



Security+ Lab Series

Lab 19: Cryptography Concepts

Document Version: 2018-08-28

Copyright © 2018 Network Development Group, Inc.
www.netdevgroup.com

NETLAB Academy Edition, NETLAB Professional Edition, NETLAB+ Virtual Edition, and NETLAB+ are registered trademarks of Network Development Group, Inc.

Contents

Introduction	3
Objectives.....	3
Lab Topology	4
Lab Settings.....	5
1 Hiding a Hidden Message Within a Picture	6
1.1 Using Steghide to Hide Hidden Messages.....	6
2 Hiding Multiple Files Within an Image File	10
2.1 Using Basic Linux Commands to Hide Zipped Archives.....	10
3 Hiding a Text File Within an Audio File	12
3.1 Using SteGUI to Hide Text Files.....	12

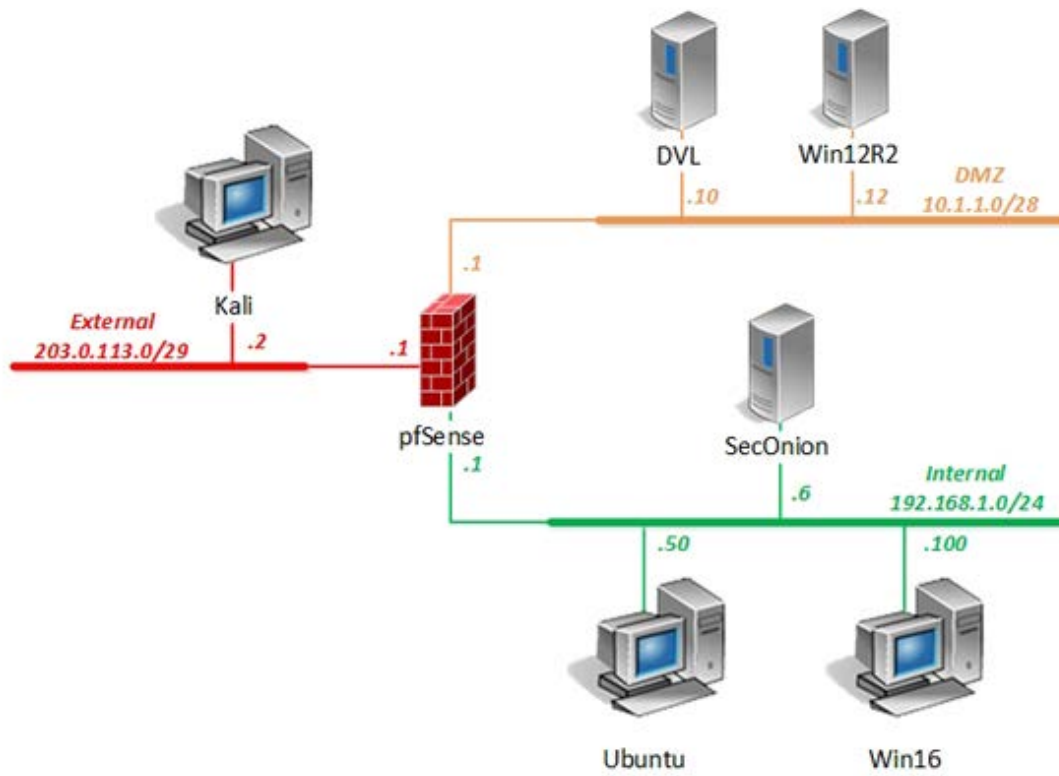
Introduction

In this lab, you will be conducting steganography techniques by using various tools.

