

Lab 4: Steganography using Steghide

Handout Date: March 27, 2025

Due Date: April 04, 2025, 11:59 pm

Total Points: 30

Tasks

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***](https://phoenixnap.com/kb/md5sum-linux). **Take a screenshot similar to the following screenshot.**

```
inet6 2000:8805:1917:3d00:ad0e:5a49:4d99:0e04  pre
└─(kali㉿kali)-[~/stegDir] 7:3d00 :: 7551  prefixlen 128  sc
$ ls  inet6 fe80::78b1:f3ff:fe8b:ac7a  prefixlen 64  sc
dog.jpeg  testfile.txt 5:1917:3d00:78b1:f3ff:fe8b:ac7a  pre
      ether 7a:b1:f3:8b:ac:7a  txqueuelen 1000  (Etherne
└─(kali㉿kali)-[~/stegDir]  es 79202 (77.3 KiB)
$ md5sum dog.jpeg  dropped 0  overruns 0  frame 0
64387b1f6a7739dc1ae20a3d45f082e921dog.jpeg 2 KiB
      TX errors 0  dropped 0  overruns 0  carrier 0  coll
└─(kali㉿kali)-[~/stegDir]
$ md5sum testfile.txt  K, RUNNING>  mtu 65536
e37ee3de304967eae5c4231b551e5d8025testfile.txt
      scopeid 0x1  broadcast 128  scopeid 0x10<host>
```

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**).

```
inet6 ::1  prefixlen 128  scopeId 0x10<host>
└─(kali㉿kali)-[~/stegDir] 00  (Local Loopback)
$ steghide embed -cf dog.jpeg -ef testfile.txt
Enter passphrase: 0  dropped 0  overruns 0  frame 0
Re-Enter passphrase: 8  bytes 2596 (2.5 KiB)
embedding "testfile.txt" in "dog.jpeg"... done 0  co
```

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.

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:

```
└─(kali㉿kali)-[~/stegDir] ff:ff:ff 0806 42: arp reply 192.168.1.11 → 192.168.1.11 (eth0)
└─$ steghide info dog.jpeg
"dog.jpeg": ac:7a ff:ff:ff:ff:ff:ff 0806 42: arp reply 192.168.1.11 → 192.168.1.11 (eth0)
format: jpeg
capacity: 88.3 kB
Try to get information about embedded data? (y/n) y
Enter passphrase:
embedded file "testfile.txt":
└─# size: 30.0 Byte
? (1) encrypted: rijndael-128,cbc:3b:84 [ether] on eth0
? (1) 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.

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

```
└─(kali㉿kali)-[~/stegDir] 7:3d00::7551
└─$ steghide --extract -sf dog.jpeg
Enter passphrase: 0:8805:1917:3d00:78b1:f
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 screenshot showing the contents of the **stegDir** directory.

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:

```
└─(kali㉿kali)-[~/stegDir]└ 0 overruns 0 carrier 0 collisions 0
└─$ exiftool dog.jpeg
ExifToolsVersion Number: 12.7665536
File Name: net 127.0.0.1 netmask :5dog.jpeg
Directory: net6 ::1 prefixlen 128: scopeid 0x10<host>
File Size: 0p txqueuelen 1000 (:01369kB backlog)
File Modification Date/Times: 259: 2024:10:24 14:38:44-04:00
File Access Date/Time: dropped 0 :v 2024:10:24 14:39:22-04:00
File Inode Changes Date/Time: 259: 2024:10:24 14:38:44-04:00
File Permissions: 0 dropped 0 ote-rw-rw-r-- carrier 0 collisions 0
File Type: : JPEG
File Type Extension: : jpg
MIME Type: [kali] : image/jpeg
JFIF Version: -i eth0 192.168.0.1: 1.02
Resolution Unit: 7a ff:ff:ff:ff:ff:ff inches 42: arp reply 192.168.0.1
X Resolution: 7a ff:ff:ff:ff:ff:ff 720806 42: arp reply 192.168.0.1
Y Resolution: 7a ff:ff:ff:ff:ff:ff 720806 42: arp reply 192.168.0.1
Image Width: 7a ff:ff:ff:ff:ff:ff 3000 06 42: arp reply 192.168.0.1
Image Height: 7a ff:ff:ff:ff:ff:ff 4206 06 42: arp reply 192.168.0.1
Encoding Process: re-arping targ:t Baseline DCT, Huffman coding
Bits Per Sample: : 8
Color Components: [kali] : 3
Y Cb Cr Sub Sampling: : YCbCr4:2:0 (2 2)
Image Size: 0.239) at d2:34:b4:ca:3000x4206 er] on eth0
Megapixels: 0.1) at 44:1c:12:f4:e::12.6 ether] on eth0
```

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

```
└─(kali㉿kali)-[~/stegDir]
└─$ exiftool -author=Alice dog.jpeg:84
? (12 image) files updated 12:f4:e4:74 [e]
```

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.

Turn-in

- Attach all the screenshots in your submission.