

## PRACTICAL: 2

### AIM:

The transmission of information needs to be secure over the communication channel and the data has to be confidential. To do so, steganography is the technique of concealing/hiding a secret file, message, audio, or video in another file format. Study and implement the practical approach for Steganography using the following tools: Steghide, StegoSuite & Xiao Steganography.

### THEORY:

#### What is Steganography?

Steganography is the practice of “hiding in plain sight.” Steganography encodes a secret message within another non-secret object in such a manner as to make the message imperceptible to those who aren’t aware of its presence. Of course, because of this secrecy, steganography generally requires the recipient to be aware that a message is forthcoming.

#### How Steganography works

Steganography works by hiding secret information within a medium, such as an image, audio file, video, or even text, in a way that makes it undetectable to anyone who doesn't know where or how to look. The core idea is to conceal the secret data so that it's not obvious or suspicious to an observer. One common method is called **Least Significant Bit (LSB)** steganography, where secret data is hidden in the least important bits of a file, like an image or audio file.

For example, in an image, each pixel is made up of three-color values: red, green, and blue. Each of these values is stored as a byte (a group of 8 bits). In LSB steganography, the last bit of each byte is changed to hide secret information. Since this change is so small, it doesn't noticeably alter the image, making it look almost the same as the original. So, if you want to hide 1 megabyte of data, you would need an image that is 8 megabytes in size.

The same method can also be used for audio and video files, where small changes are made to the sound or visual elements, making it hard for anyone to notice the hidden data.

#### Types of steganography

From a digital perspective, there are five main types of steganography. These are:

1. Text steganography
2. Image steganography
3. Video steganography
4. Audio steganography
5. Network steganography

### Text steganography

Text steganography involves hiding information inside text files. This includes changing the format of existing text, changing words within a text, using context-free grammars to generate readable texts, or generating random character sequences.

### Image steganography

This involves hiding information within image files. In digital steganography, images are often used to conceal information because there are a large number of elements within the digital representation of an image, and there are various ways to hide information inside an image.

### Audio steganography

Audio steganography involves secret messages being embedded into an audio signal which alters the binary sequence of the corresponding audio file. Hiding secret messages in digital sound is a more difficult process compared to others.

### Video steganography

This is where data is concealed within digital video formats. Video steganography allows large amounts of data to be hidden within a moving stream of images and sounds. Two types of video steganography are:

- Embedding data in uncompressed raw video and then compressing it later
- Embedding data directly into the compressed data stream

### Network steganography

Network steganography, sometimes known as protocol steganography, is the technique of embedding information within network control protocols used in data transmission such TCP, UDP, ICMP, etc.

### Uses of steganography

In recent times, steganography has been mainly used on computers with digital data being the carriers and networks being the high-speed delivery channels. Steganography uses include:

- **Avoiding censorship:** Using it to send news information without it being censored and without fear of the messages being traced back to their sender.
- **Digital watermarking:** Using it to create invisible watermarks that do not distort the image, while being able to track if it has been used without authorization.
- **Securing information:** Used by law enforcement and government agencies to send highly sensitive information to other parties without attracting suspicion.

## How to detect steganography

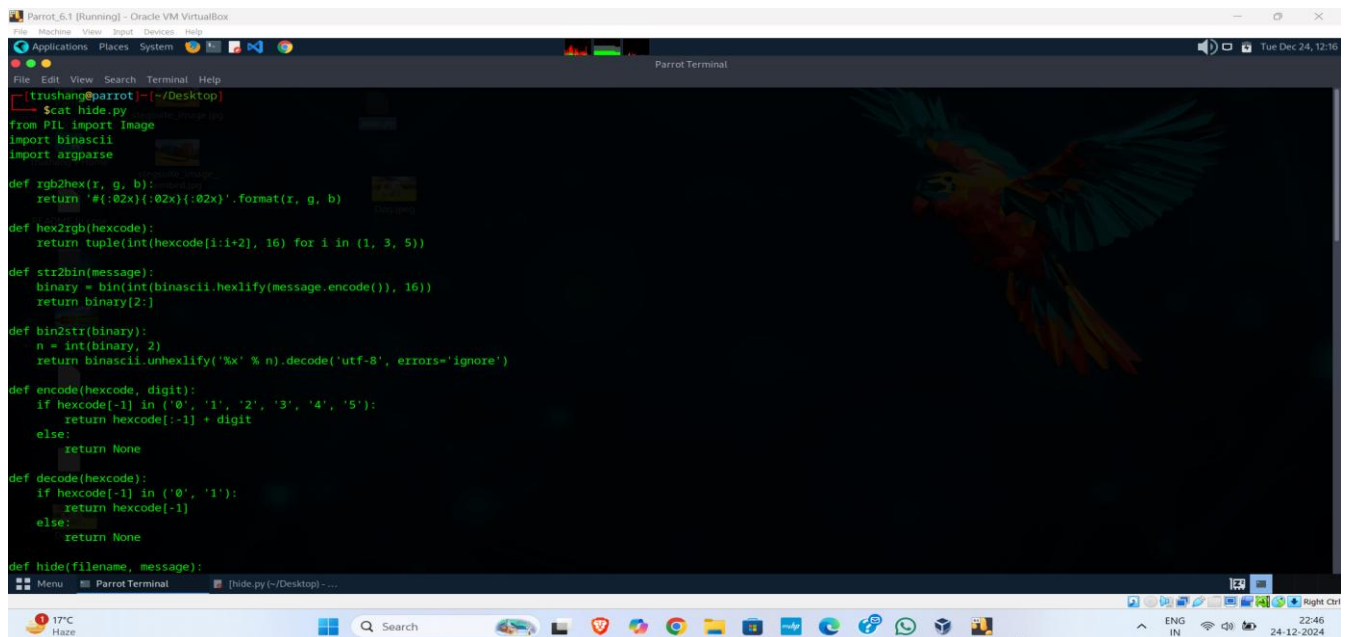
The practice of detecting steganography is called ‘steganalysis’. There are various tools that can detect the presence of hidden data, including StegExpose and StegAlyze. Analysts may use other general analysis tools such as hex viewers to detect anomalies in files.

However, finding files that have been modified through steganography is a challenge – not least because knowing where to start looking for hidden data in the millions of images being uploaded on social media every day is virtually impossible.

## CODE:

- nano hide.py
- python hide.py -e '/home/trushang/Desktop/Dog. bmp'
- python hide.py -d '/home/trushang/Desktop/Dog. bmp'
- sudo apt-get install steghide
- steghide --version
- nano secret.txt
- cat secret.txt
- steghide --embed -ef '/home/trushang/Desktop/secret .txt' -cf '/home/trushang/Desktop/steghide\_ image.jpeg' -p 22it116
- steghide --extract -sf '/home/trushang/Desktop/steghide\_image . jpeg' -p 22it116 -xf '/home/trushang/Desktop/secrets. txt '
- sudo apt-get install stegosuite
- stegosuite gui
- copy /b download .jpg + file. zip image .jpg
- ren image . jpg image. zip

## OUTPUT:



```

Parrot 6.1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
(trushang@parrot) ~/Desktop
$ cat hide.py
from PIL import Image
import binascii
import argparse

def rgb2hex(r, g, b):
    return '#{0:02x}{0:02x}{0:02x}'.format(r, g, b)

def hex2rgb(hexcode):
    return tuple(int(hexcode[i:i+2], 16) for i in (1, 3, 5))

def str2bin(message):
    binary = bin(int(binascii.hexlify(message.encode()), 16))
    return binary[2:]

def bin2str(binary):
    n = int(binary, 2)
    return binascii.unhexlify('%x' % n).decode('utf-8', errors='ignore')

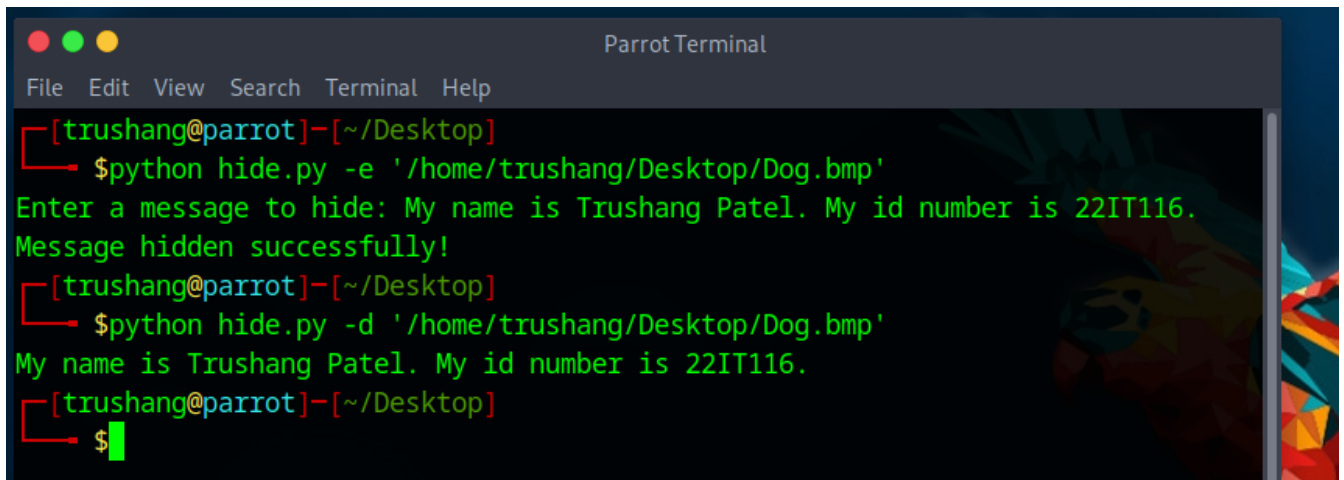
def encode(hexcode, digit):
    if hexcode[-1] in ('0', '1', '2', '3', '4', '5'):
        return hexcode[-1] + digit
    else:
        return None

def decode(hexcode):
    if hexcode[-1] in ('0', '1'):
        return hexcode[-1]
    else:
        return None

def hide(filename, message):