Objectives

- Given a scenario, use appropriate software tools to assess the security posture of an organization

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	Account	Password
DVL	10. 1. 1. 10 /28	root	toor
Kali	203. 0. 113. 2 /29	root	toor
pfSense	eth0: 192. 168. 1. 1 /24 eth1: 10. 1. 1. 1 /28 eth2: 203. 0. 113. 1 /29	admin	pfsense
Sec0nion	192. 168. 1. 6 /24	soadmin	mypassword
		root	mypassword
Ubuntu	192. 168. 1. 50 /24	student	securepassword
		root	securepassword
Win12R2	10. 1. 1. 12 /28	administrator	Train1ng\$
Win16	192. 168. 1. 100 /24	lab-user	Train1ng\$
		Administrator	Train1ng\$

1 Hiding a Hidden Message Within a Picture

1.1 Using Steghide to Hide Hidden Messages

1. Launch the **Kali** virtual machine to access the graphical login screen.
2. Log in as **root** with **toor** as the password. Open the **Kali PC Viewer**.
3. Click on the **terminal** icon located in the top menu bar.



4. While in the *terminal*, navigate to the **/tmp/random** directory.

```
root@Kali-Attacker:~# cd /tmp/random
```

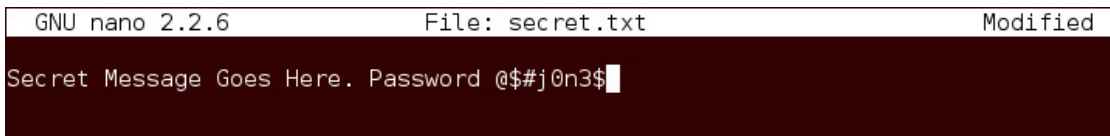
```
root@Kali-Attacker:~# cd /tmp/random
root@Kali-Attacker:/tmp/random#
```

5. Create a new text document with the *nano* text editor to create a message to hide.

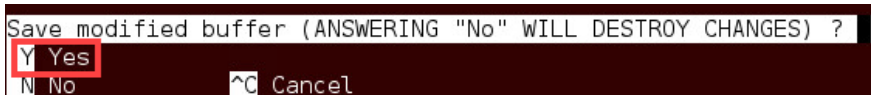
```
root@Kali-Attacker:/tmp/random# nano secret.txt
```

6. Once engaged in the *nano* text editor, type the message below.

```
Secret Message Goes Here. Password: @$#j0n3$
```



7. Once finished, press **CTRL+X** to exit and save.
8. When prompted to save the file, type **Y** for Yes.



9. When prompted for the filename, verify **secret.txt** as the file name and press **Enter**.



10. Verify the capacity of the *earth.jpg* image to see what the capacity is for being able to hide a message within the image itself.

```
root@Kali-Attacker:/tmp/random# steghide info earth.jpg
```

11. Notice the capacity amount. When asked to get more information about embedded data, type **N**.

```
root@Kali-Attacker:/tmp/random# steghide info earth.jpg
"earth.jpg":
  format: jpeg
  capacity: 5.5 KB
Try to get information about embedded data ? (y/n) n
root@Kali-Attacker:/tmp/random#
```

12. See how large the *secret.txt* file is to confirm whether we can hide it within the *earth.jpg* image. Use the **-b** option to display the file size in terms of bytes.

```
root@Kali-Attacker:/tmp/random# du -b secret.txt
```

```
root@Kali-Attacker:/tmp/random# du -b secret.txt
44      secret.txt
root@Kali-Attacker:/tmp/random#
```



Notice that we should be able to fit the 44 *byte* sized *secret.txt* file in the 5.5 *KB* *earth.jpg* image file.

13. Before we embed the *secret* message, confirm the *sha1 hash* value for the *earth.jpg* image file.

```
root@Kali-Attacker:/tmp/random# shasum earth.jpg
```

```
root@Kali-Attacker:/tmp/random# shasum earth.jpg
cebb4eba5adc74ea2f8d3fb6598b2b79bb3e82ab  earth.jpg
root@Kali-Attacker:/tmp/random#
```



Take note of this hash value for later comparison.

