Cybersecurity 701

Steganography Lab

Steganography Materials

- Materials needed
  - Kali Linux Machine

- Software Tools used
  - **`zip`** command (Linux command)
  - **`steghide`** (Linux program)

Objectives Covered

- Security+ Objectives (SY0-701)
  - Objective 1.4 – Explain the importance of using appropriate cryptographic solutions.
    - Obfuscation
      - Steganography

What is Steganography?

- A type of obfuscation where the actual message does not attract attention
  - A message is embedded in an image
  - TCP packets have message embedded inside them
  - Watermarks that are invisible to the eye on the paper

This image has the message “You may be compromised, meet your handler at headquarters” embedded in it.

Steganography Lab Overview

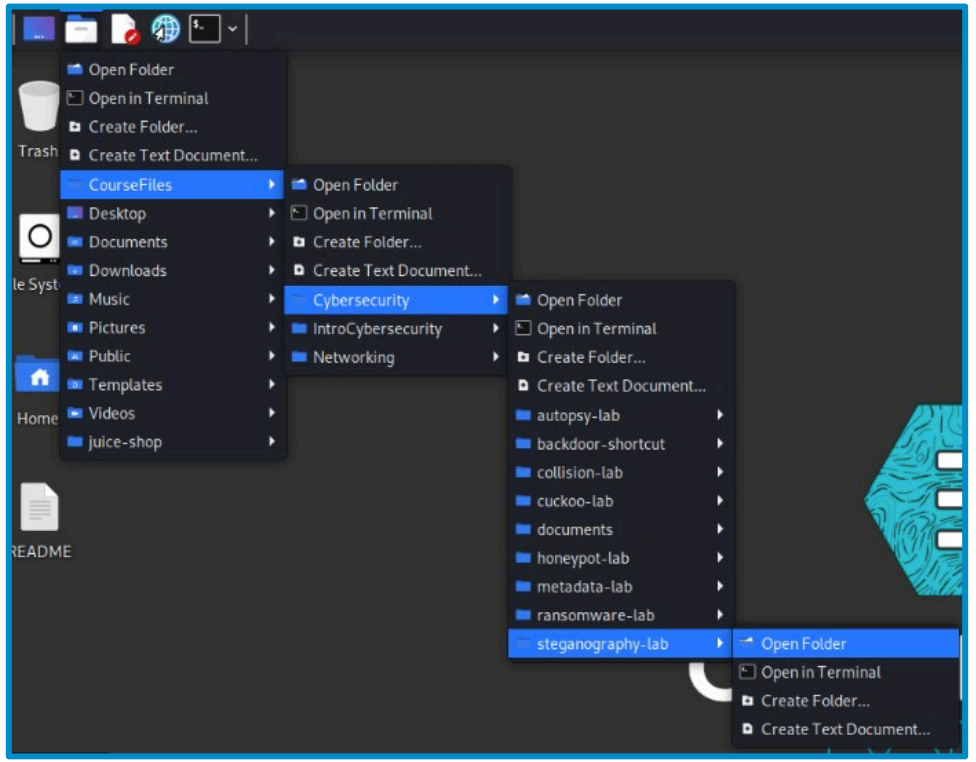
1. Set up Environment
2. Embedded text (zip command)
3. Embedded text (steghide app)

Set up Environment

- Log into your range
- Open the Kali Linux Environment
  - You should be on your Kali Linux Desktop

