



INTRODUCTION TO CYBERSECURITY

For CHIJ SECONDARY SCHOOL, August 2020



BLOCKCYBER PTE LTD ©2020

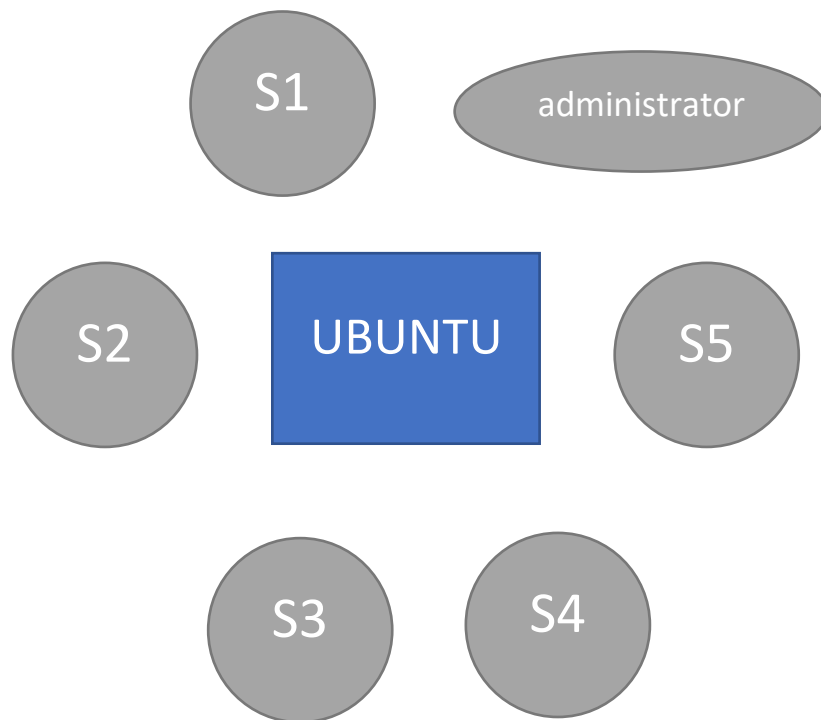
Author: Wong Ding Jie

Date: August 2020

Contents

Lab Design:	2
Module 1: Virtualization	3
Module 2: Linux basic command-line operations.....	4
Terminal:	4
Checking who the current user is – whoami:	5
Checking current directory – pwd:	5
Changing directory – cd:	5
Listing files and folders – ls:	6
Clear the console to get a fresh screen – clear:.....	7
Making a new folder – mkdir:	7
Creating a file with echo and >:	7
Removing a files and folders – rm:	8
Executing scripts and programs - ./:	8
Module 3: Linux Permissions	9
Module 4: Network	10
Checking network interfaces with ip – ip:	10
Module 5: Bash scripting and Cron Scheduling	11
Scenario 1 – erase the contents of a folder:	11
1. Script to clear out the shared_folder:.....	11
2. Schedule the script to run every minute	13
Module 6: Encryption with OpenSSL	15
OpenSSL commands	15
Encrypting data with OpenSSL.....	16
Encrypting images with OpenSSL.....	18
Module 7: Hashing	19
Module 8: Steganography.....	19
Embedding text with Stegosuite.....	19
Extracting text Stegosuite	20
Embedding text file OpenStego	20
Extracting text file OpenStego	21
Embedding and Extracting image file OpenStego	21
Embedding and Extracting sound file OpenStego	21

Lab Design:



Each remote desktop has 5 user accounts and one "administrator" account. All users are SUDO users, meaning they have administration rights.

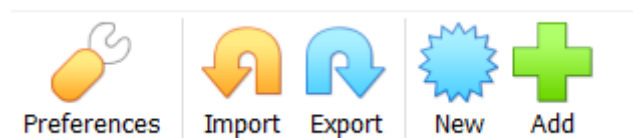
The connection IP address will be shared in class.