```

Figure 1:hide.py where we write a logic for hiding data in image

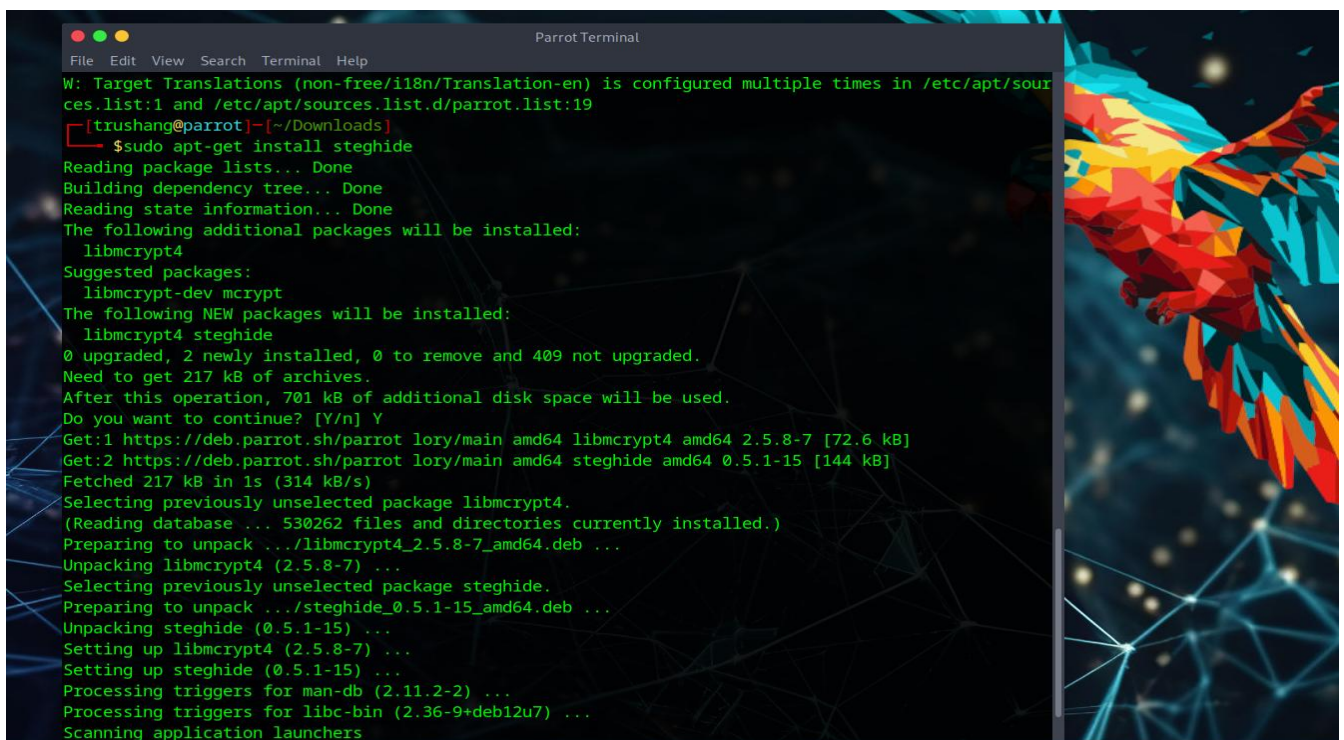


```

Parrot Terminal
File Edit View Search Terminal Help
[trushang@parrot]-[~/Desktop]
$python hide.py -e '/home/trushang/Desktop/Dog.bmp'
Enter a message to hide: My name is Trushang Patel. My id number is 22IT116.
Message hidden successfully!
[trushang@parrot]-[~/Desktop]
$python hide.py -d '/home/trushang/Desktop/Dog.bmp'
My name is Trushang Patel. My id number is 22IT116.
[trushang@parrot]-[~/Desktop]
$

```

Figure 2: Hide message in image using python code

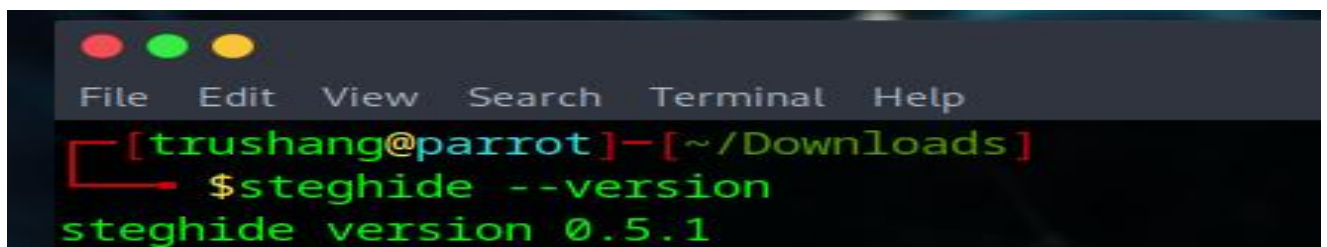


```

Parrot Terminal
File Edit View Search Terminal Help
W: Target Translations (non-free/i18n/Translation-en) is configured multiple times in /etc/apt/sources.list:1 and /etc/apt/sources.list.d/parrot.list:19
[trushang@parrot]-[~/Downloads]
$sudo apt-get install steghide
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libmccrypt4
Suggested packages:
  libmccrypt-dev mccrypt
The following NEW packages will be installed:
  libmccrypt4 steghide
0 upgraded, 2 newly installed, 0 to remove and 409 not upgraded.
Need to get 217 kB of archives.
After this operation, 701 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://deb.parrot.sh/parrot lory/main amd64 libmccrypt4 amd64 2.5.8-7 [72.6 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 steghide amd64 0.5.1-15 [144 kB]
Fetched 217 kB in 1s (314 kB/s)
Selecting previously unselected package libmccrypt4.
(Reading database ... 530262 files and directories currently installed.)
Preparing to unpack .../libmccrypt4_2.5.8-7_amd64.deb ...
Unpacking libmccrypt4 (2.5.8-7) ...
Selecting previously unselected package steghide.
Preparing to unpack .../steghide_0.5.1-15_amd64.deb ...
Unpacking steghide (0.5.1-15) ...
Setting up libmccrypt4 (2.5.8-7) ...
Setting up steghide (0.5.1-15) ...
Processing triggers for man-db (2.11.2-2) ...
Processing triggers for libc-bin (2.36-9+deb12u7) ...
Scanning application launchers

```

Figure 3: First install steghide in our parrot security os



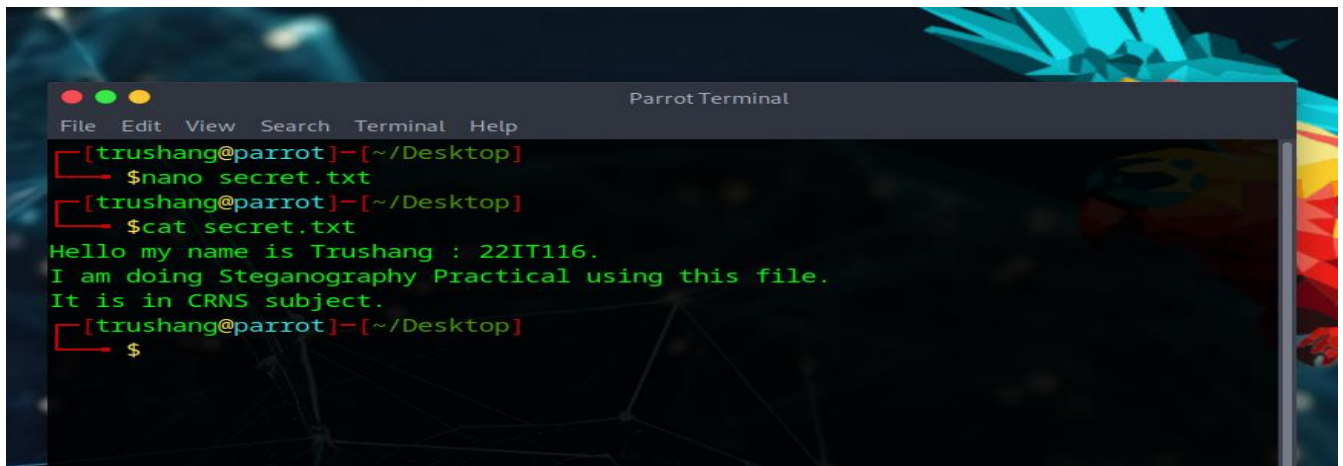
```

Parrot Terminal
File Edit View Search Terminal Help
[trushang@parrot]-[~/Downloads]
$steghide --version
steghide version 0.5.1

```

Figure 4: Check the version of steghide



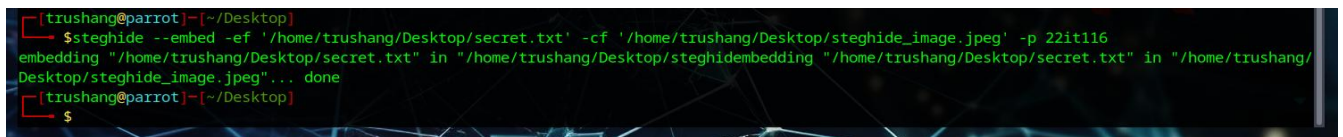


```

[trushang@parrot]~[~/Desktop]
$ nano secret.txt
[trushang@parrot]~[~/Desktop]
$ cat secret.txt
Hello my name is Trushang : 22IT116.
I am doing Steganography Practical using this file.
It is in CRNS subject.
[trushang@parrot]~[~/Desktop]
$

```

Figure 5: Simple create a secret text file which contains your secret data



```

[trushang@parrot]~[~/Desktop]
$ steghide --embed -ef '/home/trushang/Desktop/secret.txt' -cf '/home/trushang/Desktop/steghide_image.jpeg' -p 22it116
embedding "/home/trushang/Desktop/secret.txt" in "/home/trushang/Desktop/steghide_image.jpeg"... done
[trushang@parrot]~[~/Desktop]
$

