Steganography

Steganography is basically the scientific methodology of hiding information in form of images, audio files and texts. It is often being used by people to convey secret messages. As anticipated, steganography can be used both for good and bad purposes and Criminals use steganography to communicate their secret messages publicly, ensuring that a public image may not reveal more information than what the public image presents. In this project, I intend to focus on steganography from a technical perspective and demonstrates how it can be performed using OpenStego.

Performing Steganography Using OpenStego in my Home Lab

Lab Objective

This lab session demonstrates the steps involved in performing steganography using OpenStego. Upon completion of this lab, you will be able to:

- Learn how to hide a file within an image using OpenStego.
- Understand the importance of encryption and password protection in steganography.
- Export hidden files in images.

Tool

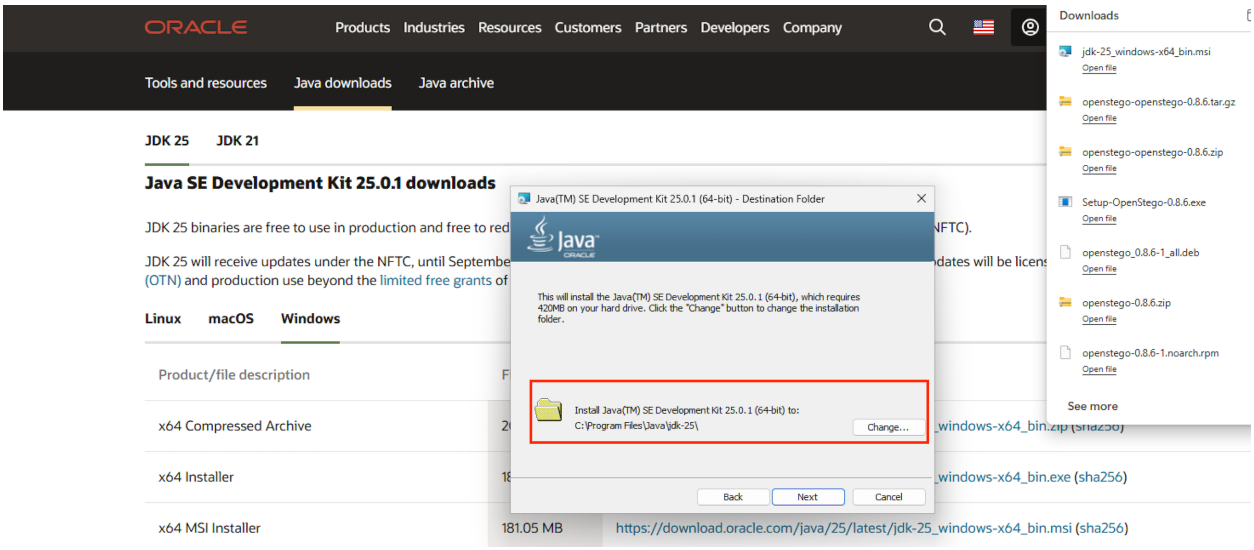
OpenStego is a free and open source steganography software that allows users to hide secret messages inside digital images or audio files.

Step 1

- I installed the latest version of OpenStego software (Setup-OpenStego-0.8.6.exe) on my Windows VM