Module 1: Virtualization

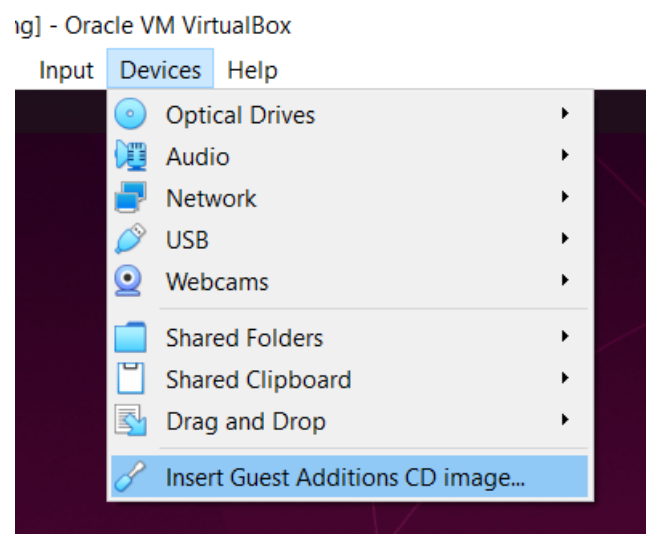
1. Go to <https://www.virtualbox.org/> and download Virtualbox.
2. Go to <https://ubuntu.com/#download> and download the latest Ubuntu ISO.

ISOs are the "disk image" that we can burn into CDs or mount onto optical drives virtually.

3. Install VirtualBox
4. Click on the "new" button to create a new virtual machine.



5. when prompted to choose OS type, choose Ubuntu 64-bit. Usually, our host computers are 64-bit.
6. When prompted on choosing an image to mount, select the Ubuntu image and mount.
7. Install Ubuntu onto the virtual machine and follow the prompts.
8. To enable full screen mode, click on "insert guest additions CD image" under the devices tab. In your Ubuntu, install the VirtualBox guest additions from the CD prompt.

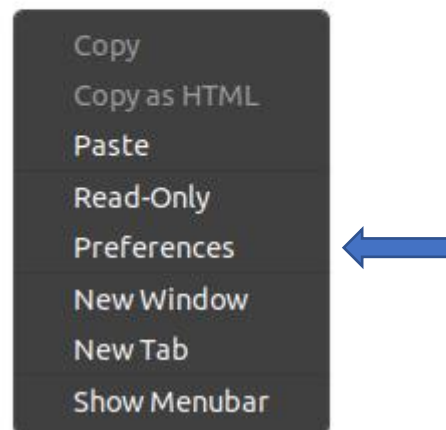


Module 2: Linux basic command-line operations.

Use Remote Desktop Connection, connect using your assigned IP and User account.

Terminal:

From the Ubuntu desktop, use CTRL+ALT+T shortcut to bring up the terminal.



Right click > Preferences.

You can change the color from preferences. E.g. if you want a dark theme, you can change it.

We can change font size as well if we need to see the terminal clearer.

Task:

Adjust the colors, fonts of your terminal to your satisfaction.

Checking who the current user is – whoami:

```
whoami
```

```
s1@ip-172-31-38-241:~/Desktop$ whoami  
s1  
s1@ip-172-31-38-241:~/Desktop$ █
```

Task:

Check who your current user is.

Checking current directory – pwd:

```
pwd
```

```
s1@ip-172-31-38-241:~$ pwd  
/home/s1  
s1@ip-172-31-38-241:~$
```

Task:

Check your current directory with "pwd".

Changing directory – cd:

```
cd Desktop
```

```
/home/s1  
s1@ip-172-31-38-241:~$ cd Desktop/  
s1@ip-172-31-38-241:~/Desktop$ █
```

Changing back to the home directory

```
cd ../
```

```
s1@ip-172-31-38-241:~/Desktop$ cd ../  
s1@ip-172-31-38-241:~$ pwd  
/home/s1
```

Autocomplete:

```
cd De
```

Press Tab key when you type halfway.

This saves us time, and also helps us if we do not know the full name of the folder or file we need.

Tasks:

Change directory to the Desktop.

Change directory back to the user's home directory.

Use the Tab key to autocomplete the file/folder name.

Listing files and folders – ls:

Change directory to the lab folder.

```
cd Desktop/lab
```

Use ls to list all items in the folder.

```
ls
```

Display more details about the files

```
ls -la
```

```
s1@ip-172-31-38-241:~/Desktop/lab$ ls -la  
total 12  
drwxrwxr-x 2 s1 s1 4096 Aug  2 13:11 .  
drwxr-xr-x 3 s1 s1 4096 Aug  2 12:34 ..  
-rw-rw-r-- 1 s1 s1   19 Aug  2 13:11 sample.txt
```

Tasks:

List the items in the lab folder.

Clear the console to get a fresh screen – clear:

```
clear
```

Tasks:

Clear your console.

Making a new folder – mkdir:

Let's make a new folder inside the lab folder.

```
mkdir myfolder
```

Check if the folder is made using ls.

```
s1@ip-172-31-38-241:~/Desktop/lab$ mkdir myfolder
s1@ip-172-31-38-241:~/Desktop/lab$ ls
myfolder  sample.txt
s1@ip-172-31-38-241:~/Desktop/lab$ █
```

Tasks:

Create a folder in the lab folder.