```

Figure 6: This command hides your secret file into steghide\_image file



```

[trushang@parrot]~[~/Desktop]
$ steghide --extract -sf '/home/trushang/Desktop/steghide_image.jpeg' -p 22it116 -xf '/home/trushang/Desktop/secrets.txt'
wrote extracted data to "/home/trushang/Desktop/secrets.txt".
[trushang@parrot]~[~/Desktop]
$

```

Figure 7: This command extracts your secret message and store it into secrets.txt

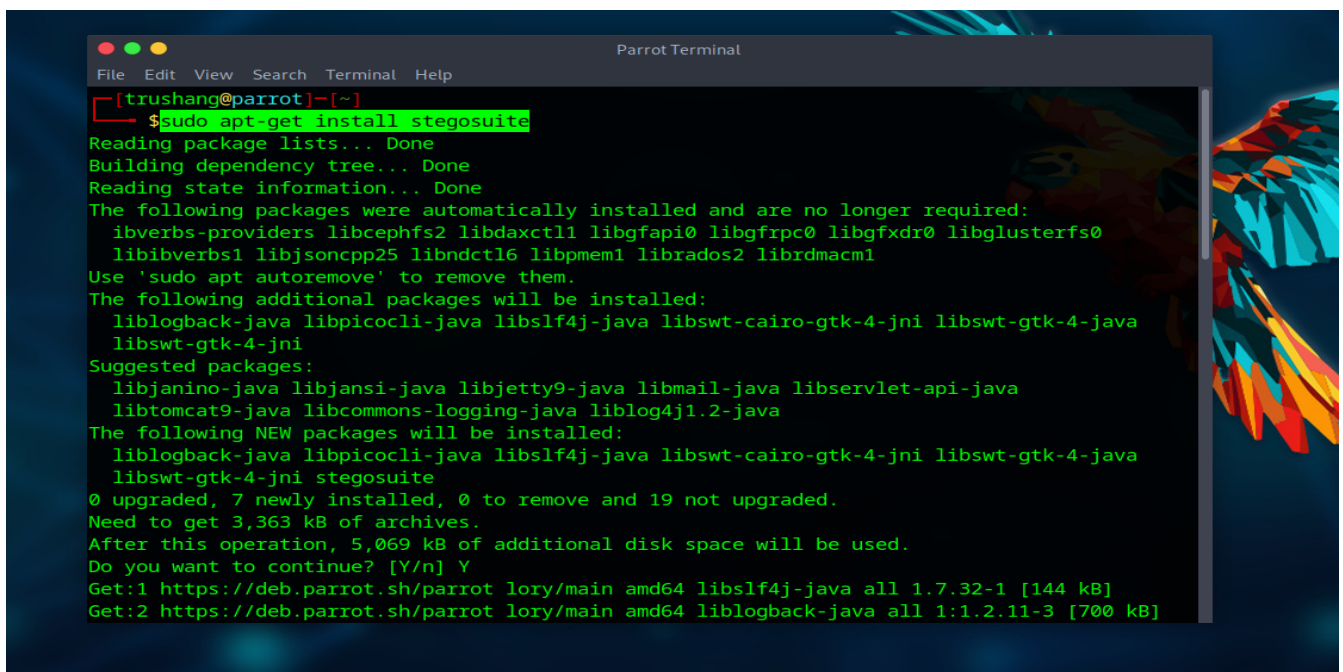


```

$ cat secrets.txt
Hello my name is Trushang : 22IT116.
I am doing Steganography Practical using this file.
It is in CRNS subject.
[trushang@parrot]~[~/Desktop]
$

```

Figure 8: Output of secrets.txt which is hidden by sender



```

[trushang@parrot]~[~]
$ sudo apt-get install stegosuite
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libverbs-providers libcephfs2 libdaxctl1 libgfs2 libgfrpc0 libgfsxdr0 libglusterfs0
  libibverbs1 libjsoncpp25 libndctl6 libpmem1 librados2 librdmacm1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  liblogback-java libpicocli-java libslf4j-java libswt-cairo-gtk-4-jni libswt-gtk-4-java
  libswt-gtk-4-jni
Suggested packages:
  libjanino-java libjetty9-java libmail-java libservlet-api-java
  libtomcat9-java libcommons-logging-java liblog4j1.2-java
The following NEW packages will be installed:
  liblogback-java libpicocli-java libslf4j-java libswt-cairo-gtk-4-jni libswt-gtk-4-java
  libswt-gtk-4-jni stegosuite
0 upgraded, 7 newly installed, 0 to remove and 19 not upgraded.
Need to get 3,363 kB of archives.
After this operation, 5,069 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 https://deb.parrot.sh/parrot lory/main amd64 libslf4j-java all 1.7.32-1 [144 kB]
Get:2 https://deb.parrot.sh/parrot lory/main amd64 liblogback-java all 1:1.2.11-3 [700 kB]