Step 2

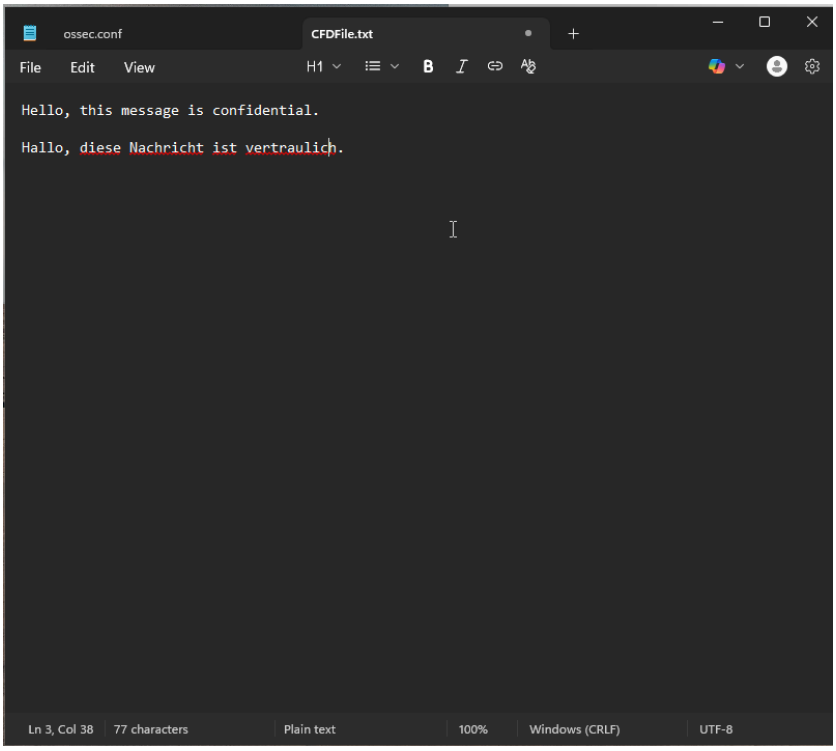
- To avoid encountering errors launching the program, I installed the latest version of Java Runtime Environment (JRE) {x64 MSI Installer > jdk-17.0.17_windows-x64_bin.msi} because OpenStego requires JRE to run.

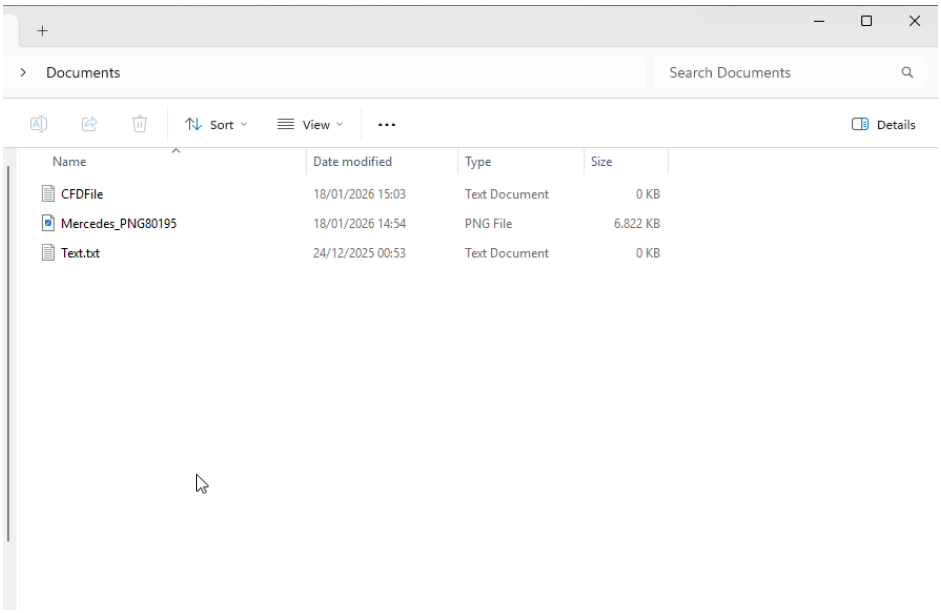


Hiding a File in OpenStego

Step 1

- I created a text folder named **CFDFile.txt,** composed a text in the text file, and uploaded an image named **Mercedes.png** in **png** format in my document.