Find Steganography Lab Files

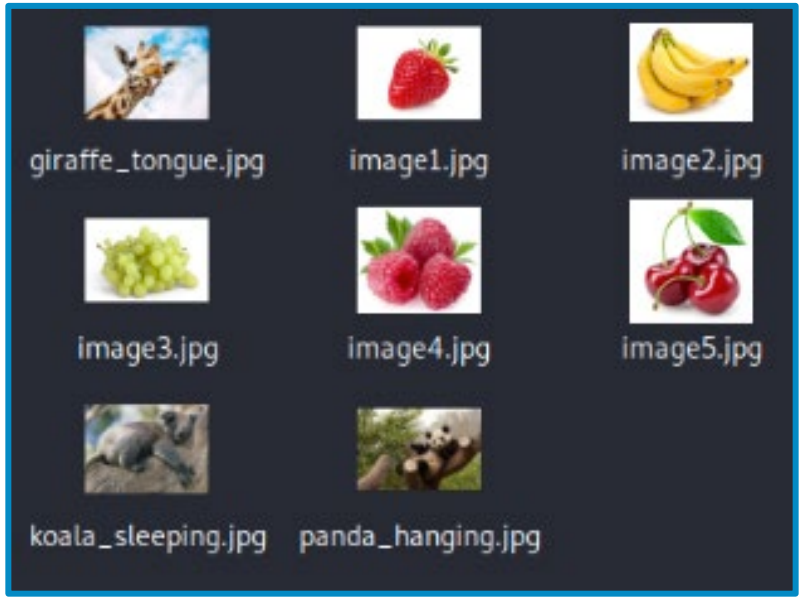
- Click the Folder icon in the top left of Kali
- Then navigate to CourseFiles > Cybersecurity > steganography-lab and select "Open Folder"



Embedded Text (`zip` Command)

Look at the images inside of this folder

- Open the images to see if you see any hidden messages
  - 3 of the images have hidden messages



Embedded Text (`zip` Command)

Let's discover the text hidden inside of the panda_hanging.jpg image

- Open a terminal
- Navigate to the folder

  `cd CourseFiles/Cybersecurity/steganography-lab`

- When inside the folder, unzip the image

  `unzip panda_hanging.jpg`

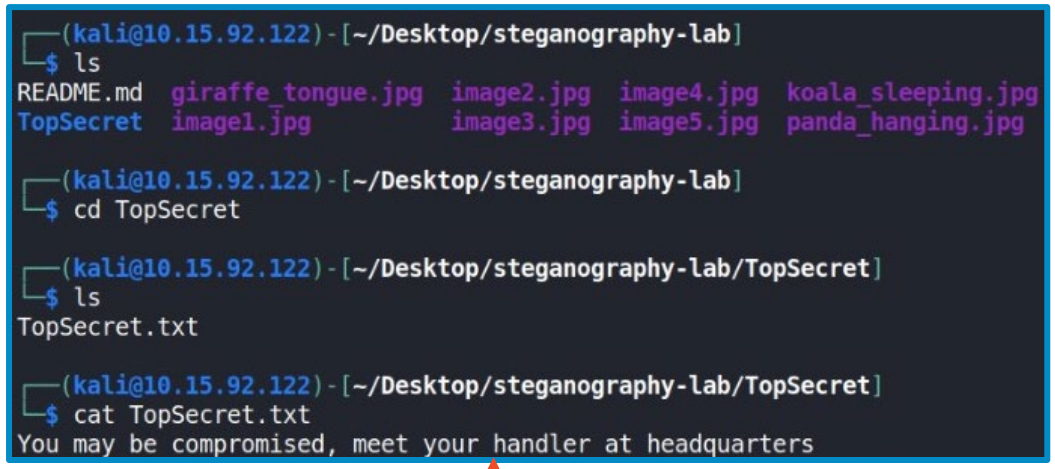
```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ unzip panda_hanging.jpg
Archive:  panda_hanging.jpg
warning [panda_hanging.jpg]:  354699 extra bytes at beginning or within zip
file
  (attempting to process anyway)
   creating: TopSecret/
  inflating: TopSecret/TopSecret.txt
```

Notice that a TopSecret.txt file was extracted from this image

Embedded Text (`zip` Command)

What does the TopSecret.txt file say?

- List all the files in the folder
  `ls`
- Notice there is a TopSecret folder
- Navigate into this folder and list all the files
  `cd TopSecret`
  `ls`
- Read the TopSecret.txt file
  `cat TopSecret.txt`



Notice the hidden message that was embedded in the image

Embedded Text (`zip` Command)

Your turn to hide a message in an image just like this one.
Create the message here:

- Navigate back to steganography-lab folder

  `cd ..`

- Create and edit the text document

  `nano SekretMessage.txt`

- Enter your secret message to be hidden
- Press **`CTRL+X`**, then **`Y`**, and **`ENTER`** to save