Creating a file with echo and >:

Output something to the console with echo.

```
echo "hey there"
```

```
s1@ip-172-31-38-241:~/Desktop/lab$ echo "hey there"
hey there
_
```


Save the output into a text file

```
echo "hey there" > myfile.txt
```

```
s1@ip-172-31-38-241:~/Desktop/lab/myfolder$ echo "hey there" > myfile.txt
s1@ip-172-31-38-241:~/Desktop/lab/myfolder$ ls
myfile.txt
s1@ip-172-31-38-241:~/Desktop/lab/myfolder$ cat myfile.txt
hey there
s1@ip-172-31-38-241:~/Desktop/lab/myfolder$
```

Tasks:

Create a text file in any folder.

Use "ls" to check if your file is created.

Use "cat" to read the file that you have just created.

Removing a files and folders – rm:

Make sure your console is in the same directory as your file

```
rm myfile.txt
```

To remove a folder, use

```
rm -rf myfolder
```

Tasks:

Delete individual files

Delete myfolder

Executing scripts and programs - ./:

In the lab folder, there is a bash script that will echo to console.

To execute the bash script,

```
./echo_program.sh
```

```
s1@ip-172-31-38-241:~/Desktop/lab$ ./echo_program.sh  
This is a program that echos and output to console :D
```

./ represent the current directory.

We can execute the program with the full path as follows:

```
/home/s1/Desktop/lab/echo_program.sh
```

```
s1@ip-172-31-38-241:~/Desktop/lab$ /home/s1/Desktop/lab/echo_program.sh  
This is a program that echos and output to console :D
```

Tasks:

Execute the script with the console in the same folder.

Execute the script with the full path.

Module 3: Linux Permissions

Tasks:

Change the echo_program.sh to 744.

744 means current user has full read, write and execute permissions. Everyone else can only read.

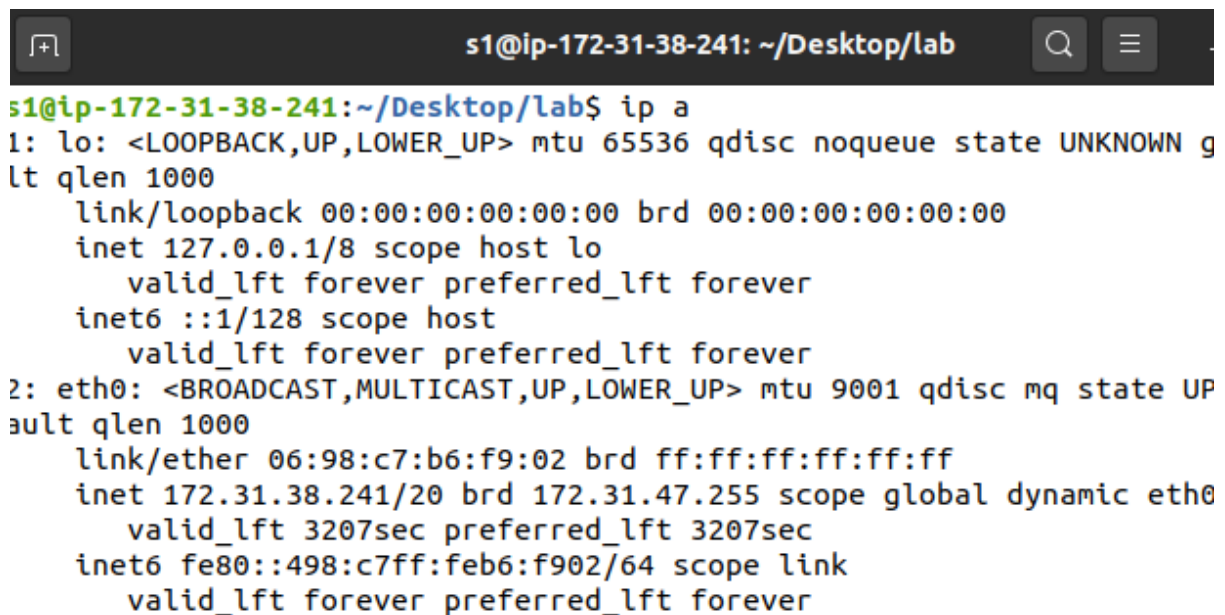
```
chmod 744 echo_program.sh
```

Module 4: Network

Checking network interfaces with ip – ip:

Checking our network

```
ip a
```

A terminal window with a dark background. The title bar shows 's1@ip-172-31-38-241: ~/Desktop/lab'. The prompt is 's1@ip-172-31-38-241:~/Desktop/lab\$'. The command 'ip a' has been entered. The output shows details for the loopback interface 'lo' and the ethernet interface 'eth0'.

```
s1@ip-172-31-38-241:~/Desktop/lab$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
lt qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc mq state UP
ault qlen 1000
    link/ether 06:98:c7:b6:f9:02 brd ff:ff:ff:ff:ff:ff
    inet 172.31.38.241/20 brd 172.31.47.255 scope global dynamic eth0
        valid_lft 3207sec preferred_lft 3207sec
    inet6 fe80::498:c7ff:feb6:f902/64 scope link
        valid_lft forever preferred_lft forever
```

Tasks:

Check your current ip address.

Which network interface is the internet connection on? _____

Use the Desktop interface to check your ip address.

*** Do not turn off the network as all remote desktop connections will be lost, and we will not be able to access the lab.**

Module 5: Bash scripting and Cron Scheduling

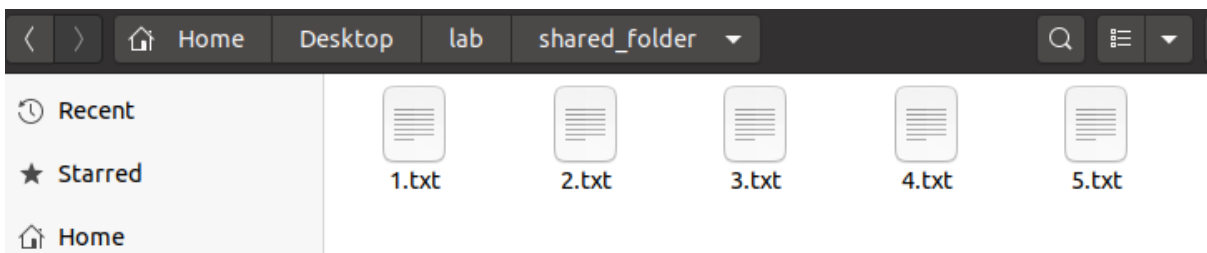
We have seen the individual commands; we can use bash scripting to automate commands.

Scenario 1 – erase the contents of a folder:

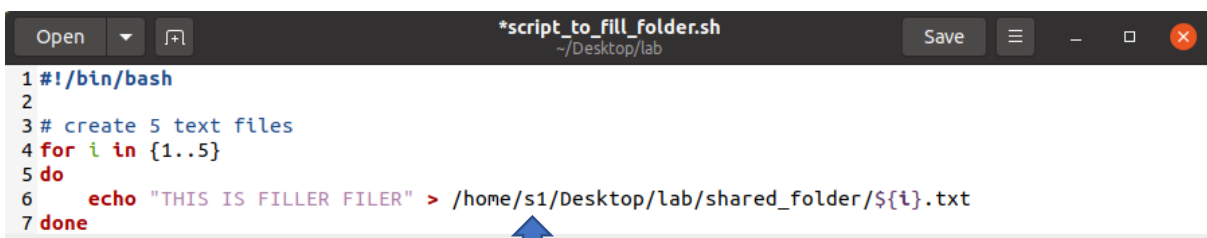
Sarah is the system administrator for Fakeville Secondary School. She oversees and manages the school office systems.

There is a centralized Linux file server that teachers use to share resources, and the shared folder is automatically cleared at midnight every day.

The shared folder has some files inside.



These files are generated by the script_to_fill_folder.sh.



Make sure the path is set to your user folder.

1. Script to clear out the shared_folder:

Tasks:

Let's write a script to remove all files in the shared_folder.