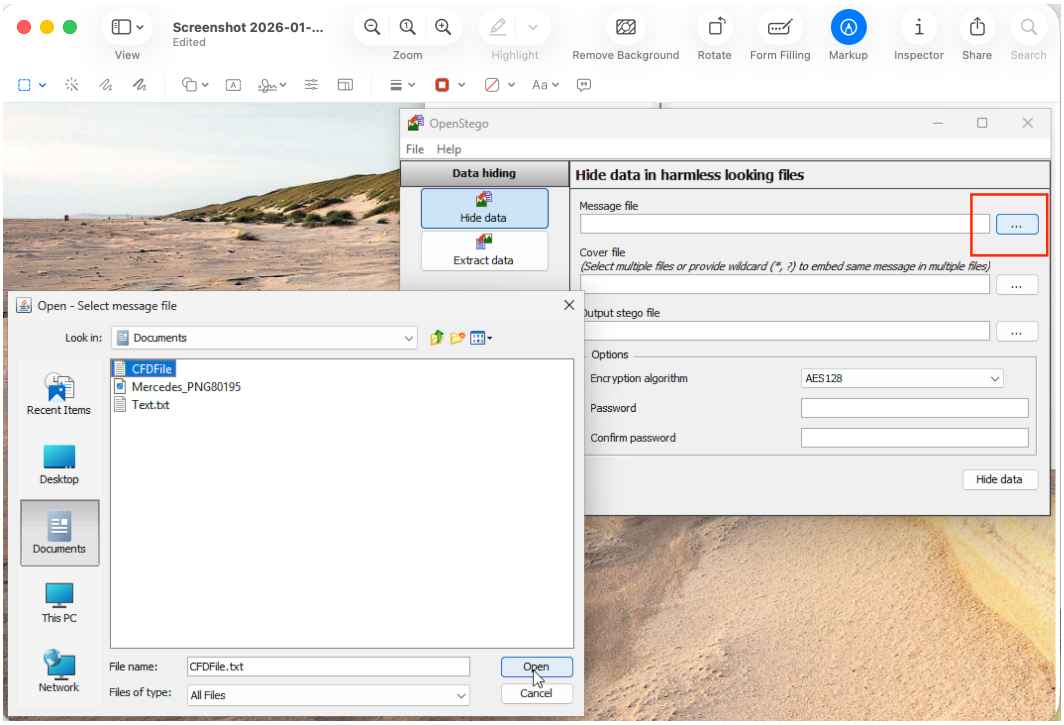


Step 2

- I accessed the OpenStego application by double clicking it on the Desktop or wherever it might have been installed.