```

Figure 9: Install Steg suite in your system

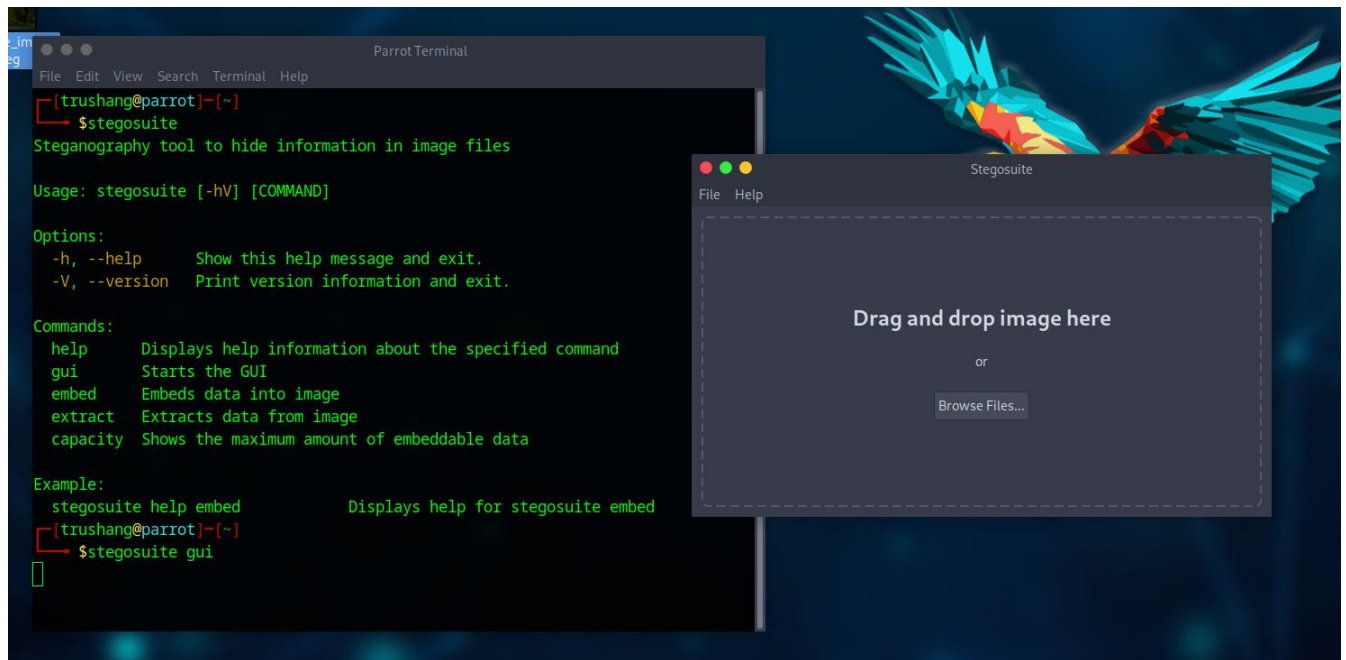


Figure 10: Open stego suite GUI

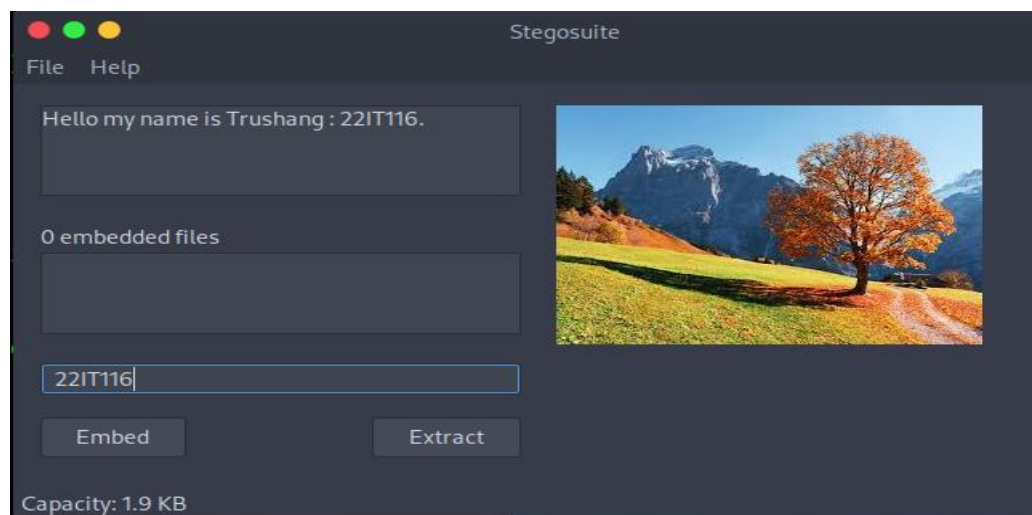


Figure 11: Drag file and set secret message and password

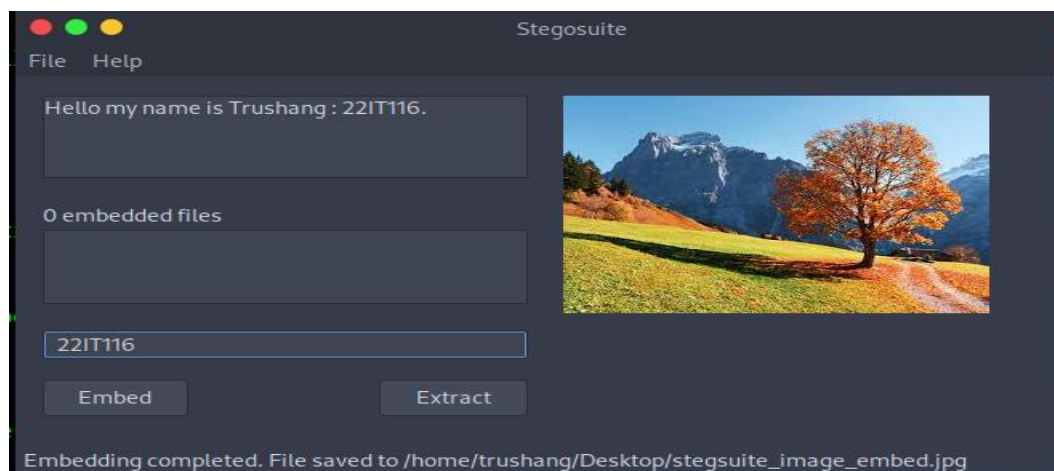


Figure 12: After clicking Embed we create new file with hiding message

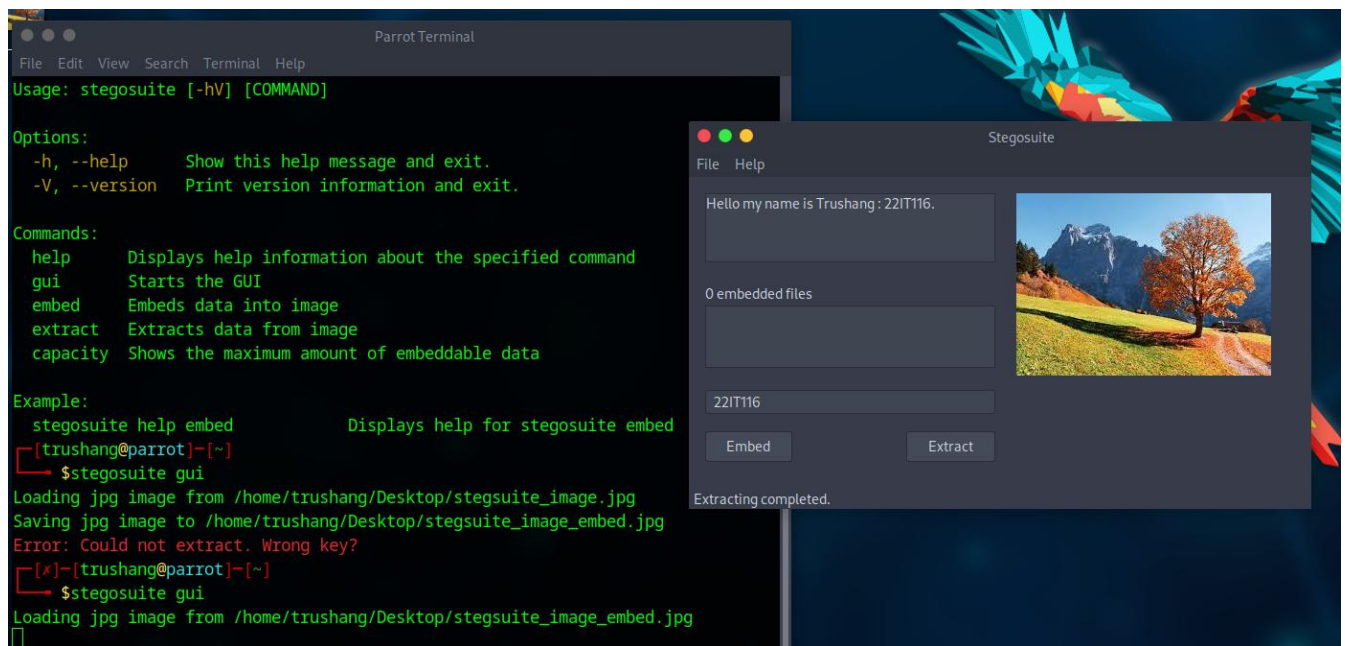