1. Click on the menu button to bring up the ubuntu menu



2. Search for text editor



3. Write the following code into the text editor

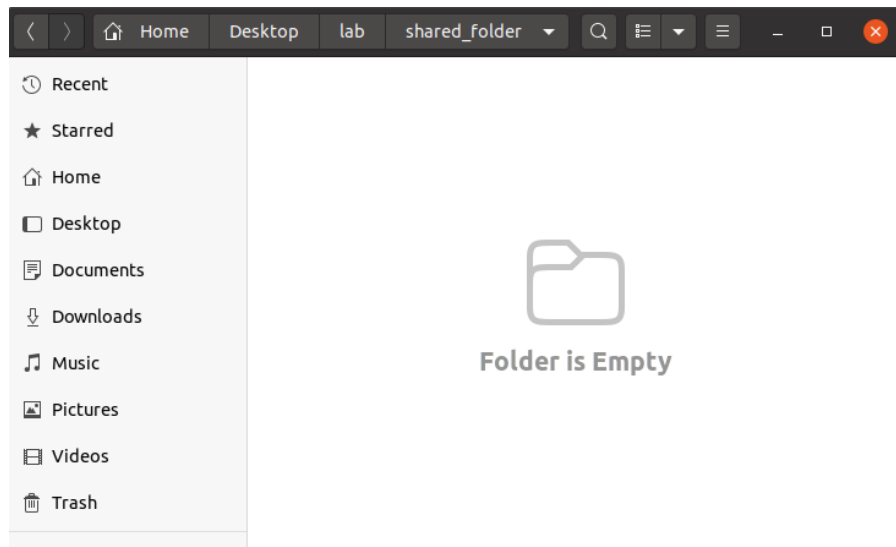


Make sure the path is set to the correct user folder.

Note: "*" the asterisk mark in this case indicates "any". In this case, the logic is: "Remove any files or folder in /home/s1/Desktop/lab/shared_folder"

4. Save the text file into the lab folder and name it **empty_folder.sh**
5. Update the empty_folder.sh permissions to 744
6. Run the script and check if the folder is successfully cleaned.

```
s1@ip-172-31-38-241:~/Desktop/lab$ chmod 744 empty_folder.sh
s1@ip-172-31-38-241:~/Desktop/lab$ ./empty_folder.sh
s1@ip-172-31-38-241:~/Desktop/lab$
```



If there are permission errors, we can run the script with SUDO permissions.

```
s1@ip-172-31-38-241:~/Desktop/lab$ sudo ./empty_folder.sh
s1@ip-172-31-38-241:~/Desktop/lab$
```

2. Schedule the script to run every minute

Tasks:

Let's schedule the script with crontab.

1. run "crontab -e" in the terminal
2. If prompted, choose Nano command line editor

```
s1@ip-172-31-38-241:~/Desktop/lab$ crontab -e
no crontab for s1 - using an empty one
```

Select an editor. To change later, run 'select-editor'.

1. /bin/nano <---- easiest
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny
4. /bin/ed

```
Choose 1-4 [1]: █
```

3. Scroll to the bottom and fill in the cron schedule

```
GNU nano 4.8 /tmp/crontab.FrSiFl/crontab Modif
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
* * * * * /home/s1/Desktop/lab/empty_folder.sh
```



Make sure the path is set to the correct user folder.

4. Press CTRL+O and Enter to save. Press CTRL+X to exit.

The folder should be cleared every minute.

To test it, run the script `to_fill_folder.sh` to fill the folder and see if it deletes it.

If it doesn't work, make sure the crontab is written correctly.

We can also restart the CRON service with:

```
sudo systemctl restart cron
```

Module 6: Encryption with OpenSSL

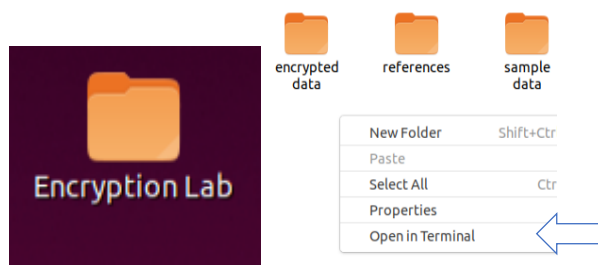
OpenSSL commands

OpenSSL is an encryption tool that is used for HTTPS certificates. We will only look at the basic encryption usage here in this exercise.

Setup:

Go to the “Encryption Lab” folder.

1. Right click and “open terminal here”.



OpenSSL important commands

1. openssl version – to check and confirm version.

Type in the following

```
openssl version
```

```
s1@ip-172-31-38-241:~$ openssl version
OpenSSL 1.1.1f  31 Mar 2020
```

2. Find out which Cipher algorithm to use.

```
openssl help
```

```
s1@ip-172-31-38-241:~$ openssl help
Standard commands
asn1parse          ca                  ciphers             cms
crl                 crl2pkcs7          dgst                 dhparam
dsa                 dsaparam            ec                   ecparam
enc                 engine              errstr               gendsa
genpkey             genrsa              help                 list
```

3. enc

enc is the command we use for **encryption**

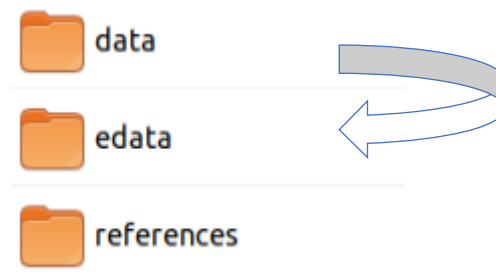
4. enc In / out

In declares the file that we want to encrypt, out is the output filename of the encrypted data.