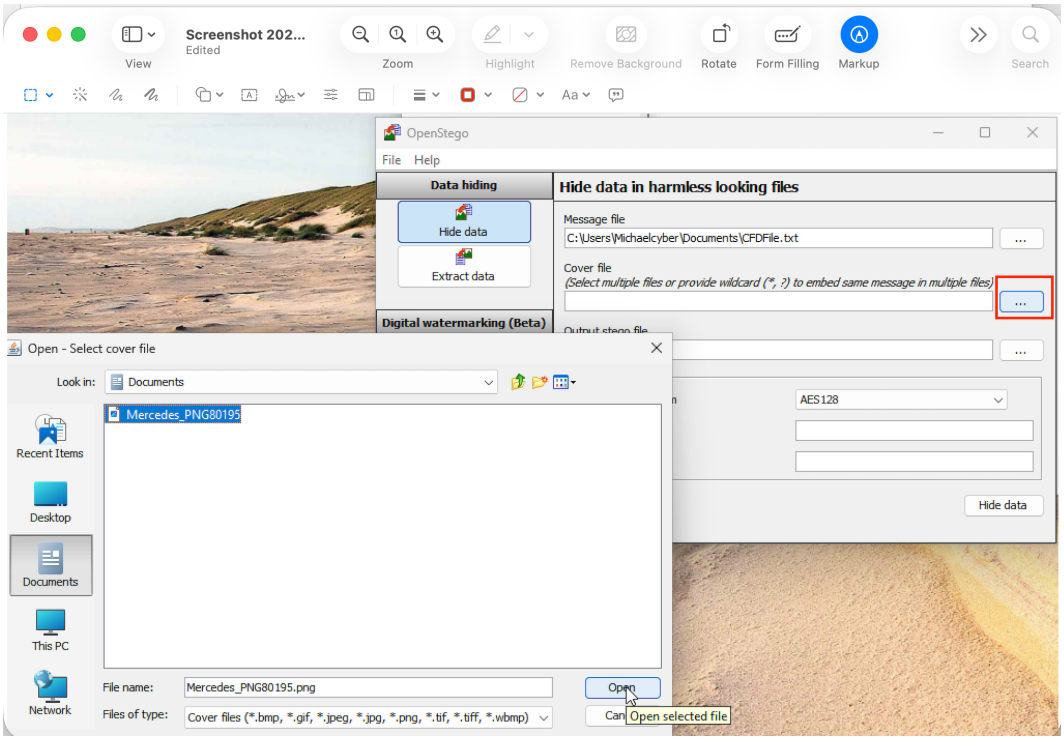
Step 3

- In the OpenStego window, under Message file, I clicked the three dots (...) icon and double-clicked CFDFile.



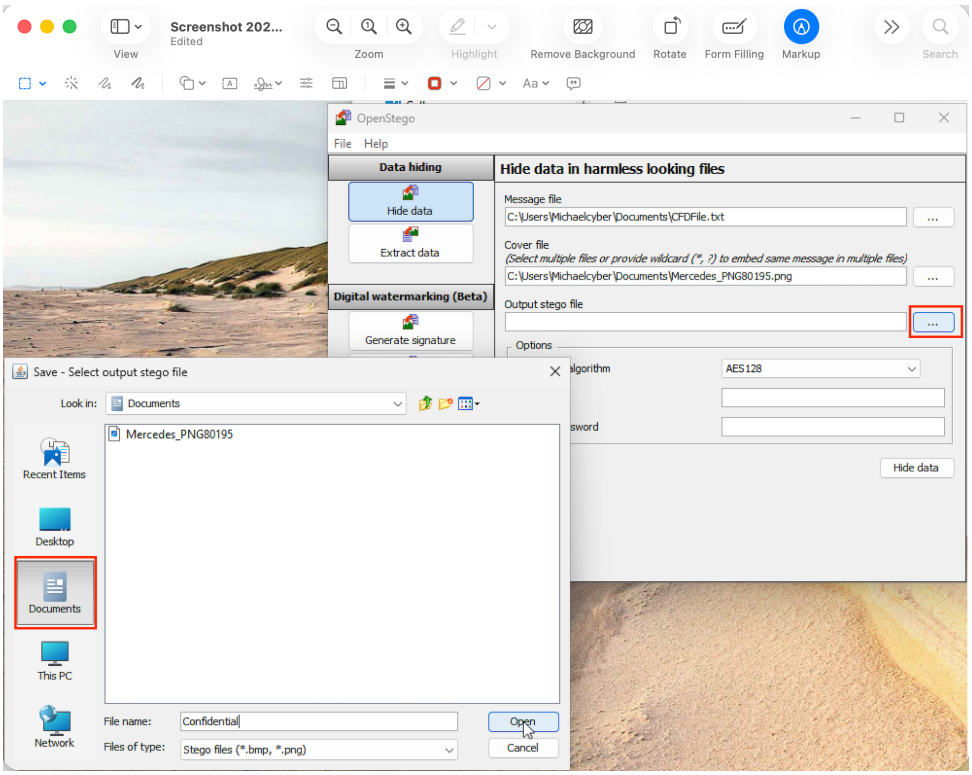
Step 4

- In the OpenStego window, under Cover file, I clicked the three dots (...) icon and double-clicked Mercedes.png