Figure 13: Now Extract the hide data from image

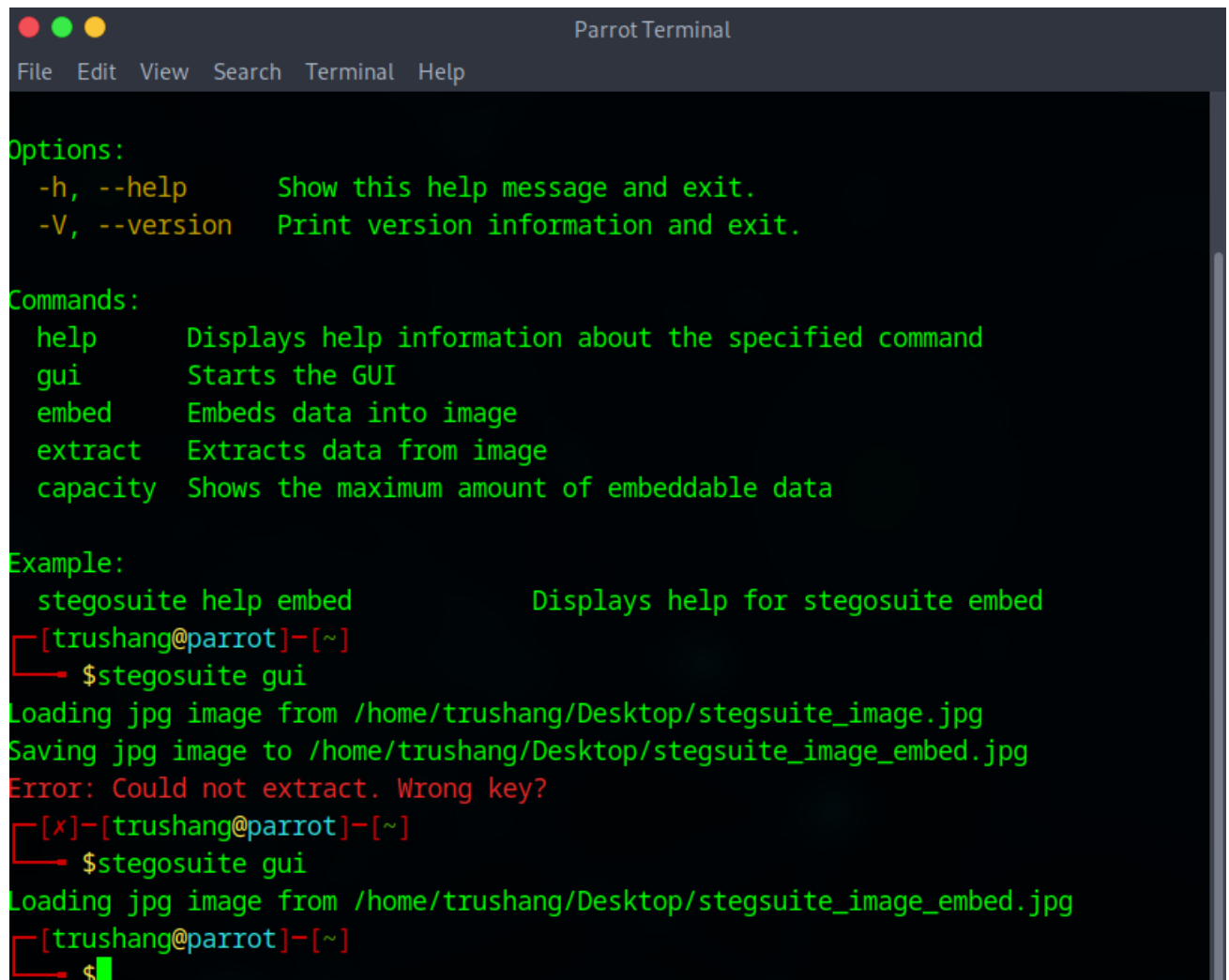


Figure 14: CLI for Steg suite



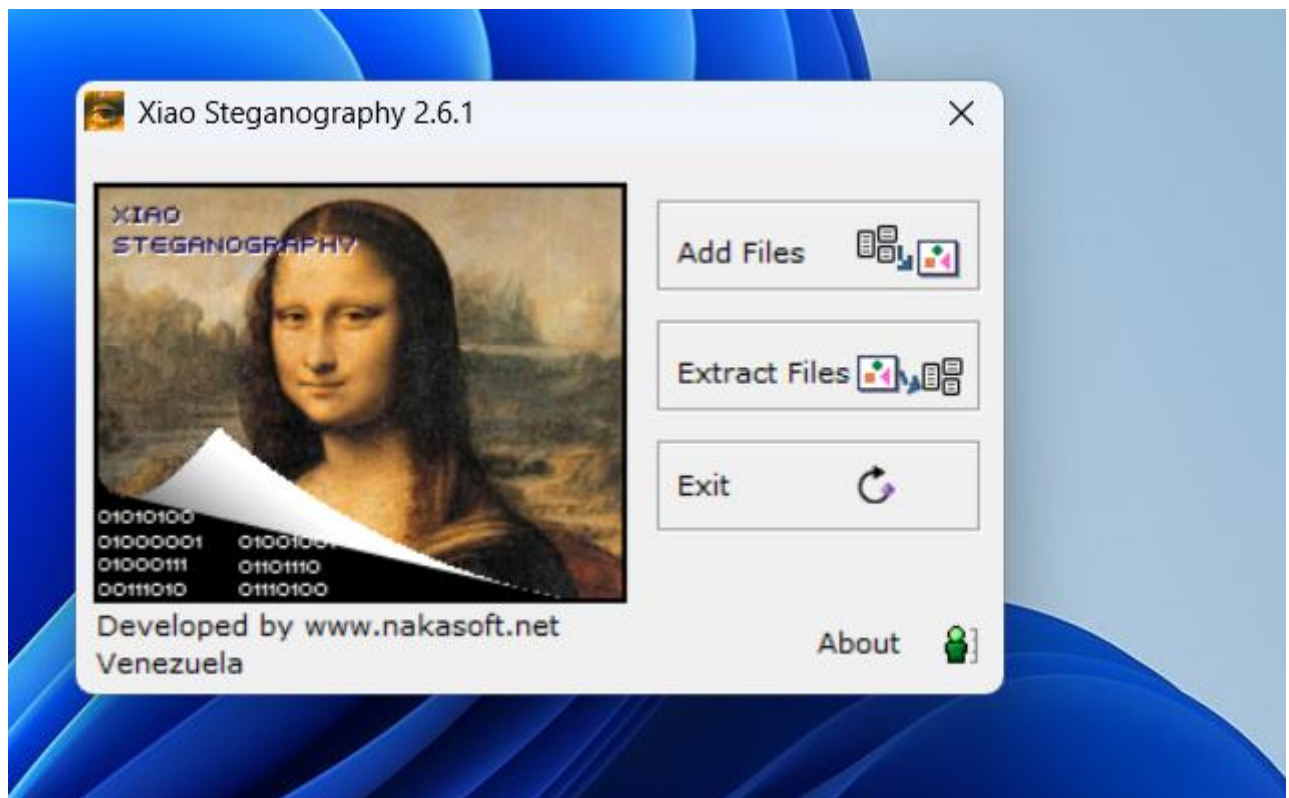


Figure 15:After installing the xiao Steganography

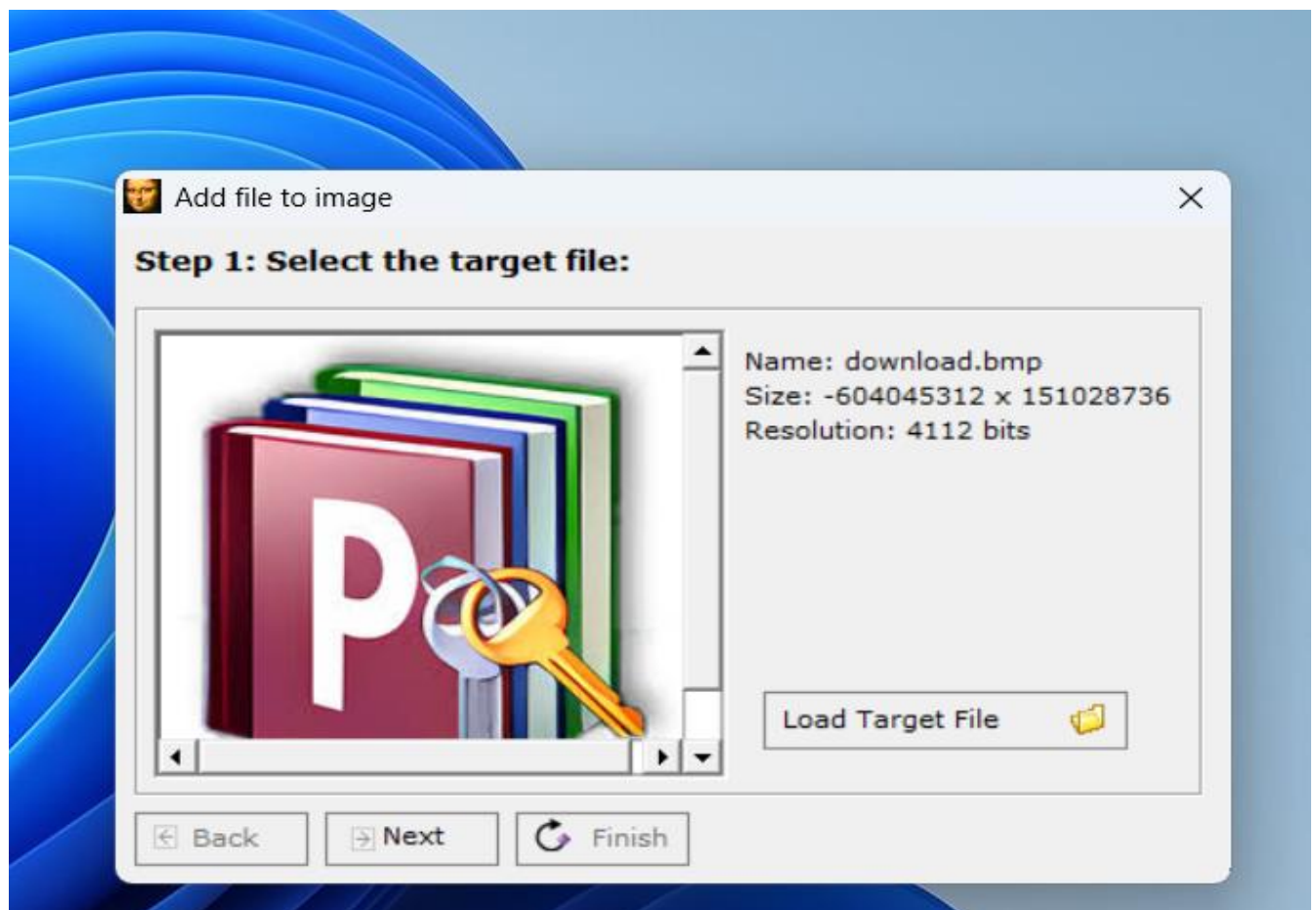


Figure 16:Add image into xios Steganography



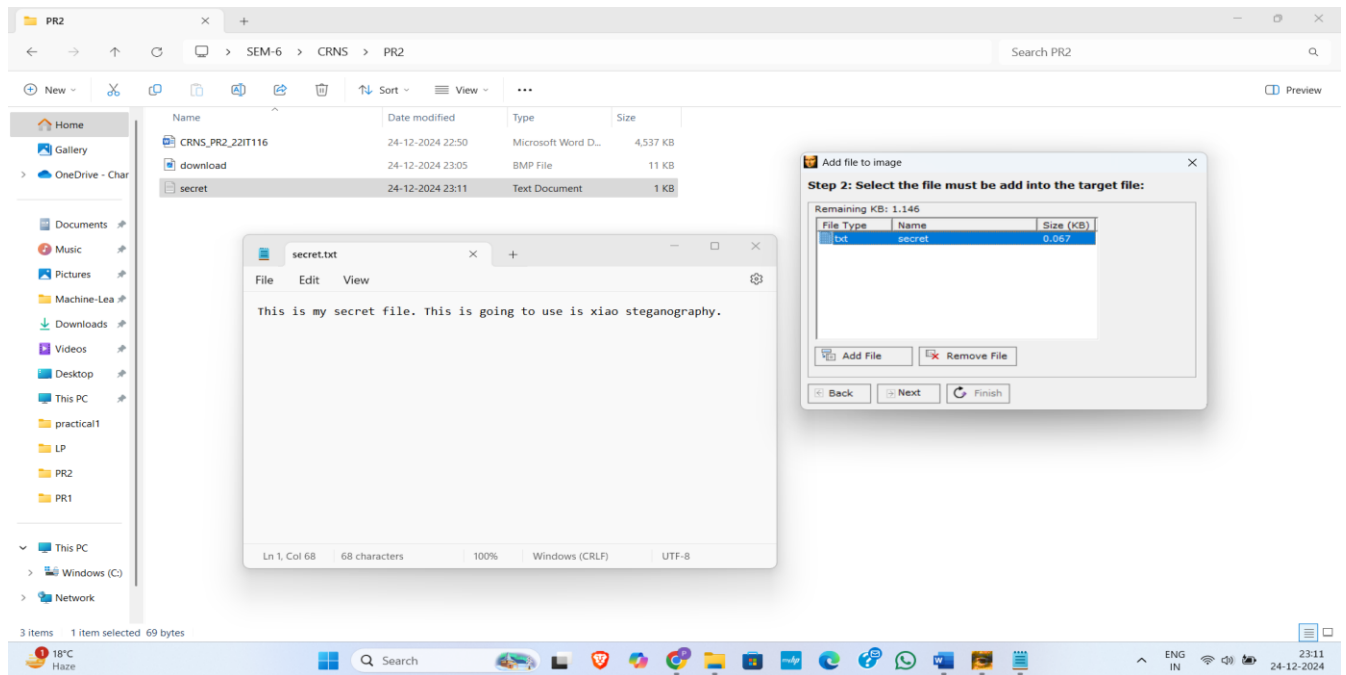


Figure 17: Add secret file to the xiao steganography