5. Detailed information of enc options are in the **references folder**

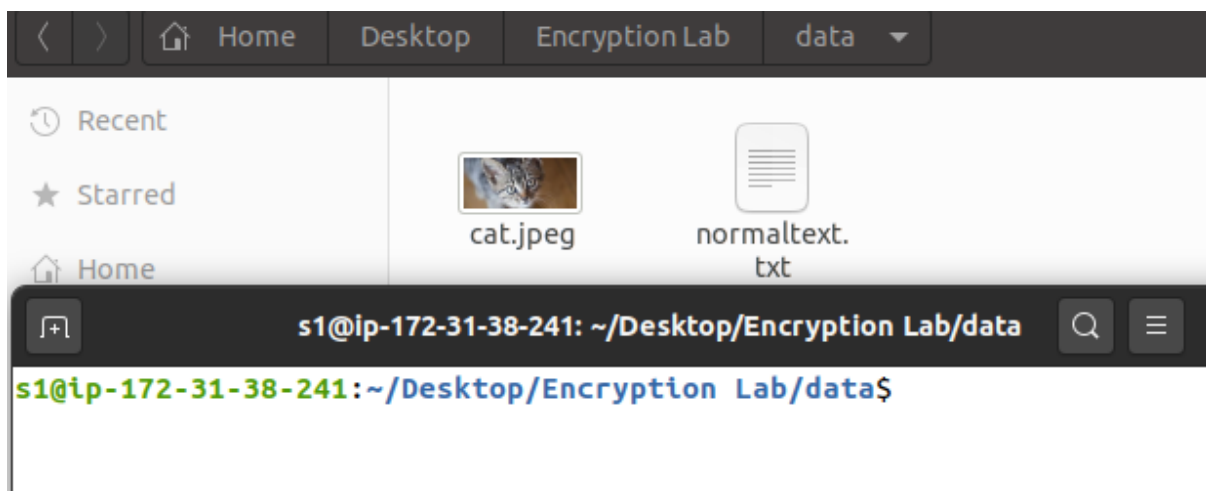
Encrypting data with OpenSSL

We will take the files from **data** folder, encrypt it and output to the **encrypted_data(edata)** folder.



Setup

1. Go into the data folder.
2. Right-Click and select open terminal at **Encryption Lab > data**.



Encrypting a text file

```
openssl enc -aes-128-cbc -in normaltext.txt -out ../edata/output.txt
```

What the command is doing:

“**Openssl encrypt** using **aes-128-cbc** , input file is **normaltext.txt**, save output file called **output.txt** in the folder **edata**.”

```
s1@ip-172-31-38-241: ~/Desktop/Encryption Lab/data
s1@ip-172-31-38-241:~/Desktop/Encryption Lab/data$ openssl enc -aes-128-cbc -in
normaltext.txt -out ../edata/output.txt
enter aes-128-cbc encryption password:
Verifying - enter aes-128-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
s1@ip-172-31-38-241:~/Desktop/Encryption Lab/data$
```

The warning says to use -iter or -pbkdf2, so we can use if we want.

#optional

```
openssl enc -aes-128-cbc -in normaltext.txt -
out ../edata/output.txt -pbkdf2
```

Let's look at the difference in data.

Input file data

Hello. This is a normal text file.
Please use OpenSSL to encrypt me.
Thank you.
Regards,
NormalText.txt

output file data

output.txt
~/Desktop/Encryption Lab/edata
Salted__G= \BF\D9\E9'ho\8A\9E\DE\EB\80&K\CBQG\F3\08
z\8D3\C60\C8\FB\ED\A1I\D9\E9\AA\FC\DB\F7\B38{4 A7\0C%\80\CAH\E6\E6
A8%VL\FEg4k\80\U6I\A6\EA\B2\EB\9E\B8\EC\BFU\B8\BC\FC\CBD\8C\EC\$:
CF2\FEH\BD\B8(\00\EEL\A4t\C4\A6\AB\CS\C6

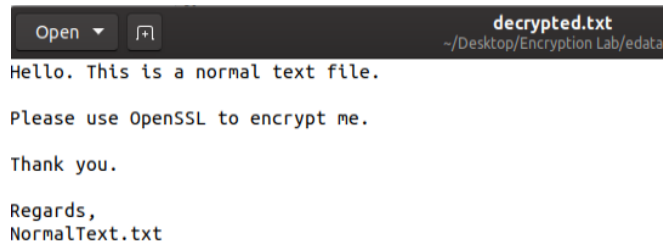
Decrypt our encrypted file.

