2 and 3. The value you typed is the header for JPG files and will take you to the beginning of a JPG file if it exists inside the file.



- If you did everything correctly, you will be taken to the beginning of the first JPG file, as seen below. Now that we have seen a header, let us note the offset 0xA18, as seen in item 2 below, and move on to searching for the end of file (EOF) signature.

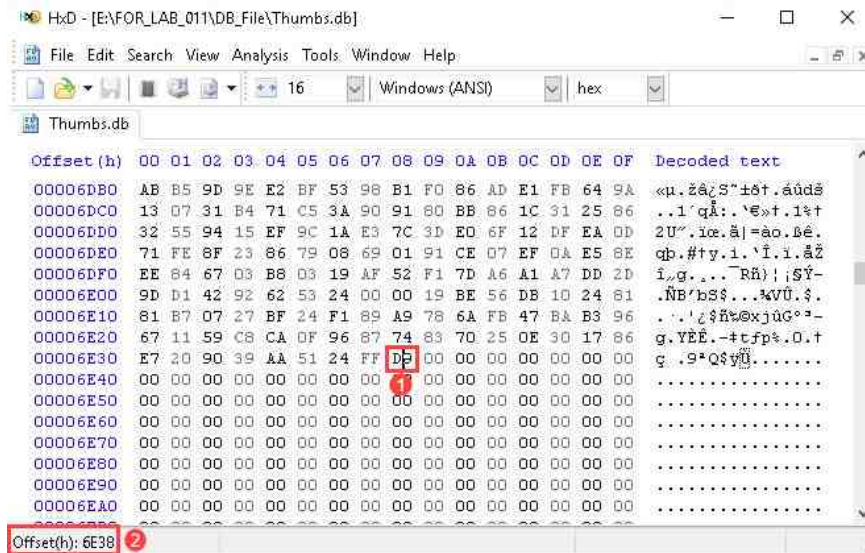


- Go back to the Find window and type FF D9 in the Search for field and then click OK, as seen in items 1 and 2 below.

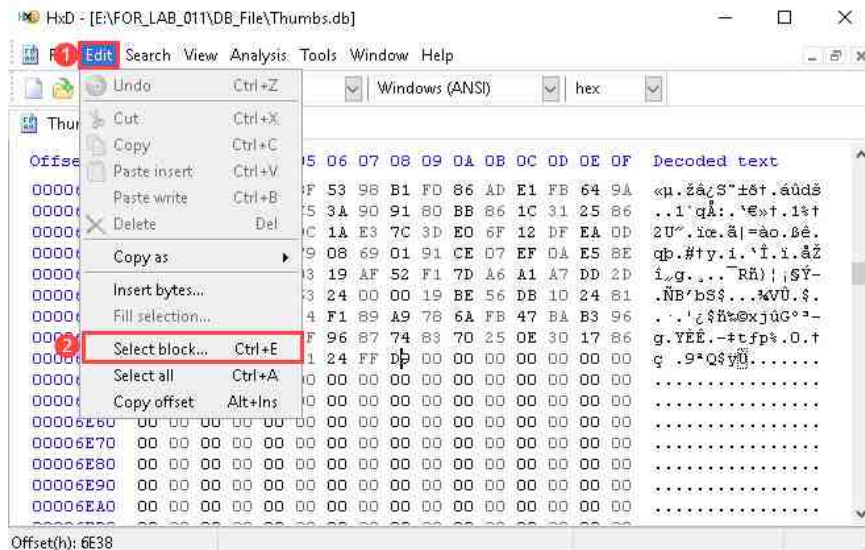




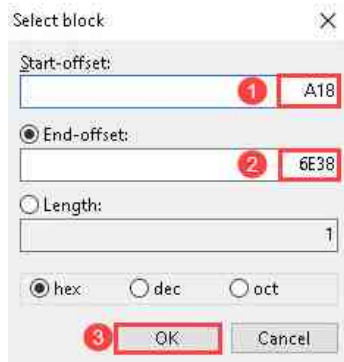
9. You will now be taken to the end of file signature (footer) for this file. Let us note the offset by clicking the last value, as seen in item 1 below. This will reveal the offset 0x6E38 in the status bar, as seen in item 2. Note this offset as this denotes the end of the JPG file.