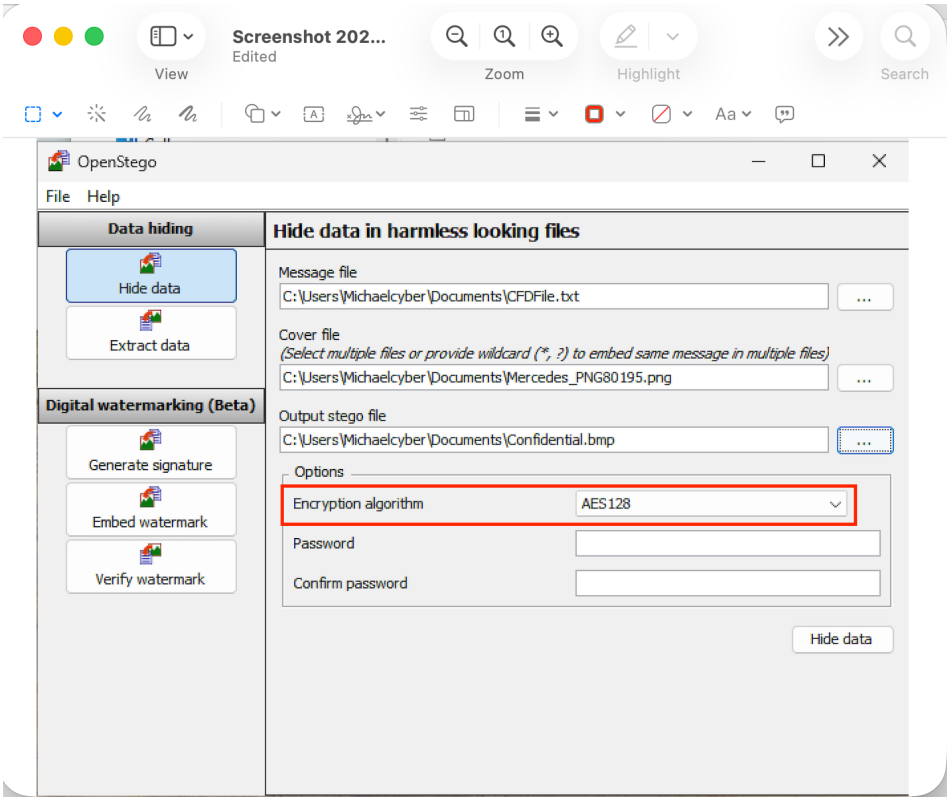
Step 5

- Under Output stego file, I clicked the three dots (...) icon
- I selected Desktop, typed "Confidential.png" and clicked Open