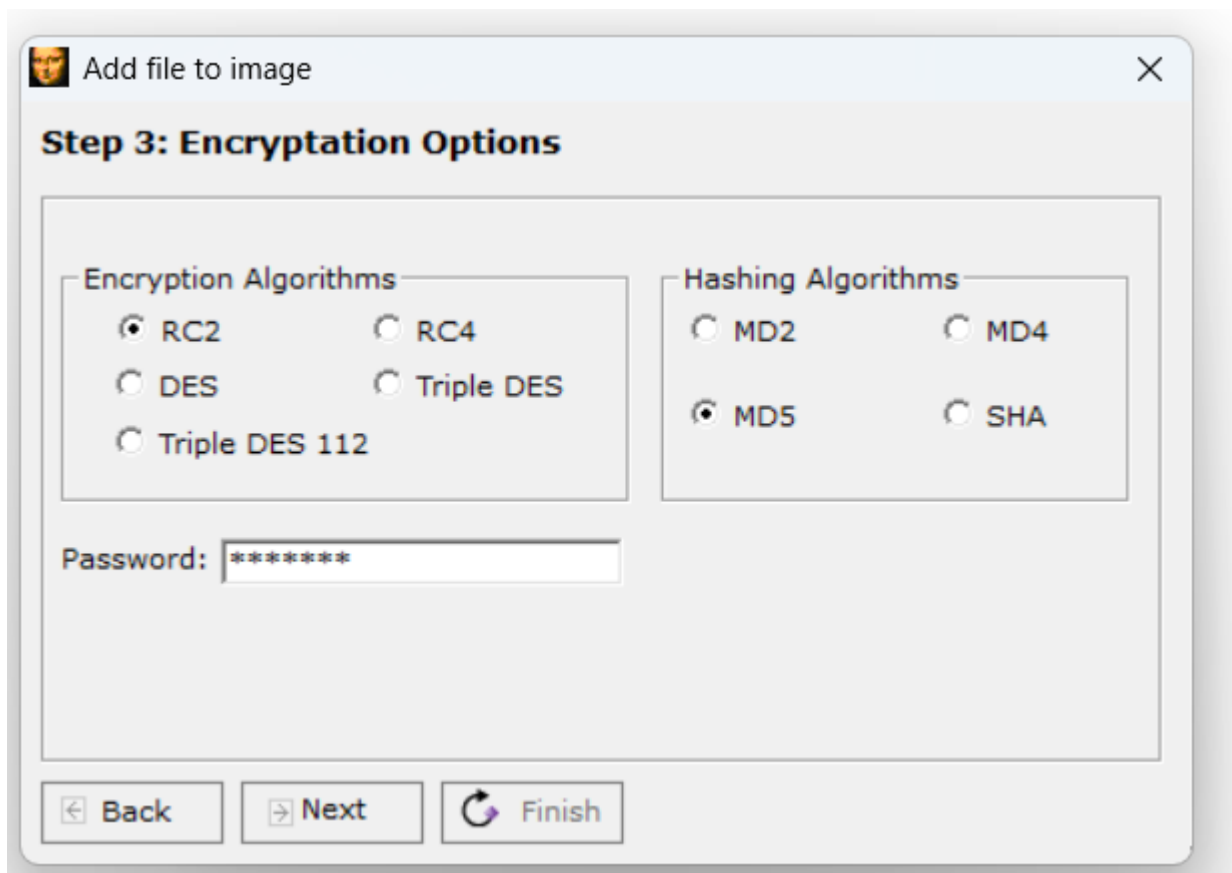


Figure 18: Choose encryption options and set password

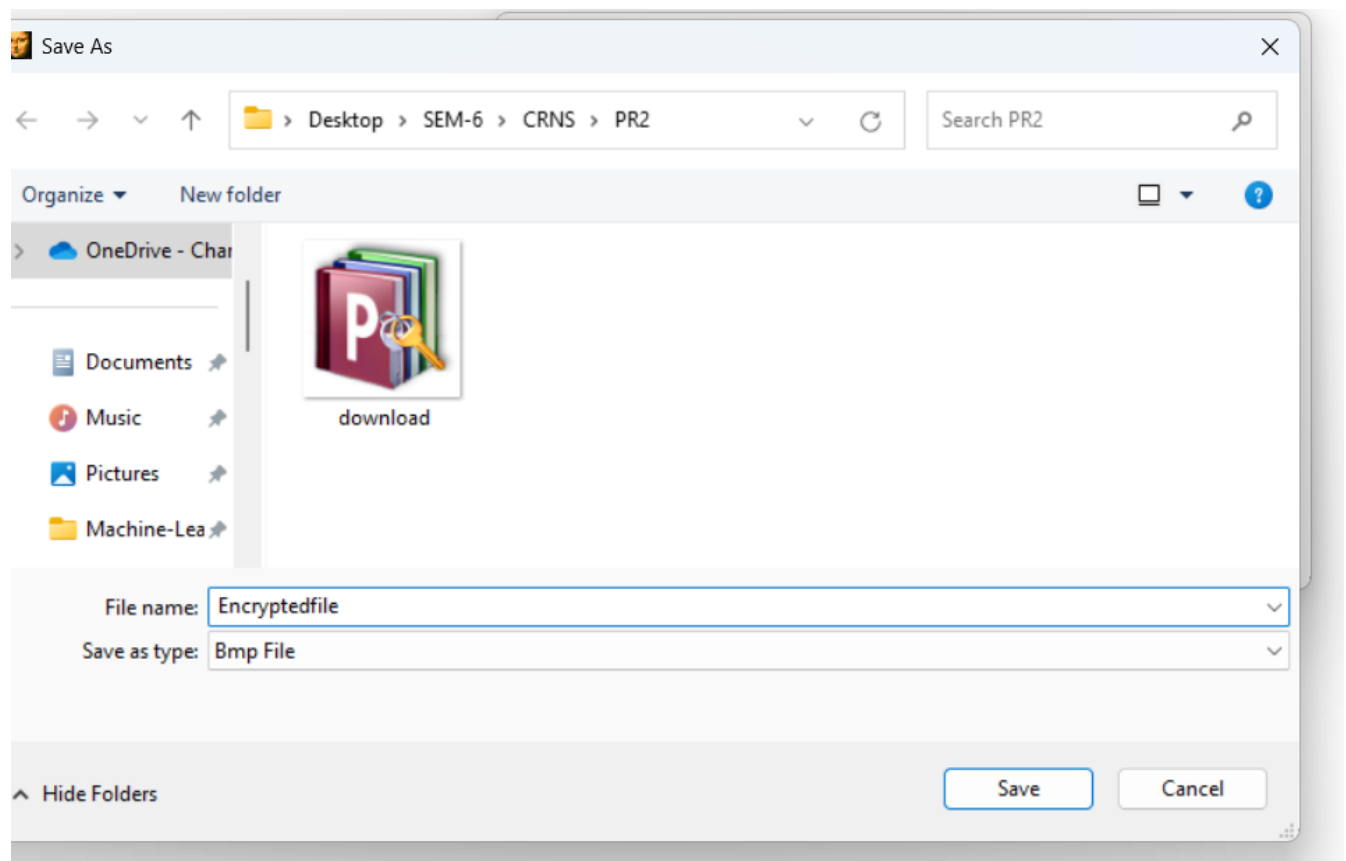


Figure 19: Save file in our local system

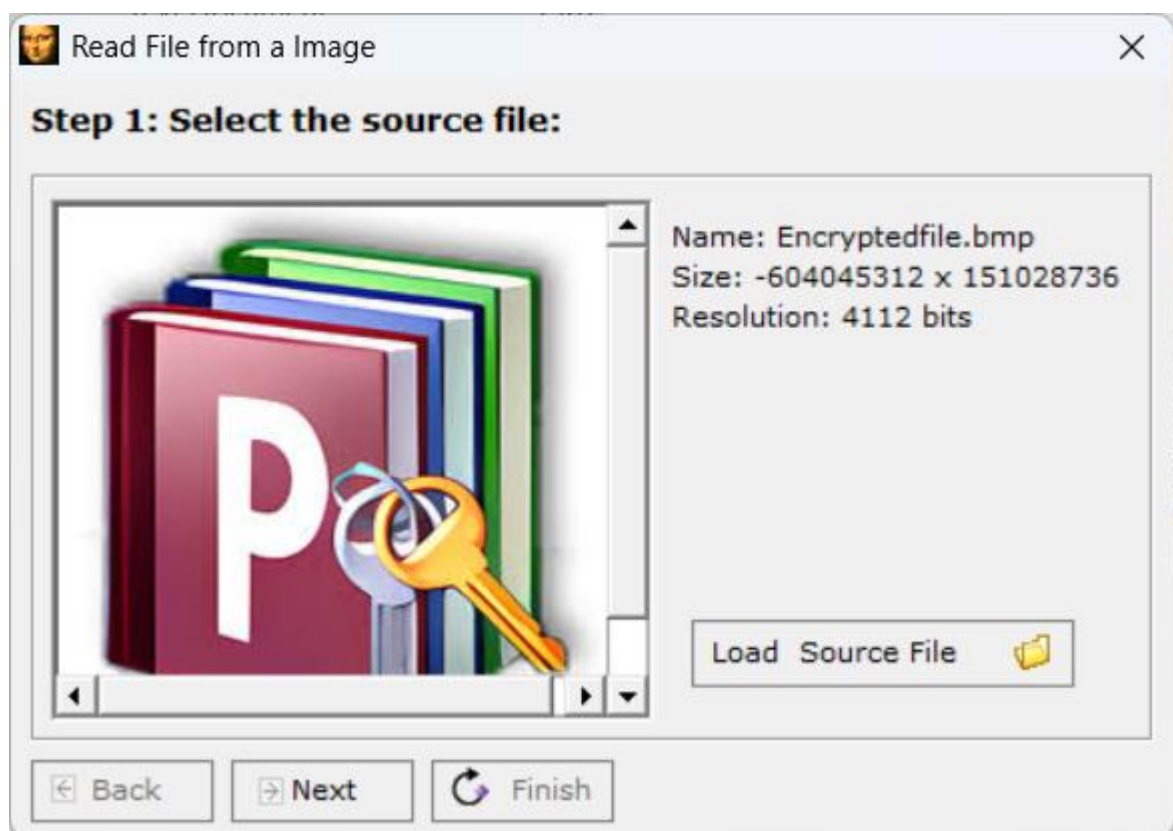


Figure 20: Select stegno file for decryptions

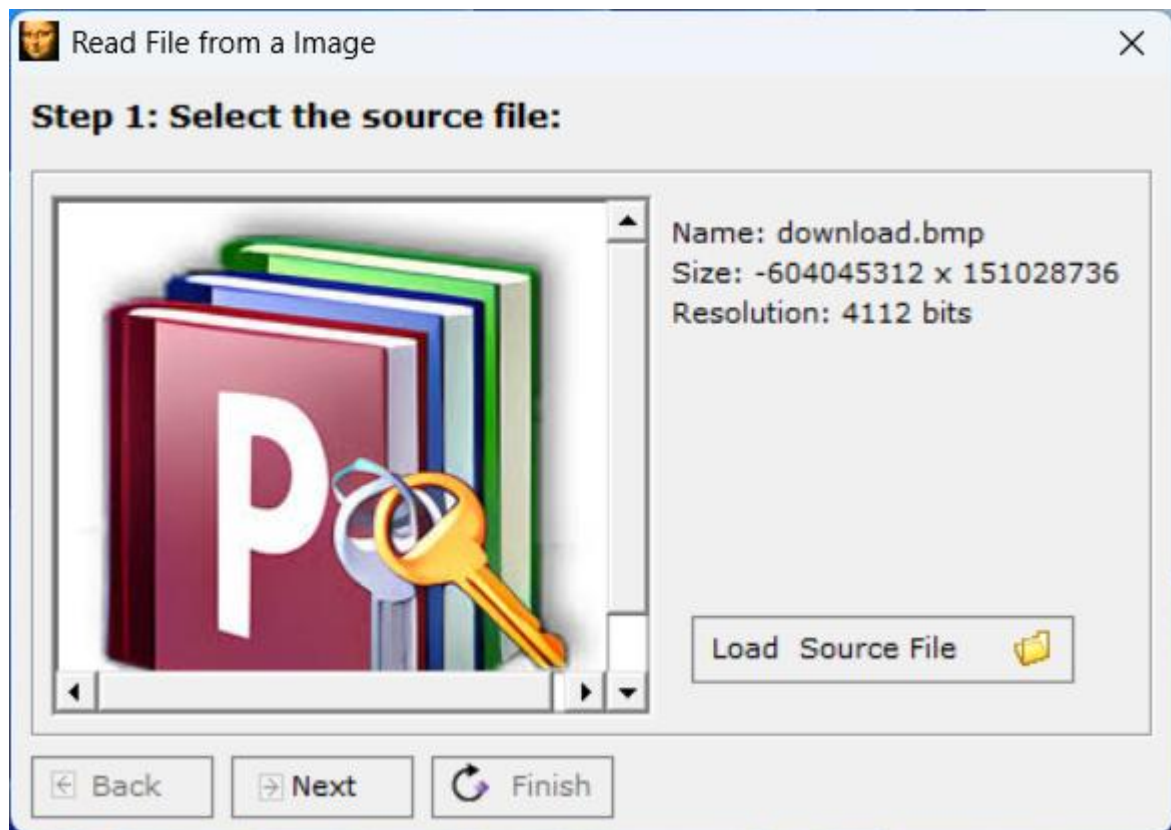


Figure 21: To extraction we use same file as we use for original image

