Old Dominion University
CYSE 450: Ethical Hacking and Penetration Testing

Lab 4: Steganography using Steghide

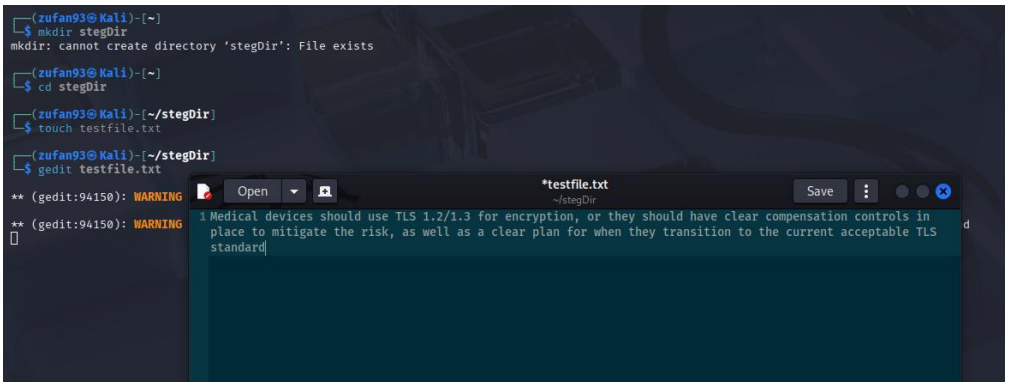
---

1. Open the terminal in Kali Linux and install ***gedit*** using the command: ***sudo apt install gedit***.

2. Create a new directory named ***stegDir*** using the ***mkdir*** command.

3. Go to the ***stegDir*** directory and create a new file named ***testfile.txt*** using the ***touch*** command.

4. Open the file ***testfile.txt*** using ***gedit*** and add some secret message there as the file content. Take a screenshot showing the secret message you added.



5. Open Firefox (in Kali Linux) and download a random image of a dog. Name the downloaded file as ***dog.jpeg***. The image will be downloaded in the ***Downloads*** folder by default.

6. Copy the image from the ***Downloads*** directory to the ***stegDir*** directory using the ***cp*** command. The ***stegDir*** directory should have two files by now: *testfile.txt* and *dog.jpeg*.

Use ***ls*** command to show the contents of the ***stegDir*** directory and take a screenshot to attach it in your submission.

7. Execute the ***md5sum*** command to check the checksums for both ***testfile.txt*** and ***dog.jpeg***. Learn about MD5 here: https://phoenixnap.com/kb/md5sum-linux). Take a screenshot similar to the following screenshot.

8. Learn about ***steghide*** command here: https://manpages.ubuntu.com/manpages/trusty/man1/steghide.1.html.

Use the ***steghide*** command to embed your ***testfile.txt*** (with secret message) into the image file **dog.jpeg** as shown in the following example screenshot (**note: *when prompted for the passphrase, you may type any password of your choice***).

Take a screenshot showing the command and the relevant output from the terminal.

9. Execute the command ***md5sum*** for ***dog.jpeg*** to check the hash for the image file. Do you see any difference? Take a screenshot showing the command and the output hash.

I had to change the file from .jpeg to .jpg because ***Steghide*** did not work with .jpeg for me. The original .jpeg file didn't work with the command, so I converted it to .jpg to make sure I could hide the message without any issues.

```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ steghide embed -cf dog.jpg -ef testfile.txt
Enter passphrase:
Re-Enter passphrase:
embedding "testfile.txt" in "dog.jpg"... done

┌──(zufan93㉿Kali)-[~/stegDir]
└─$ md5sum dog.jpg
9df3ce6fae530f4bb494ccd0398fa8b1  dog.jpg
```

I used the ***md5sum*** command to check the hash of the image file before and after I hid the message. The hash values indicated a change in the image file. Despite the picture's similar appearance, this change indicates the successful concealment of the secret message within the image.

10. Execute the ***steghide*** command to get some information about ***dog.jpeg*** before extracting it, use the ***info*** command as shown in this following example screenshot:

```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ steghide info dog.jpg
"dog.jpg":
  format: jpeg
  capacity: 3.9 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "testfile.txt":
    size: 222.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

Note that you will be asked to input the passphrase you set earlier when you embed the text file into the image. Take a screenshot showing the command and the output.

11. Now, delete the file ***testfile.txt*** using the ***rm*** command. Use the ***ls*** command to show the contents of the ***stegDir*** directory and take a screenshot.