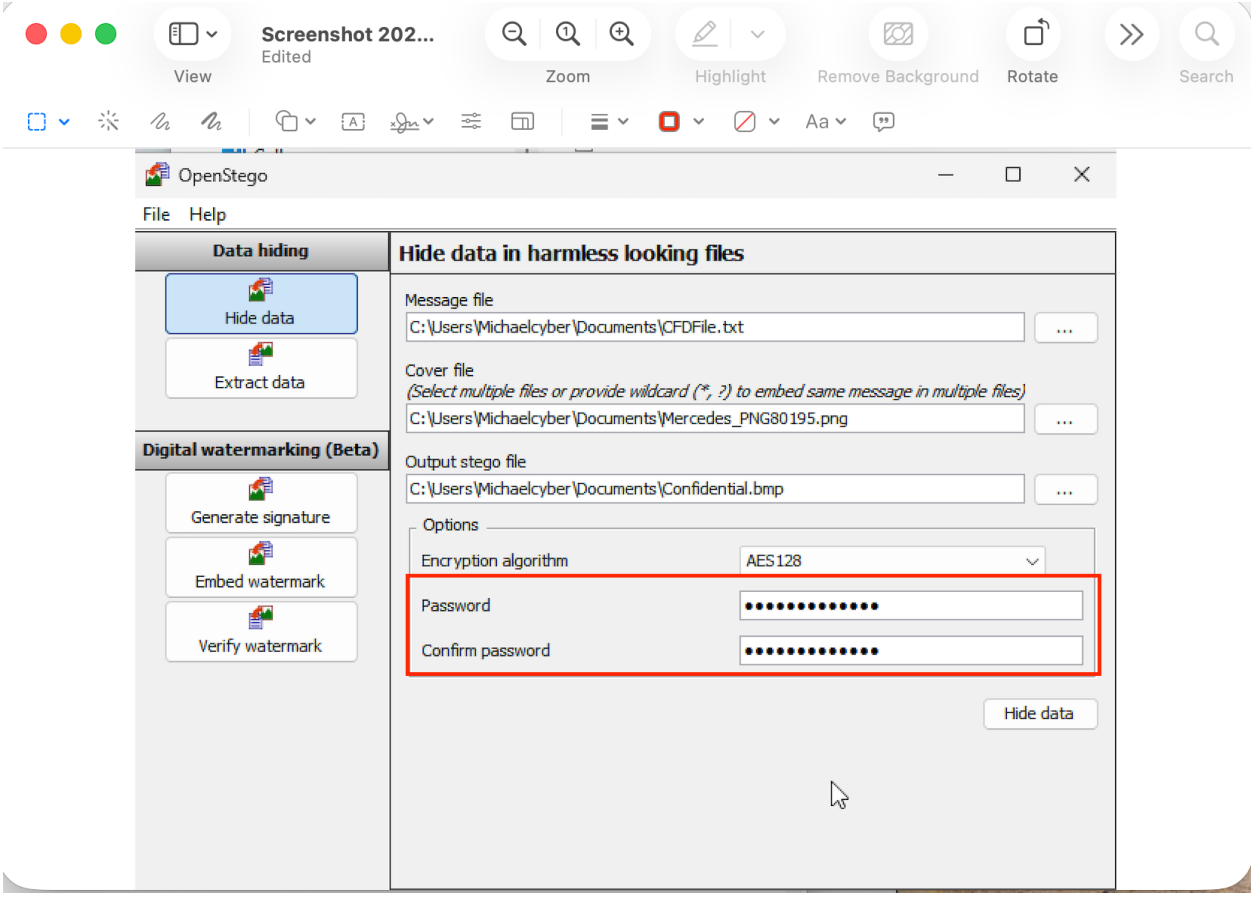
Step 6

Under Options, I ensured that the Encryption Algorithm AES128 is selected



Step 7

- Under Password, I typed my password
- Under Confirm Password, I retyped my password