14. Type the command below to initialize the process of hiding the secret message. When prompted for a passphrase, type **secret** followed by pressing **Enter**. Type **secret** once more. Press **Enter**.

```
root@Kali-Attacker:/tmp/random# steghide embed -cf earth.jpg -ef secret.txt
```

```
root@Kali-Attacker:/tmp/random# steghide embed -cf earth.jpg -ef secret.txt
Enter passphrase:
Re-Enter passphrase:
embedding "secret.txt" in "earth.jpg"... done
root@Kali-Attacker:/tmp/random#
```

15. Verify the *hash value* again with the same *earth.jpg* image file.

```
root@Kali-Attacker:/tmp/random# shasum earth.jpg
```

```
root@Kali-Attacker:/tmp/random# shasum earth.jpg
1ef6071d85cd2alc8b63106214658885843eccba  earth.jpg
root@Kali-Attacker:/tmp/random#
```



Notice the integrity has been lost in the steganography process due to a different hash value.

16. Type the command below to gather info on the embedded data within the **earth.jpg** file. When asked to get information about embedded data, type **Y**. Type **secret** as the passphrase. Press **Enter**.

```
root@Kali-Attacker:/tmp/random# steghide info earth.jpg
```

```
root@Kali-Attacker:/tmp/random# steghide info earth.jpg
"earth.jpg":
  format: jpeg
  capacity: 5.5 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "secret.txt":
    size: 44.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
root@Kali-Attacker:/tmp/random#
```



Notice the output, highlighting that there is a *secret.txt* file present within the *earth.jpg* file.

17. Attempt to extract the secret data within the *earth.jpg* image file. When prompted for the passphrase, type **secret** followed by pressing **Enter**. When prompted that the file *secret.txt* already exists, type **Y** to overwrite.

```
root@Kali-Attacker:/tmp/random# steghide extract -sf earth.jpg
```

```
root@Kali-Attacker:/tmp/random# steghide extract -sf earth.jpg
Enter passphrase:
the file "secret.txt" does already exist. overwrite ? (y/n) y
wrote extracted data to "secret.txt".
root@Kali-Attacker:/tmp/random#
```


18. Confirm that the secret message has been preserved by viewing the contents.

```
root@Kali-Attacker:/tmp/random# cat secret.txt
```

```
root@Kali-Attacker:/tmp/random# cat secret.txt
Secret Message Goes Here. Password @$#j0n3$
root@Kali-Attacker:/tmp/random#
```

19. Leave the *terminal* window open to continue with the next task.

2 Hiding Multiple Files Within an Image File

2.1 Using Basic Linux Commands to Hide Zipped Archives

1. While still on the *Kali* system, in the terminal window, verify that you are currently in the **/tmp/random** directory. List the files in the current directory using the command below. Take note of the file sizes for both *cybersec.jpg* and *secret.txt*.

```
root@Kali-Attacker:/tmp/random# ls -lh
```

```
root@Kali-Attacker:/tmp/random# ls -lh
total 536K
-rw-r--r-- 1 root root 33K Apr 10 2015 bird.wav
-rw-r--r-- 1 root root 136K Apr 10 2015 cybersec.jpg
-rw-r--r-- 1 root root 198K Apr 10 2015 dc.jpg
-rw-r--r-- 1 root root 115K Aug 15 10:32 earth.jpg
-rw-r--r-- 1 root root 43K Apr 10 2015 penguin.jpg
-rw-r--r-- 1 root root 44 Aug 15 10:39 secret.txt
root@Kali-Attacker:/tmp/random#
```

2. Create a *zipped* archive named **secret_files** to include the files: **secret.txt** and **dc.jpg**.

```
root@Kali-Attacker:/tmp/random# zip secret_files secret.txt dc.jpg
```

```
root@Kali-Attacker:/tmp/random# zip secret_files secret.txt dc.jpg
adding: secret.txt (stored 0%)
adding: dc.jpg (deflated 0%)
root@Kali-Attacker:/tmp/random#
```