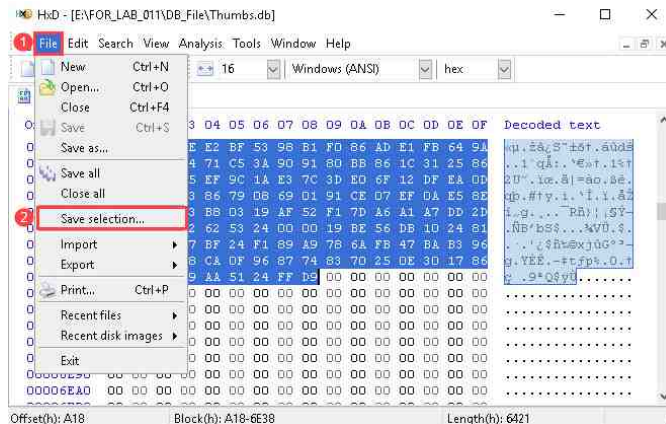
10. Now let us use the Select block feature to highlight the data between the first and last byte offsets. To do this, navigate to Edit > Select block, as seen in items 1 and 2 below. Alternatively, you can use Ctrl+E.



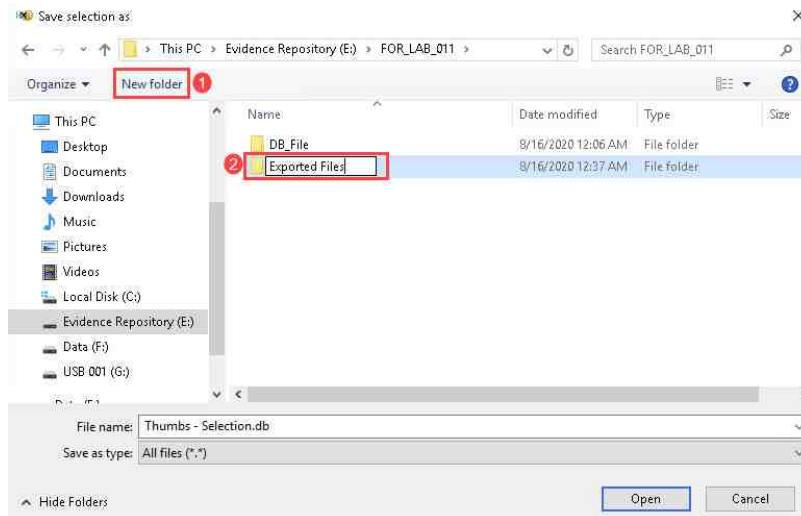
11. In the Select block window, type the respective offsets as seen in items 1 and 2 below and then click OK as seen in item 3. This will highlight the entire area between the two offsets.



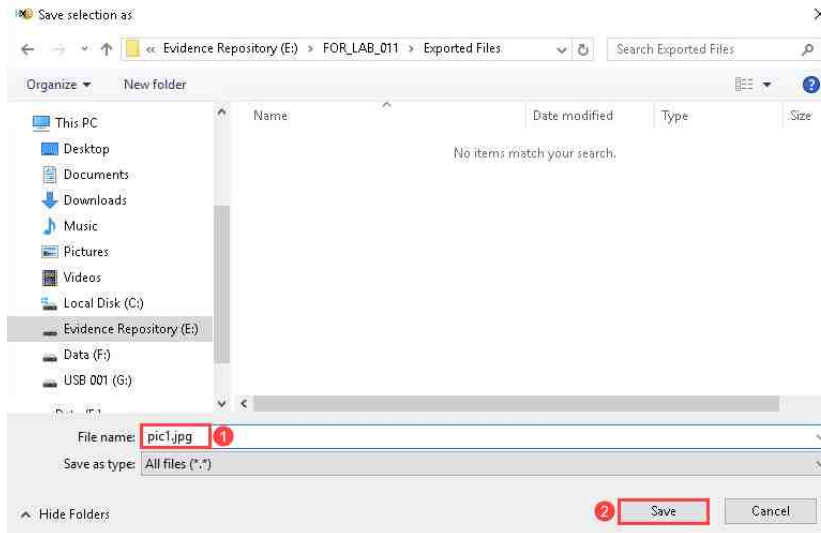
12. Now that the file is highlighted, let us export it. To do this, navigate to File > Save selection as seen in items 1 and 2 below.