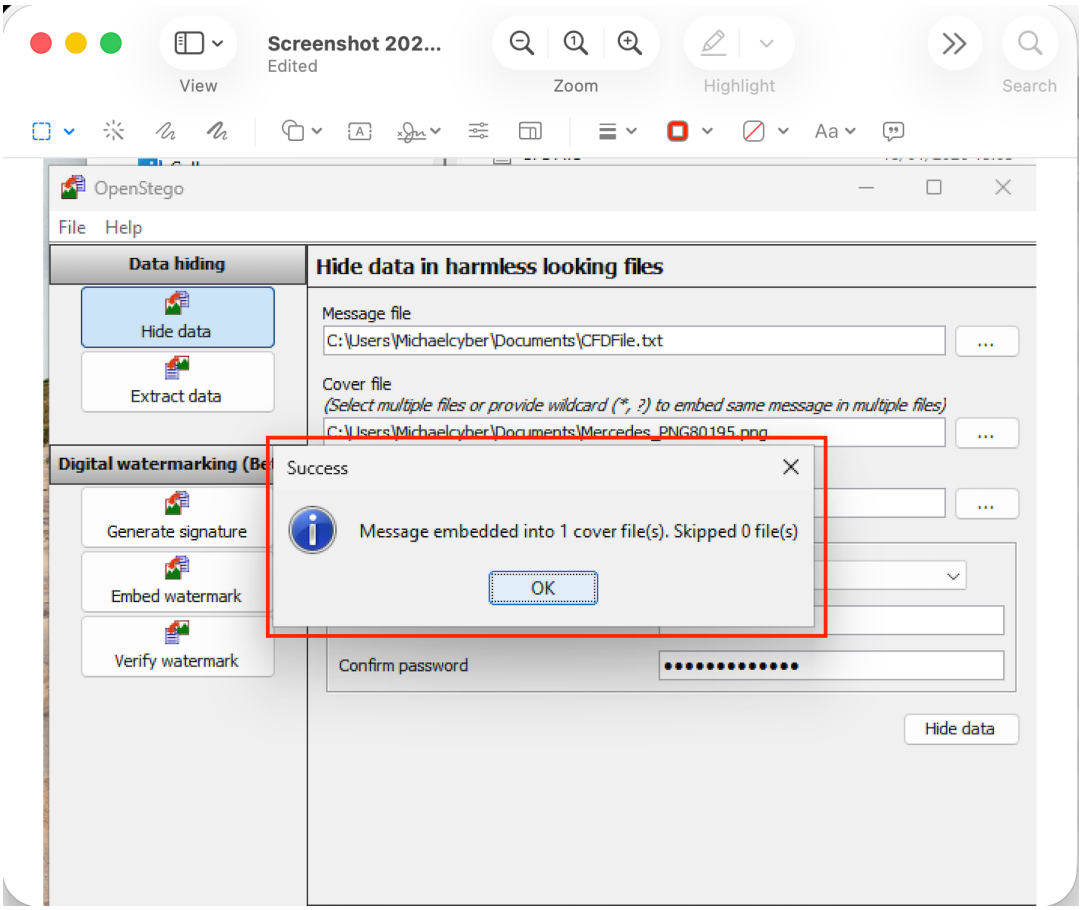


Step 8

I clicked Hide Data

Step 9

Popped-up Message shows "**Message Embedded into 1 cover file(s). Skipped 0 files(s)**"



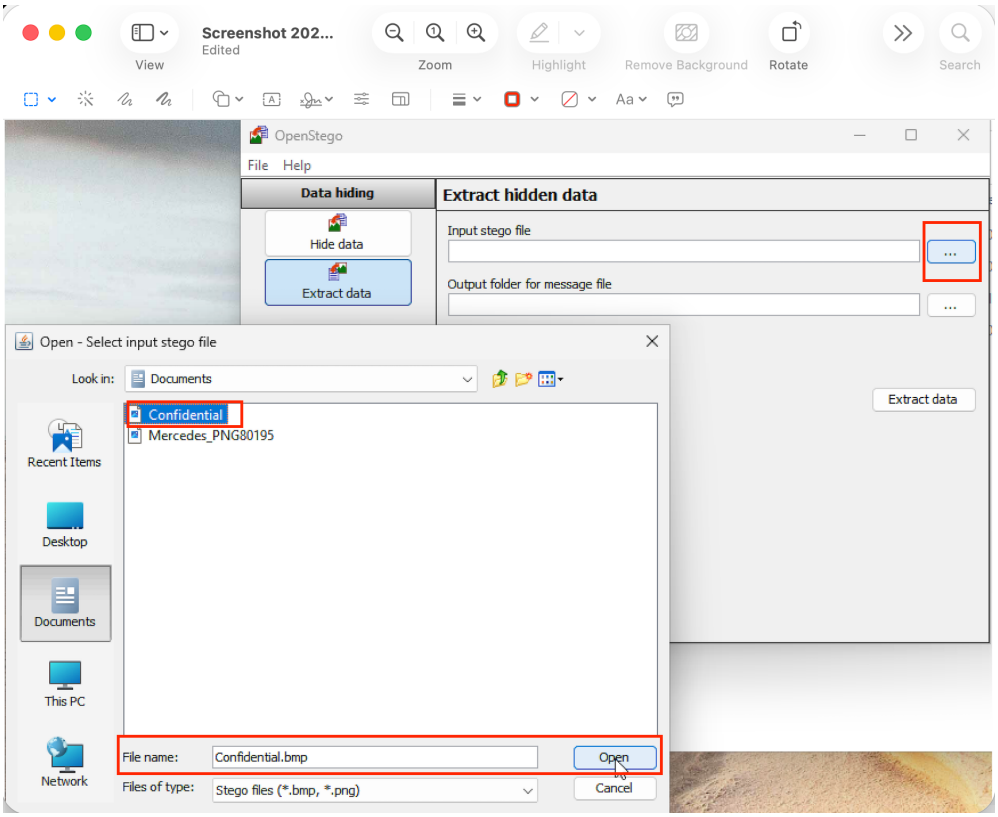
Stage 2: Extracting the File in OpenStego in my Home Lab

Step 1

- In the left pane, under Data hiding, click Extract data.