3. List the current files in the directory to verify that *secret_files.zip* is present.

```
root@Kali-Attacker:/tmp/random# ls -l
```

```
root@Kali-Attacker:/tmp/random# ls -l
total 736
-rw-r--r-- 1 root root 33580 Apr 10 2015 bird.wav
-rw-r--r-- 1 root root 138713 Apr 10 2015 cybersec.jpg
-rw-r--r-- 1 root root 201816 Apr 10 2015 dc.jpg
-rw-r--r-- 1 root root 117671 Aug 15 10:32 earth.jpg
-rw-r--r-- 1 root root 43211 Apr 10 2015 penguin.jpg
-rw-r--r-- 1 root root 201989 Aug 15 11:07 secret_files.zip
-rw-r--r-- 1 root root 44 Aug 15 10:39 secret.txt
root@Kali-Attacker:/tmp/random#
```

4. Using the **cat** command, enter the command below to hide the zipped archive within the image file called **cybersec.jpg**. This will enable the *cat* command to concatenate the image and zip file together in a new file named **cyber.jpg**.

```
root@Kali-Attacker:/tmp/random# cat cybersec.jpg secret_files.zip > cyber.jpg
```

```
root@Kali-Attacker:/tmp/random# cat cybersec.jpg secret_files.zip > cyber.jpg
root@Kali-Attacker:/tmp/random#
```

5. List the files in the current directory to verify that the **cyber.jpg** image file has been successfully created. Also, take note of the file size and compare it to **cybersec.jpg**.

```
root@Kali-Attacker:/tmp/random# ls -lh
```

```
root@Kali-Attacker:/tmp/random# ls -lh
total 1.2M
-rw-r--r-- 1 root root 33K Apr 10 2015 bird.wav
-rw-r--r-- 1 root root 395K Aug 15 11:09 cyber.jpg
-rw-r--r-- 1 root root 198K Aug 15 11:08 cybersec.jpg
-rw-r--r-- 1 root root 198K Apr 10 2015 dc.jpg
-rw-r--r-- 1 root root 115K Aug 15 10:32 earth.jpg
-rw-r--r-- 1 root root 43K Apr 10 2015 penguin.jpg
-rw-r--r-- 1 root root 198K Aug 15 11:07 secret_files.zip
-rw-r--r-- 1 root root 44 Aug 15 10:39 secret.txt
root@Kali-Attacker:/tmp/random#
```

6. Initiate the **unzip** command against the **cyber.jpg** image file in an attempt to extract the data hidden within the image file.

```
root@Kali-Attacker:/tmp/random# unzip -t cyber.jpg
```

```
root@Kali-Attacker:/tmp/random# unzip -t cyber.jpg
Archive: cyber.jpg
warning [cyber.jpg]: 201989 extra bytes at beginning or within zipfile
(attempting to process anyway)
  testing: secret.txt          OK
  testing: dc.jpg             OK
No errors detected in compressed data of cyber.jpg.
root@Kali-Attacker:/tmp/random#
```



Notice the two file names displayed in the output. This confirms that the files have been successfully concatenated to the image file.

7. Leave the *terminal* window open to continue with the next task.

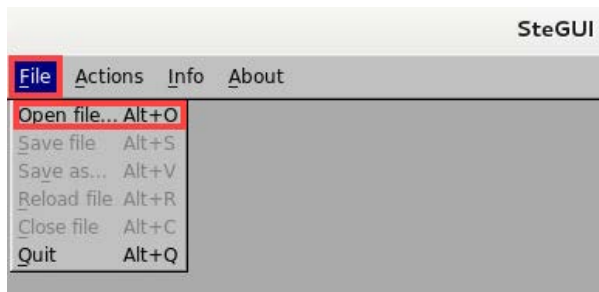
3 Hiding a Text File Within an Audio File

