**Lab - Using the EXIFtool to Read and Write EXIF Tags**

**Overview**

This lab will learn how to use the EXIFtool to examine the tagging and metadata information within an image file. EXIFstands for “**Exchangeable Image File Format**.”

Exiftool was developed by Phil Harvey. It is a platform-independent Perl library coupled with a full-featured command-line implementation for reading, writing, and manipulating the metadata across a broad range of files, particularly JPEG images. This metadata can include the camera make, file type, permissions, file size, etc. Exiftool further offers more details about the photograph, like the exposure, the shutter speed, and whether the flash fired or not.

Though some metadata is readily available viewing the property tab of an image, this is just a small subset of the metadata tags available under the hood of an image file. The EXIF tool is a free and open-source software program for reading, writing, and manipulating an image, audio, video, and PDF metadata.

**Lab Requirements**

In this lab, I am using **CSI Linux**. **CSI Linux** comes with Exiftool preinstalled, and I find it more conducive for conducting digital analysis. This is not to say you cannot use Kali and get the same results. I find that getting the Exiftool to run and work correctly in Kali takes some troubleshooting.

To install the Exiftool onto **Kali**, you first need to clone it from Github.

**git clone https://github.com/exiftool/exiftool.git**

In addition, you will need to install the necessary Pearl package to run it.

**sudo apt-get install libimage-exiftool-perl**

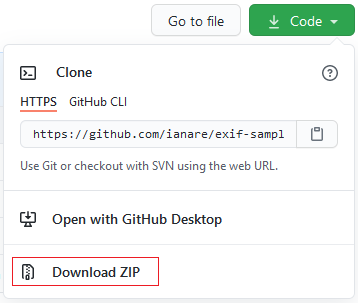
For your CSI or Kali desktop, download the following images for Exiftool to work with. Once you have the images downloaded, you can extract the archive to your desktop for this lab.

What follows is the instructions for downloading and extracting the image files from Dropbox.

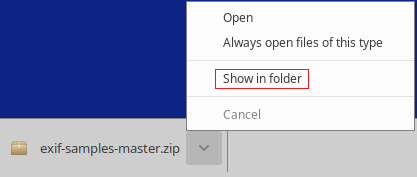
From your Kali or CSI Desktop, open a browser and paste the following URL into the address bar.

<https://github.com/ianare/exif-samples>

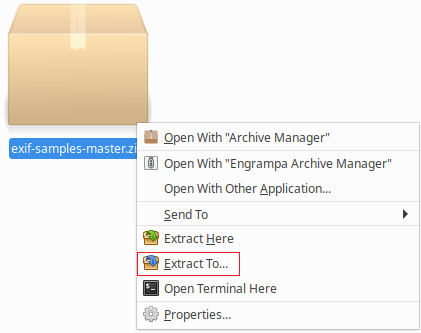
Scroll to the right side of the page and click on the green button that says Code. From the context menu, select the open to download as a zip file.



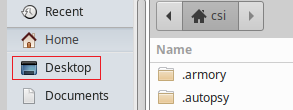
Scroll to the bottom of your browser window, find the download, click on the down arrow, and from the context menu, select, **Show in folder**.



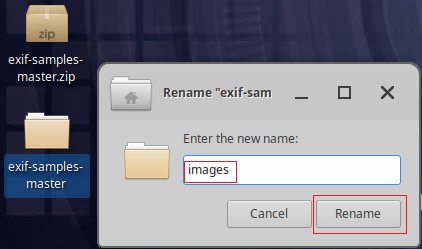
Right-click on the downloaded archive.



From the menu on the left, select the Desktop as the location to extract to.



Locate the extracted folder on your Desktop. Right-click and rename the extracted folder, **images**, all lower case.



**Begin the lab!**

From the desktop, launch a terminal. At the prompt, we need to change the directory location to where our target image is located.