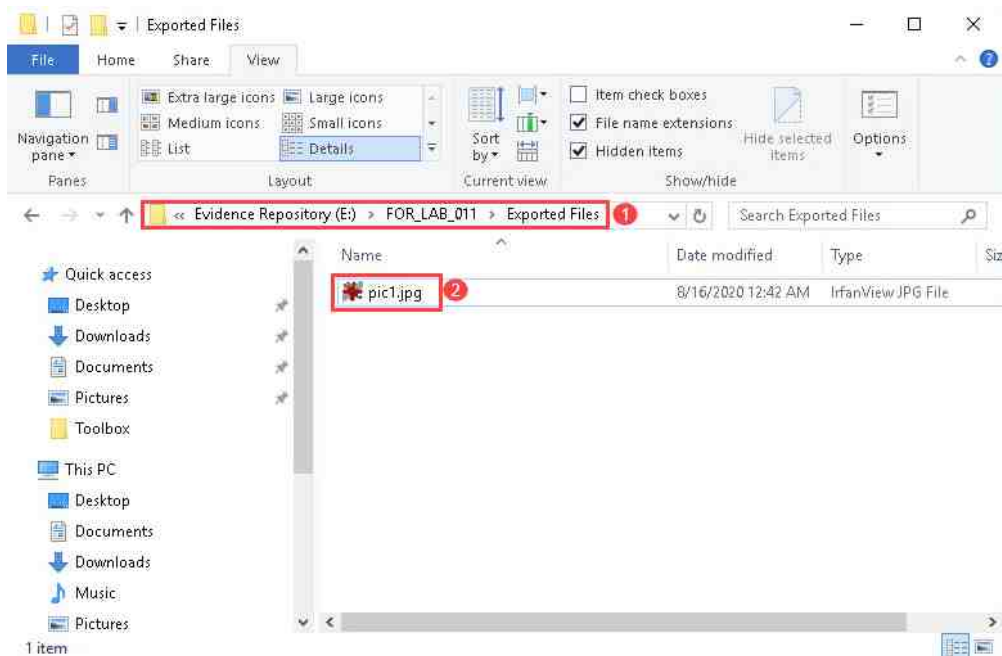
13. The Save selection as window will appear. Browse to Evidence Repository (E:) > FOR\_LAB\_011 and click New Folder to create a new folder as seen in item 1 below. Name this new folder Exported Files as seen in item 2.



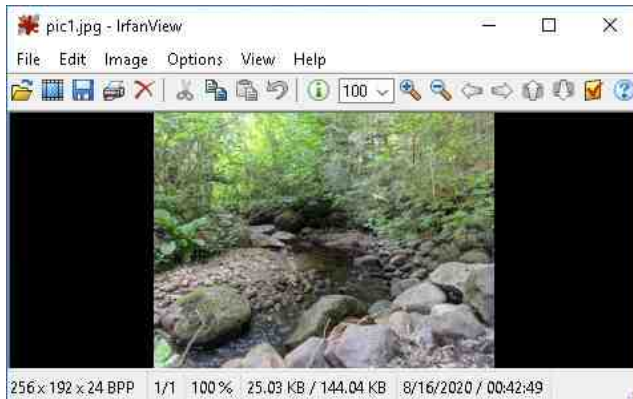
14. Double-click the folder you just created called Exported Files to open it. Now that we are in the Exported Files folder, let us save the file here. Delete any text in the File name field and type `pic1.jpg` as seen in item 1. Once you are done, click Save, as seen in item 2 below.



15. This will save the JPG file in the Exported Files folder. Let us look at the file we exported by opening File Explorer and navigating to This PC > Evidence Repository (E:) > FOR\_LAB\_011 > Exported Files and then double-clicking on the file called `pic1.jpg`, as seen in items 1 and 2 below.



16. If you did everything correctly, the file will open as you see below. This is a picture file that was embedded within the Thumbs.db file. It contains the same picture as the file that was called IMG\_001.JPG. Even if this file was deleted from the folder, the Thumbs.db file might still retain a thumbnail-sized copy of that photo.



17. Now that you have been guided through the first exercise, go back to HxD, and continue to carve JPG files until you get to the EOF. This can be done by continuing the search for headers and footers in sequential order. Export each file in the same folder and view its contents. Note the offsets that each file was taken from.

**Please Note**

When using the Find feature, pay attention to the Search direction. If you continue the search from where we left off, then the Forward option is all you need.

18. You are now at the end of Lab 11. In this lab, you learned to identify detailed metadata within picture files and how to extract thumbnails from Thumbs.db files. This knowledge will be priceless in digital investigations involving graphics files. Now that you are done, close all open programs by clicking the X at the top-right corner of each window, as seen below.