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ nano SekretMessage.txt
```

Embedded Text (`zip` Command)

Now, put the message in a zipped folder:

- Create a directory:
  `mkdir SekretMessage`
- Move the message into the directory
  `mv SekretMessage.txt SekretMessage`
- Zip the directory
  `zip -r SekretMessage.zip SekretMessage`
- Remove the old directory
  `rm SekretMessage -rf`

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ mkdir SekretMessage

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ mv SekretMessage.txt SekretMessage

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ zip -r SekretMessage.zip SekretMessage
  adding: SekretMessage/ (stored 0%)
  adding: SekretMessage/SekretMessage.txt (stored 0%)

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ rm SekretMessage -rf
```

Embedded Text (`zip` Command)

Concatenate the zipped file with **`image1`**:

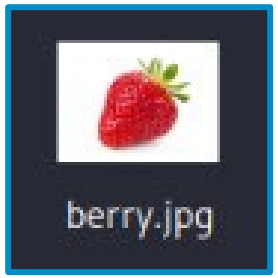
- Combine the files

```
cat image1.jpg SekretMessage.zip > berry.jpg
```

- Delete the old message
  - **`rm SekretMessage.zip`**

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ cat image1.jpg SekretMessage.zip > berry.jpg

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ rm SekretMessage.zip
```

Navigate to the **`berry.jpg`** file, notice it's just an image of a strawberry…



…or is it?

Embedded Text (`zip` Command)

Without unzipping the image, try to find the embedded message inside the **`berry.jpg`** image:

- Now, unzip the file and find the hidden message

```
unzip berry.jpg
```

- Display the message
  - **`cat SekretMessage/SekretMessage.txt`**

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ unzip berry.jpg
Archive:  berry.jpg
warning [berry.jpg]:  94376 extra bytes at beginning or within zipfile
  (attempting to process anyway)
   creating: SekretMessage/
 extracting: SekretMessage/SekretMessage.txt

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ cat SekretMessage/SekretMessage.txt
This is my super sekret message!!
```

Embedded Text (`zip` Command) – Your Turn

Your turn: Hide your own message

- Create a text file (hidden message!)
- Put the text file into a folder
- Zip the folder (*Recursively*!)
- Concatenate the zipped folder and **`image2.jpg`** to hide the message
- Delete the old files (except new image with hidden message)
- Then have someone else discover the hidden message!

Embedded Text (steghide app)

- Once **`steghide`** is installed, discover the hidden message inside of the giraffe image:

  `steghide extract -sf giraffe_tongue.jpg`

- When prompted for a passphrase:
  - Use "**verytall**" (without quotations)
- Display the hidden message

  `cat HiddenMessage.txt`

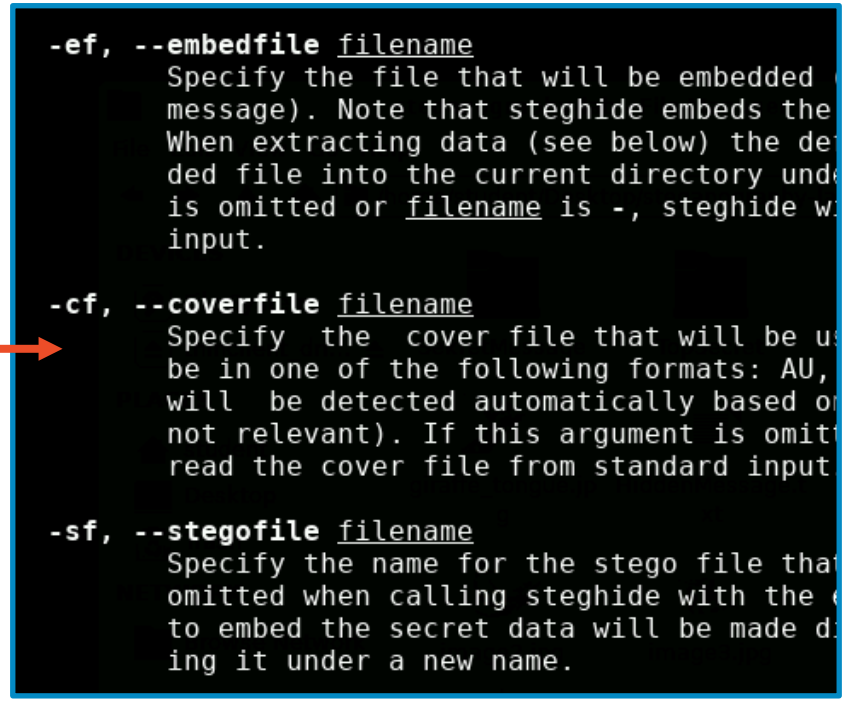
```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ steghide extract -sf giraffe_tongue.jpg
Enter passphrase:
wrote extracted data to "HiddenMessage.txt".

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ cat HiddenMessage.txt
Password for the koala_sleeping.jpg is EuCaLyPtUs
```