Change directory location to the following location:

**cd Desktop/images/jpg/gps**

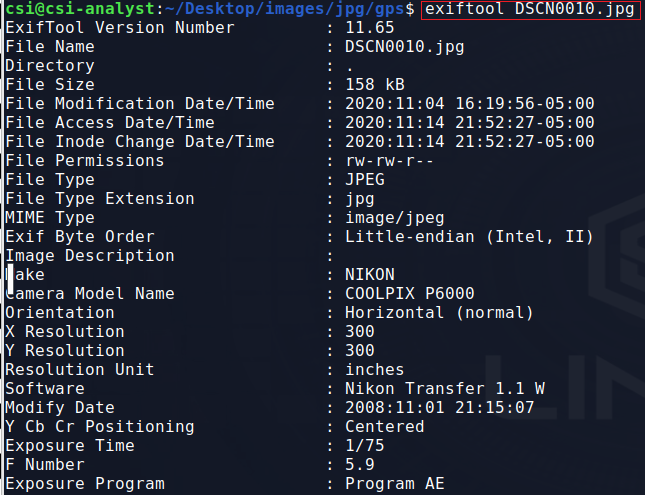
At the prompt, type **ls** to see the contents of the **gps** directory.



Copy and paste the name of the first image.

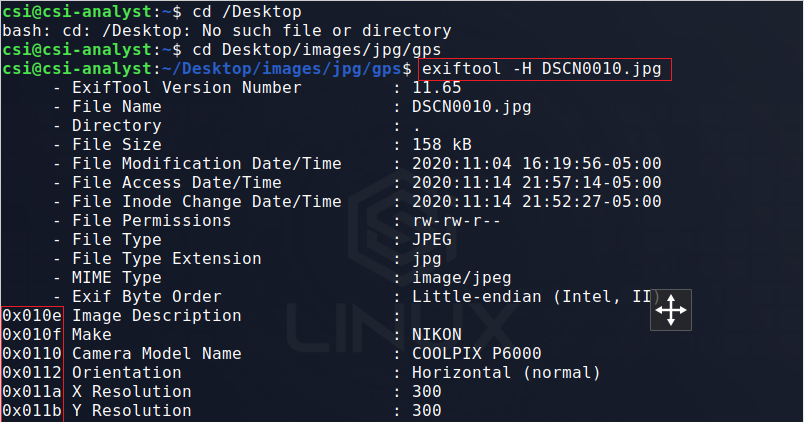
At the terminal prompt, to launch the ExifTool, type exiftool followed by the name of the image.

**exiftool DSCN0010.jpg**

****

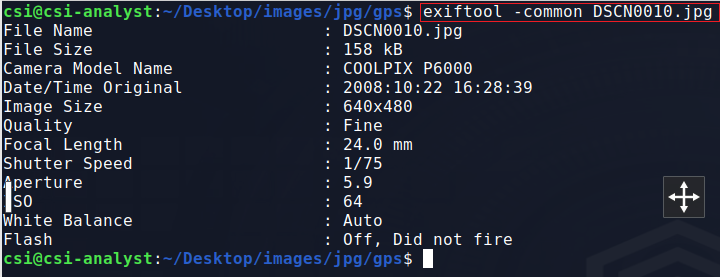
To show the IDs and exif tags in a Hexa-Decimal format, we can run the following command.

**exiftool -H DSCN0010.jpg**

****

To see just a subset of the most common Exif tags of the image file, type the following command.

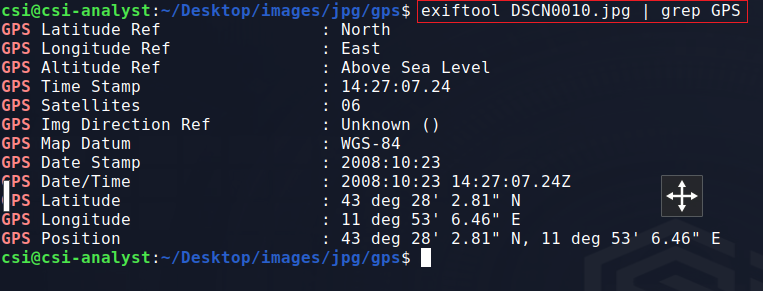
**exiftool -common DSCN0010.jpg**

****

If the image was taken using a smartphone or a camera with the GPS information embedded in the device, this to can be extracted.