3.1 Using SteGUI to Hide Text Files

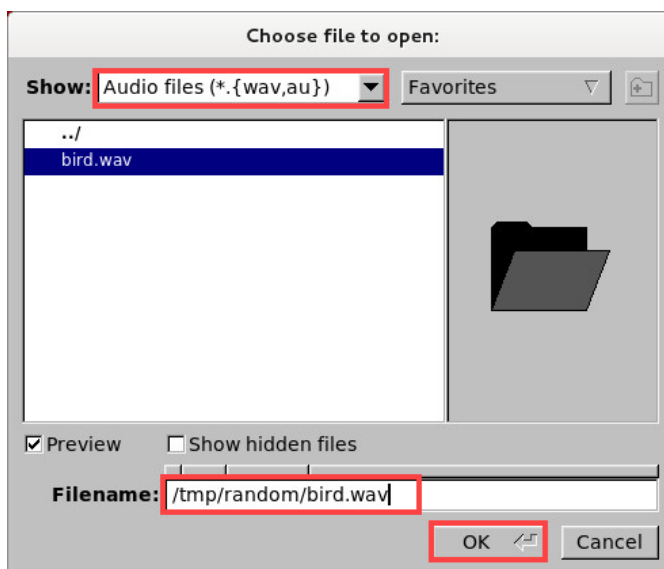
1. While still on the *Kali* system, type the command below in a *terminal* shell to launch the *SteGUI* application. (*SteGUI* is case-sensitive)

```
root@Kali-Attacker:/tmp/random# /opt/stegui/bin/SteGUI
```

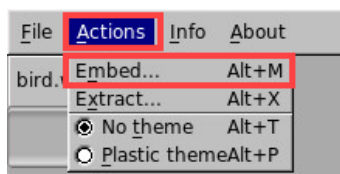
2. Notice the *SteGUI* application opens. Click on the **File** menu option and select **Open file**.



3. Within the *Choose file to open* window, select the **drop-down menu** next to *Show:* and select **Audio files**. Towards the bottom of the window, notice the *white space* after *Filename:*. Type **/tmp/random/bird.wav**. Select the audio file and click **OK**.



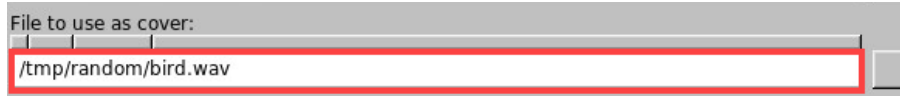
4. Next, click the **Actions** file menu option and click on **Embed**.



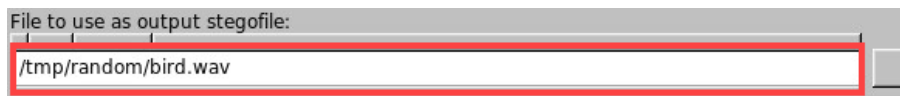
- Another pop-up window appears. Click the **white space** area for *File to embed in cover file* and type **/tmp/random/secret.txt**.



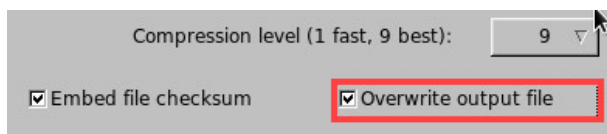
- Verify that **/tmp/random/bird.wav** is the selected file for *File to use as cover*.



- For *File to use as output stegofile*, click in the white space and type **/tmp/random/bird.wav**.