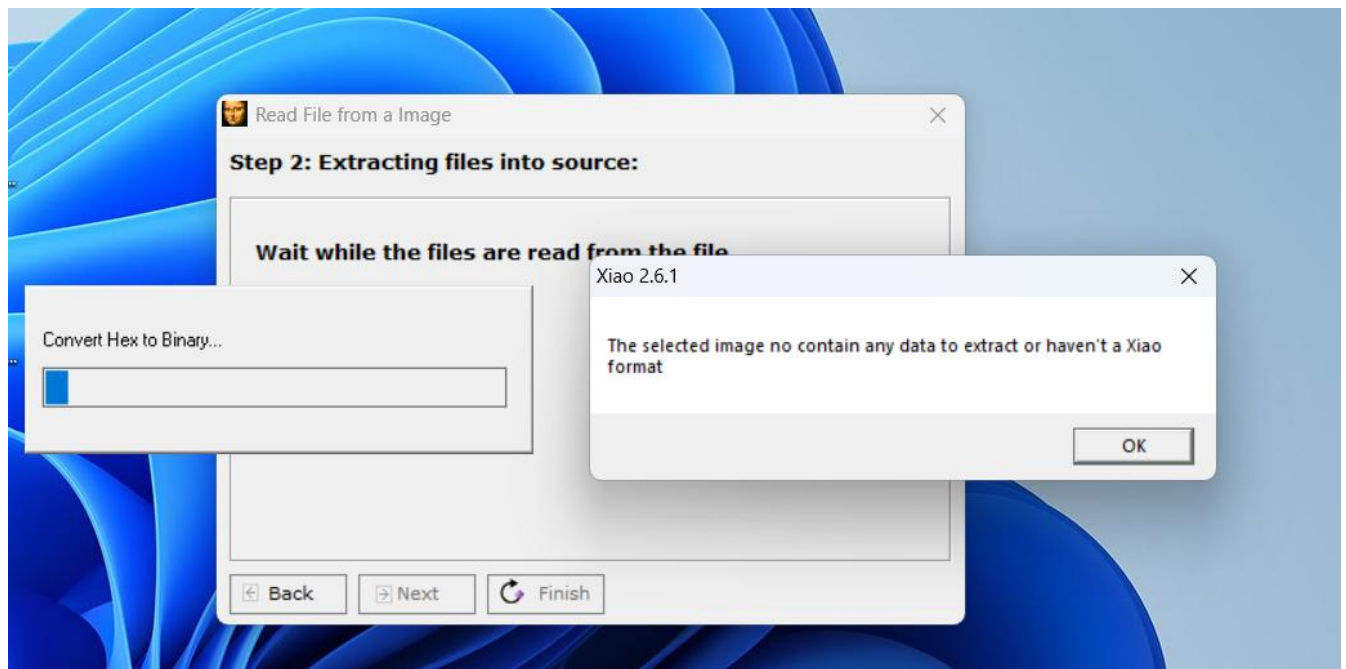


Figure 22: We uploaded original file without steganography so we get output like the selected image no contain any data to extract or haven't a Xiao

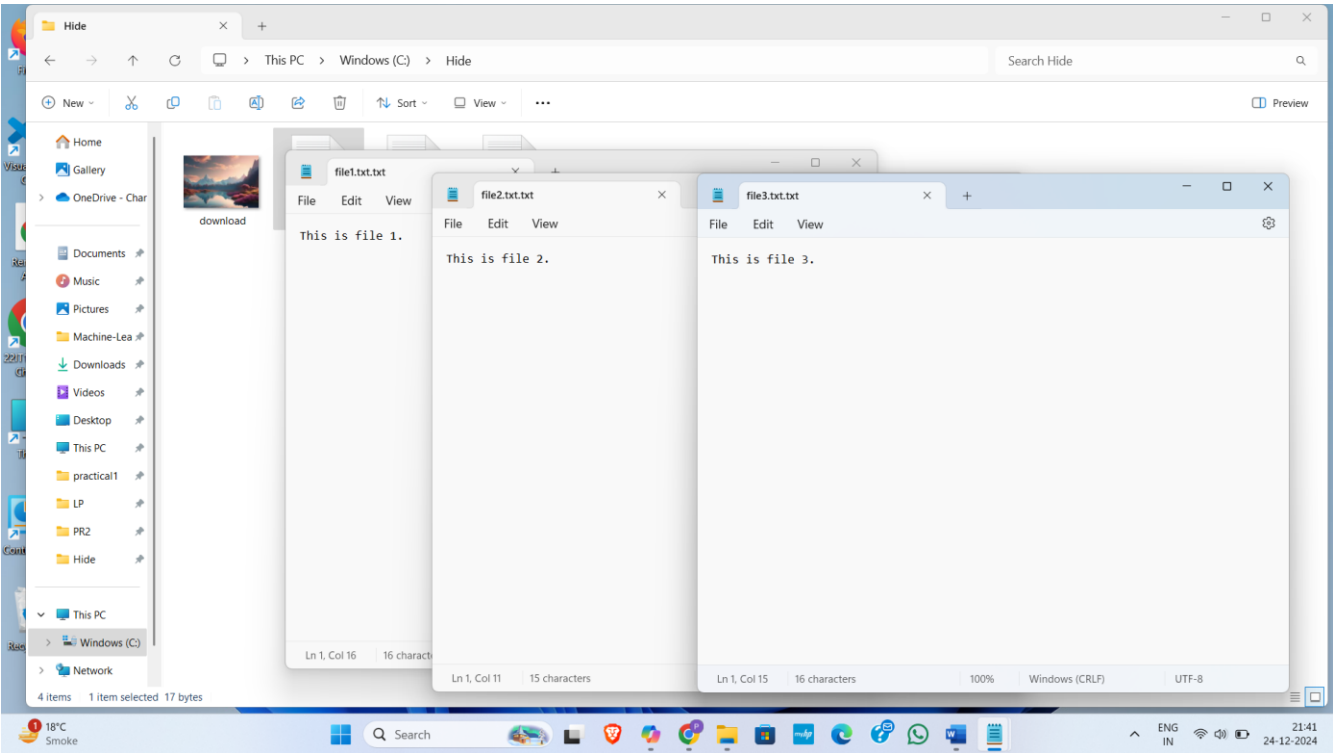


Figure 23: Created a folder in C drive name hide and we have an image name download and three file which we have to hide under image