1. Bring your terminal to the edata folder.
2. Or go to the edata folder, right click and open terminal in edata.

3. Type in the following command.

```
openssl enc -d -aes-128-cbc -in output.txt -out  
decrypted.txt
```

New decrypted file contents.



Summary

To encrypt and decrypt a file, we must know which **cipher** we used, which **password** we set.

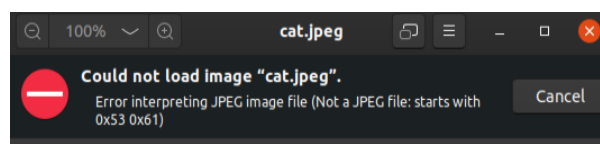
Encrypting images with OpenSSL

1. Go back to the **data** folder.
2. Open terminal in data folder.
3. Type in the following command

```
openssl enc -bf-cbc -in cat.jpeg -out ../edata/cat.jpeg
```

Our input filename and output filename are the same.

Let's look at our encrypted image.



Error! No one can see the image now.

Decrypt the image on your own!

Module 7: Hashing

Tasks:

Hash the text file in the data folder with SHA256.

Change some text in the file and hash it again.

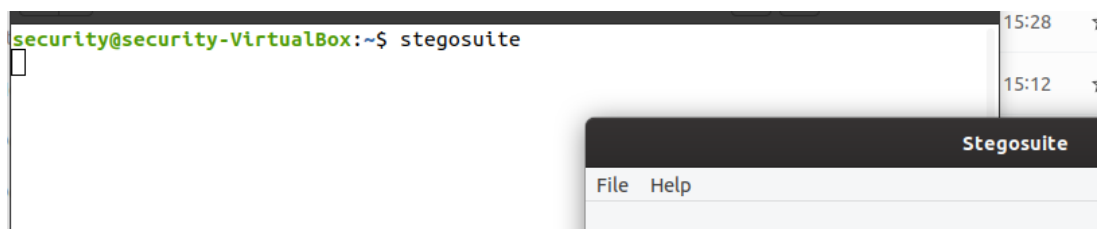
Will the hash be the same? Does the hash look absolutely different?

```
openssl dgst -sha256 normaltext.txt
```