```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ rm testfile.txt

┌──(zufan93㉿Kali)-[~/stegDir]
└─$ ls
dog.jpeg  dog.jpg  testfile.text
```

12. Extract the secret message by executing the ***steghide*** command with **- -** ***extract*** option as shown in the following example screenshot:

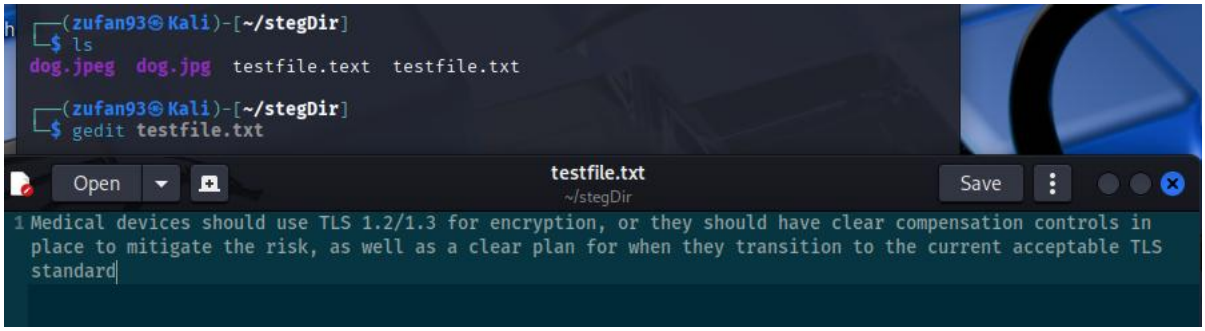
```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ steghide extract -sf dog.jpg
Enter passphrase:
wrote extracted data to "testfile.txt".
```

Take a screenshot showing the command and the output in the terminal.

13. Execute the ***ls*** command to list the contents in the ***stegDir*** directory. You should see ***testfile.txt*** there because it was hidden in the ***dog.jpeg*** image file and appeared after extracting the image file in the previous step (step-12). Take a screenshotn showing the contents of the ***stegDir*** directory.

```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ ls
dog.jpeg  dog.jpg  testfile.text  testfile.txt
```

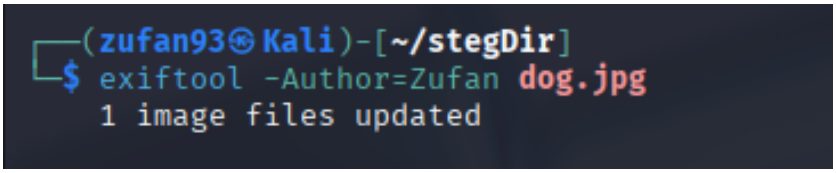
14. See the contents of the file ***testfile.txt*** using ***gedit***. Take a screenshot showing the contents.



15. See the metadata of the file ***dog.jpeg*** using the ***exiftool*** command as shown in the following example screenshot:

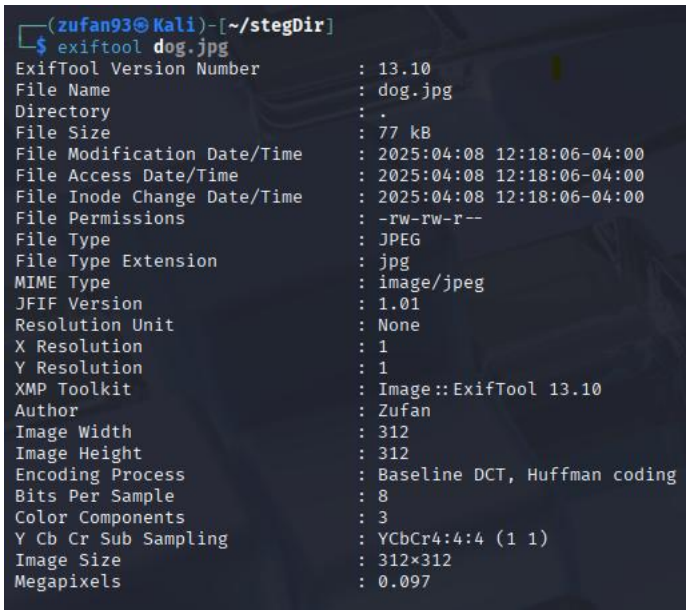
```
┌──(zufan93㉿Kali)-[~/stegDir]
└─$ exiftool dog.jpg
ExifTool Version Number         : 13.10
File Name                       : dog.jpg
Directory                       : .
File Size                       : 74 kB
File Modification Date/Time     : 2025:04:08 11:51:12-04:00
File Access Date/Time           : 2025:04:08 11:55:31-04:00
File Inode Change Date/Time     : 2025:04:08 11:51:12-04:00
File Permissions                : -rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Image Width                     : 312
Image Height                    : 312
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:4:4 (1 1)
Image Size                      : 312×312
Megapixels                      : 0.097
```

16. Change the author of the file ***dog.jpeg*** using the ***exiftool*** command as shown in the following example screenshot:



Note: when you enter the *exiftool* command in the terminal to update the author's name, make sure you replace "Alice" with your own name.

17. Repeat the step-15 and take a screenshot showing the updated metadata of the file ***dog.jpeg***. Highlight the author's name in the screenshot.



18. Execute the **md5sum** command for ***dog.jpeg***. Do you see any change in the hash value? If yes, take a screenshot of the new hash and compare it with the previous hash you received in step-9.

The hash values are different. That means the file changed even though the picture still looks the same. This data tells us the secret message was successfully hidden inside the image.