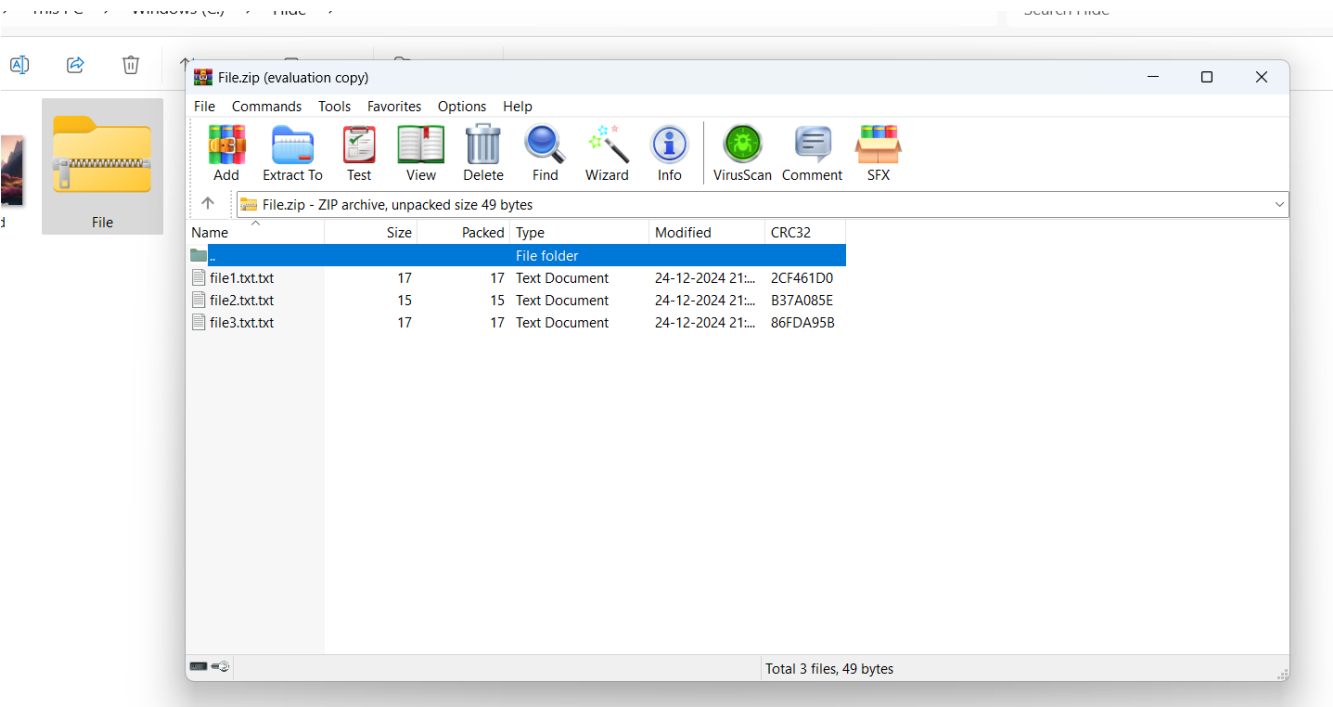


Figure 24: Create ZIP file which you want to hide in image



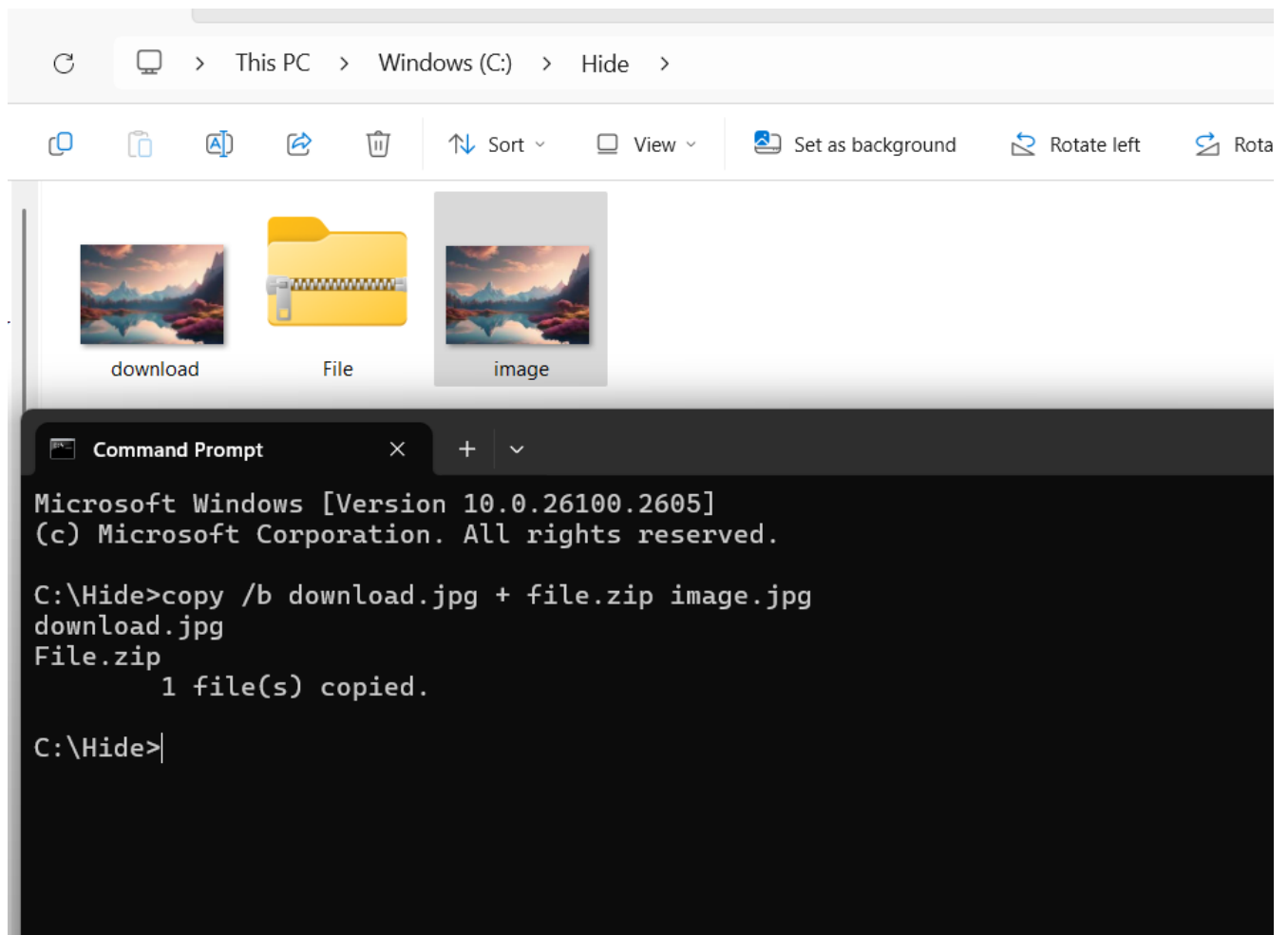


Figure 25:After performing command we create new image with file.zip

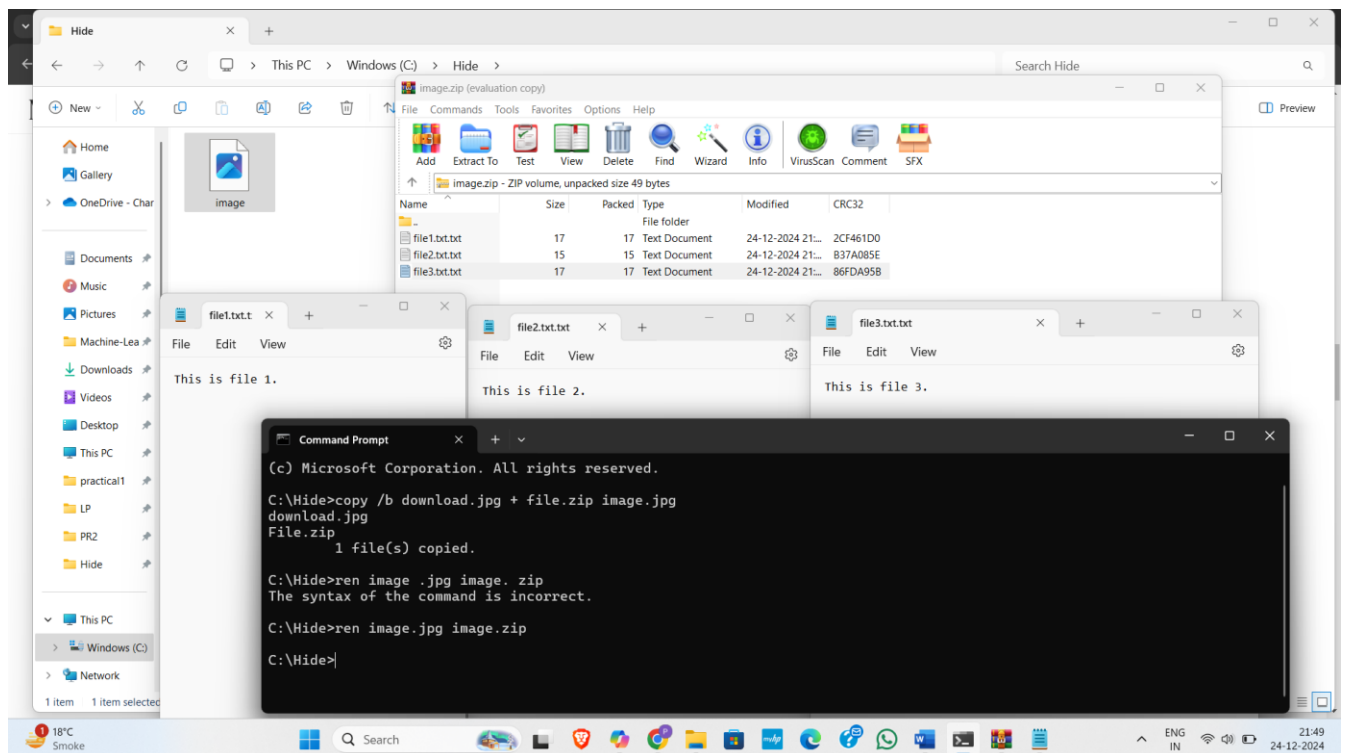


Figure 26:After change image type we can get our zip file with data

**LATEST APPLICATIONS:**

- E-commerce skimming
- SolarWinds
- Industrial enterprises
- Blockchain and Cryptocurrency
- Social Media and Messaging Apps

**LEARNING OUTCOME:** In this practical, we learn about Steganography and tools for steganography like steghide, Steg Suite, Xiao Steganography, using simple CLI and also write python code for steganography.

**REFERENCES:**

1. EC-council : <https://www.eccouncil.org/Steganography>
2. Kaspersky: <https://www.kaspersky.com/Steganography>
3. GeeksforGeeks: <https://www.geeksforgeeks.org/Stegosuite-in-linux/>
4. YouTube: <https://www.youtube.com/watch?v=xepNoHgNj0w>