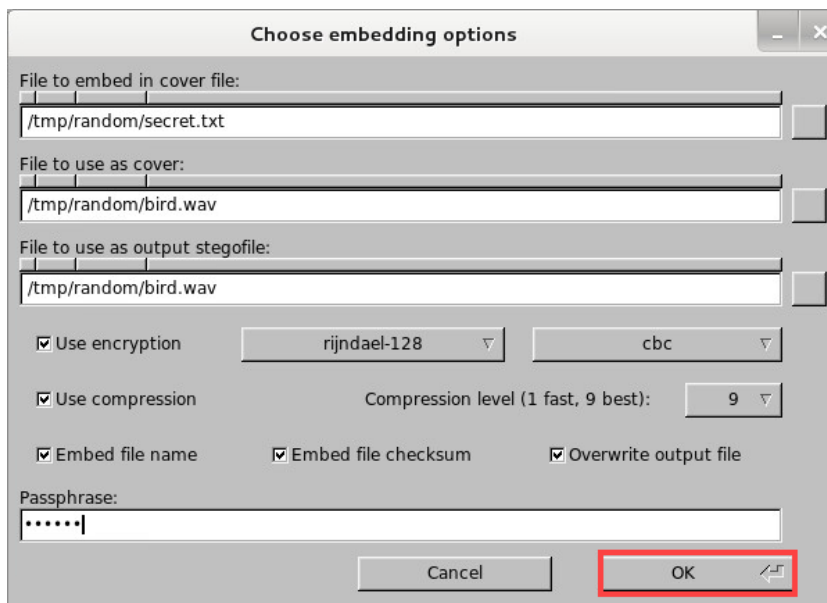
- Check the checkbox for **Overwrite output file**.



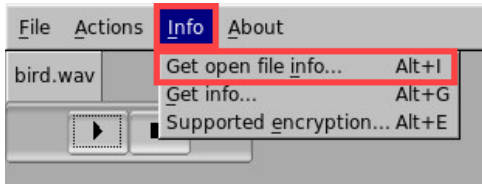
- Type **secret** as the *Passphrase*.



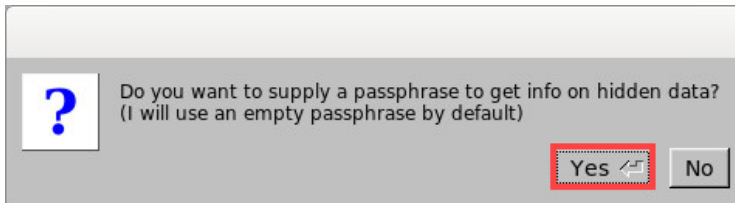
- Leave the remaining fields as **defaults** and click **OK**.



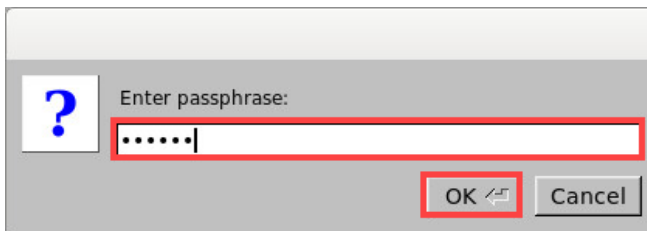
11. When prompted that the *Steghide* message completed successfully, click **OK**.
12. Confirm that the *secret.txt* file is embedded in the *bird.wav* file by clicking on the **Info** menu option and clicking on **Get open file info**.



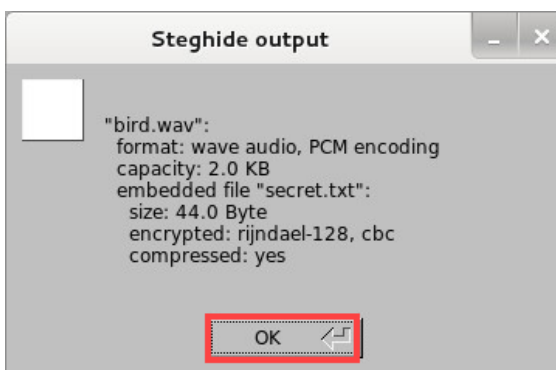
13. When asked to supply a *passphrase*, click **Yes**.



14. Type **secret** and press **OK**.



15. Notice the output given confirming that the "*secret.txt*" file is embedded in the *audio file*. Click **OK**.



16. The lab is now complete; you may end the reservation.