Notice that this gives the passphrase for the koala image. Find the hidden message embedded in the koala image

Embedded Text (steghide app)

- Let's read the manual for **`steghide`**:
  - **`man steghide`**
    - Locate the following flags/options:
      - **`-sf`**
      - **`-ef`**
      - **`-cf`**

What are the purpose of these flags/options?



Embedded Text (steghide app)

- Create and edit the message

  `nano MessageToHide.txt`

- Enter your secret message to be hidden
- Press **`CTRL+X`**, then **`Y`**, and **`ENTER`** to save

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ nano MessageToHide.txt
```

Embedded Text (steghide app)

- Embed the message inside of **`image3.jpg`**

```
steghide embed -ef MessageToHide.txt -cf image3.jpg
```

- Enter a password when prompted
- Re-enter to confirm the password
- Remove the old message

```
rm MessageToHide.txt
```

```
┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ steghide embed -ef MessageToHide.txt -cf image3.jpg
Enter passphrase:
Re-Enter passphrase:
embedding "MessageToHide.txt" in "image3.jpg"... done

┌──(kali@10.15.73.236)-[~/CourseFiles/Cybersecurity/steganography-lab]
└─$ rm MessageToHide.txt
```

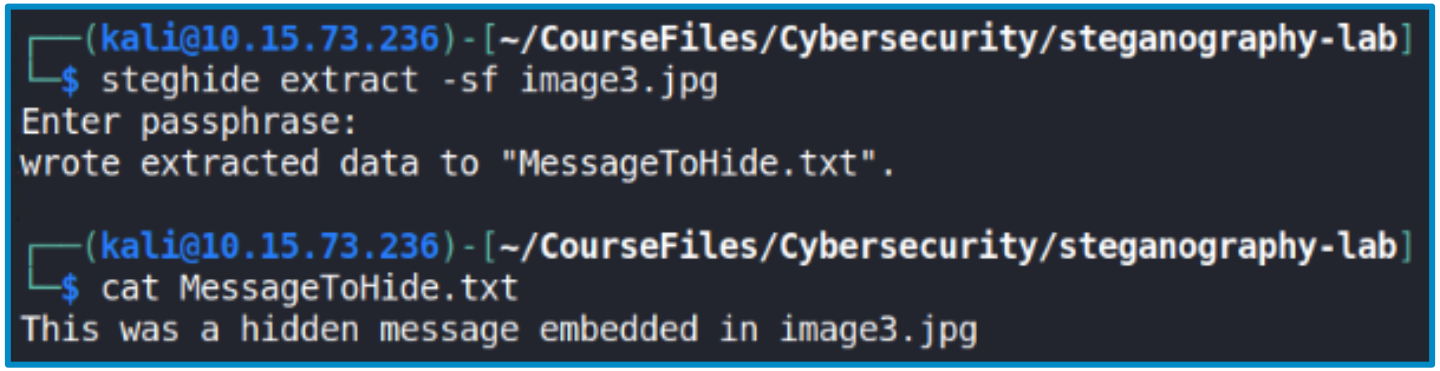
Embedded Text (steghide app)

- Try to find the embedded text without **`steghide`**
- Use **`steghide`** to find the message

```
steghide extract -sf image3.jpg
```

- Enter password when prompted
- Display the message

```
cat MessageToHide.txt
```



Embedded Text (steghide app) – Your Turn

Your turn: Hide your own message

- Create a text file (hidden message!)
- Hide the text file inside **`image4.jpg`**
- Remember the password!
- Delete the old message
- Have someone else try to discover the hidden message!