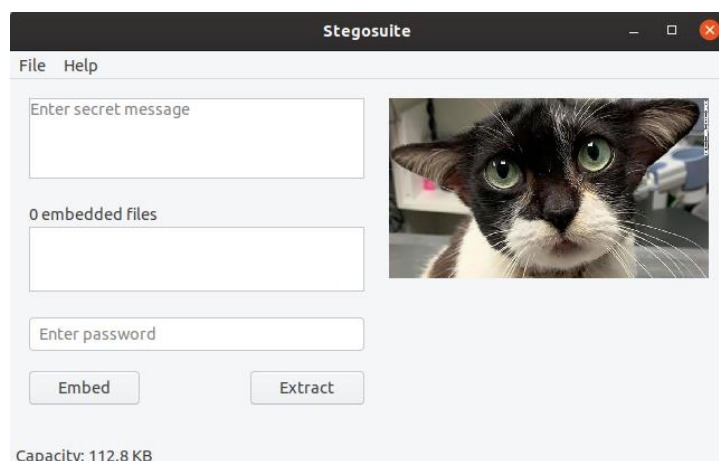
Module 8: Steganography

Embedding text with Stegosuite

1. Open a terminal anywhere.
2. Type in stegosuite and run.



3. Go to **File** > **open** > open the **image1.png** in the **stego_lab** folder on your desktop.



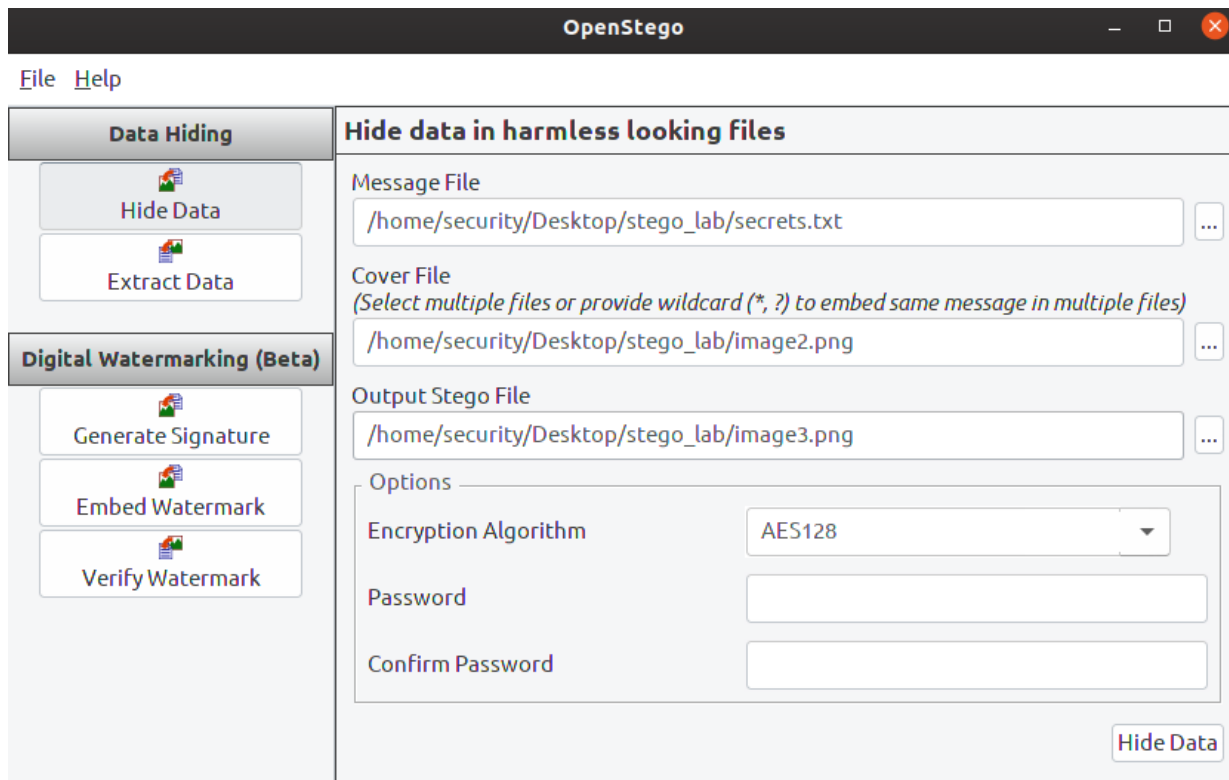
4. Type in any secret message you have.
5. Password is optional, but it can improve the security of the steg file.
6. Click on embed.
7. You should see an image1_embedded.png.

Extracting text Stegosuite

1. **Open > file > open the new image1_embedded.png**
2. Click on **Extract**. 点击 **Extract**.

Embedding text file OpenStego

1. Open OpenStego from the Ubuntu sidebar.
2. In the hide data tab, do the following.



3. The data is now in image3.png.

Extracting text file OpenStego

1. Click on the **Extract Data** tab, select **image3.png** and select the output folder as **Extracted**.

File Help

Data Hiding	Extract hidden data
<div>Hide Data</div> <div>Extract Data</div>	<div>Input Stego File</div> <input type="text" value="/home/security/Desktop/stego_lab/image3.png"/> <div>Output Folder for Message File</div> <input type="text" value="/home/security/Desktop/stego_lab/extracted"/> <div>Password</div> <input type="text"/> <div>Extract Data</div>
Digital Watermarking (Beta)	
<div>Generate Signature</div> <div>Embed Watermark</div>	

2. Click on Extract Data.
3. Go to the **Extracted** folder and open the **secret.txt**.

Embedding and Extracting image file OpenStego

1. Repeat the steps in the previous exercise.
2. Choose image1.png as the message file.

Data Hiding	Hide data in harmless looking files
<div>Hide Data</div> <div>Extract Data</div>	<div>Message File</div> <input type="text" value="/home/security/Desktop/stego_lab/image1.png"/> <div>Cover File</div> <div>(Select multiple files or provide wildcard (*, ?) to embed same message in multiple files)</div> <input type="text" value="/home/security/Desktop/stego_lab/image2.png"/> <div>Output Stego File</div> <input type="text" value="/home/security/Desktop/stego_lab/image3.png"/> <div>Options</div> <div>Encryption Algorithm</div> <div>AES128</div>
Digital Watermarking (Beta)	
<div>Generate Signature</div> <div>Embed Watermark</div>	

3. Extract it to see the difference

Embedding and Extracting sound file OpenStego

1. Repeat the steps with the **doorbell.wav** sound file.
2. Notice we can only put sound file data into image, but can't put image file into the sound file.