To see just the GPS information, type the following command.

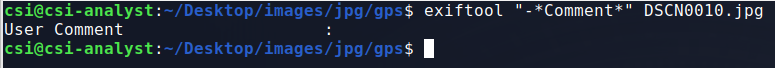
**exiftool DSCN0010.jpg | grep GPS**

****

With Exiftool, we can also conduct a keyword search to look for a particular tag being present among the metadata. In this example, I am looking for any tag with the keyword ‘comment’ as being present.

**exiftool "-\*Comment\*" DSCN0010.jpg**

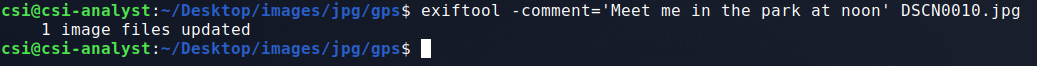
In this example, the asterisk character (\*) is being used as a wildcard in the search.



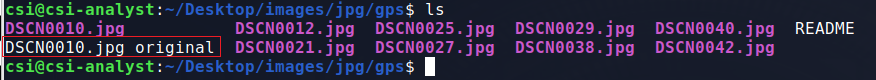
Exiftool also allows us to write to certain metadata tags that are not restricted. Restricted tags would be any tag that has anything to do with the physical characteristics of the image file, such as the compression ratio.

In this example, I added a message to the comment tag. This would be an example of someone hiding a message within the metadata of the image.

**exiftool -comment='Meet me in the park at noon’ DSCN0010.jpg**

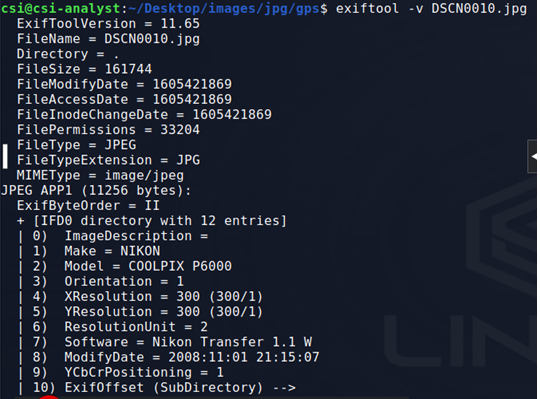
****

Anytime you modify an image’s original metadata with ExifTool, a copy of the original file is backed up.



Exiftool has a verbose feature (-v) to provide more comprehensive data about the process being performed.

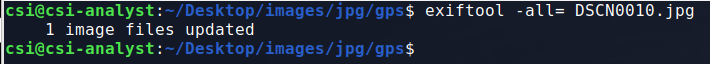
**exiftool -v DSCN0010.jpg**



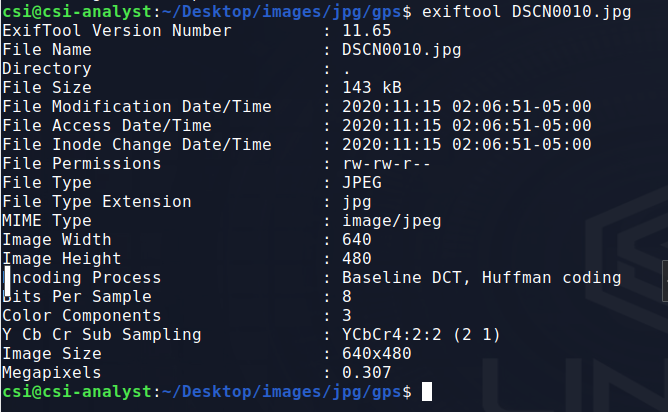
Exiftool can also remove the metadata for an image file leaving only a small subset of information.

To remove the metadata from an image, type the following command.

**exiftool -all= DSCN0010.jpg**

****

After the removal….

****