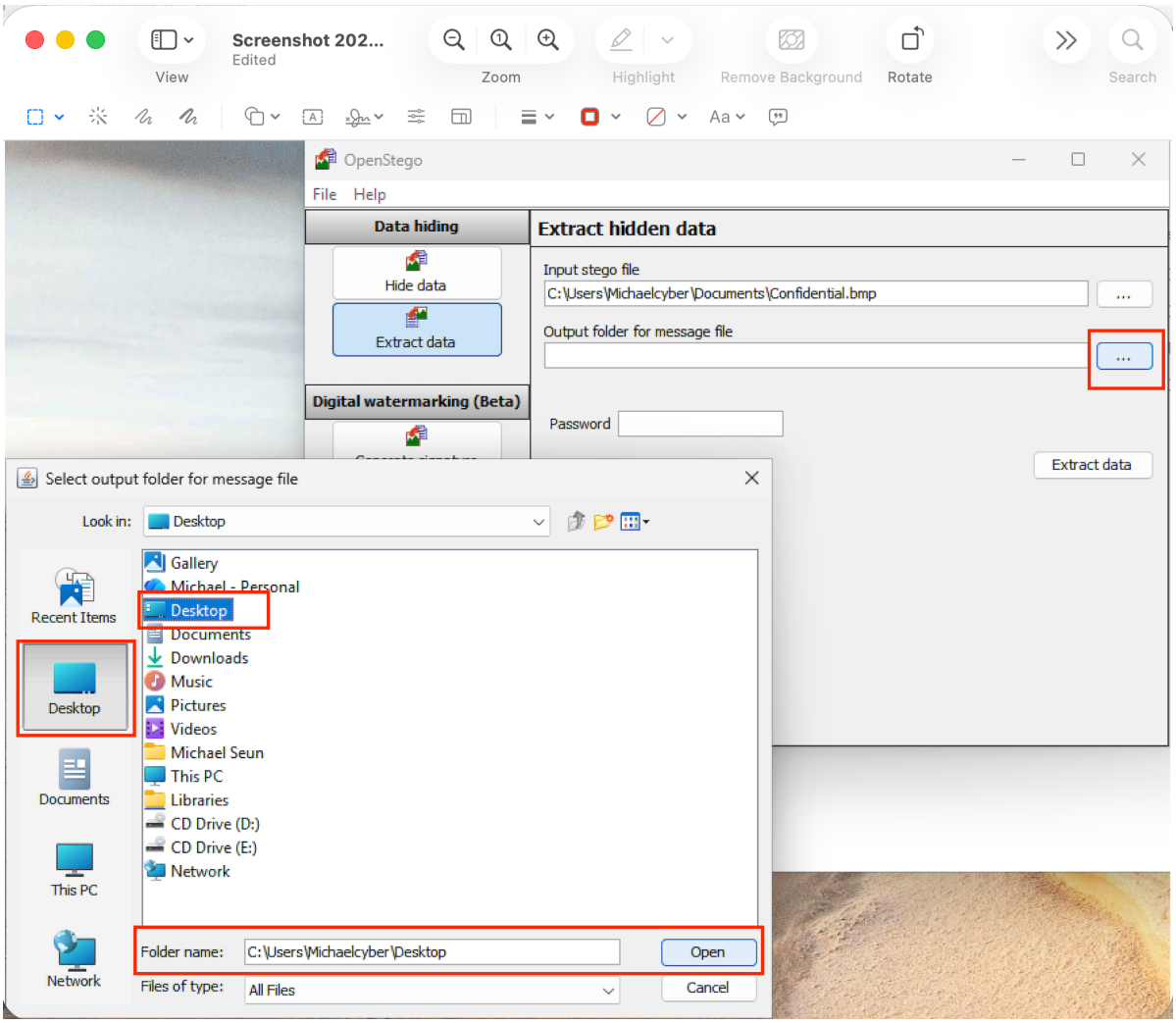
Step 2

Under Input stego file, I clicked the three dots (...) icon and I double-clicked the Confidential image



Step 3

Under Output stego file, I clicked the three dots (...) icon and then in the left pane, verified that the location is selected as Desktop and clicked Open.

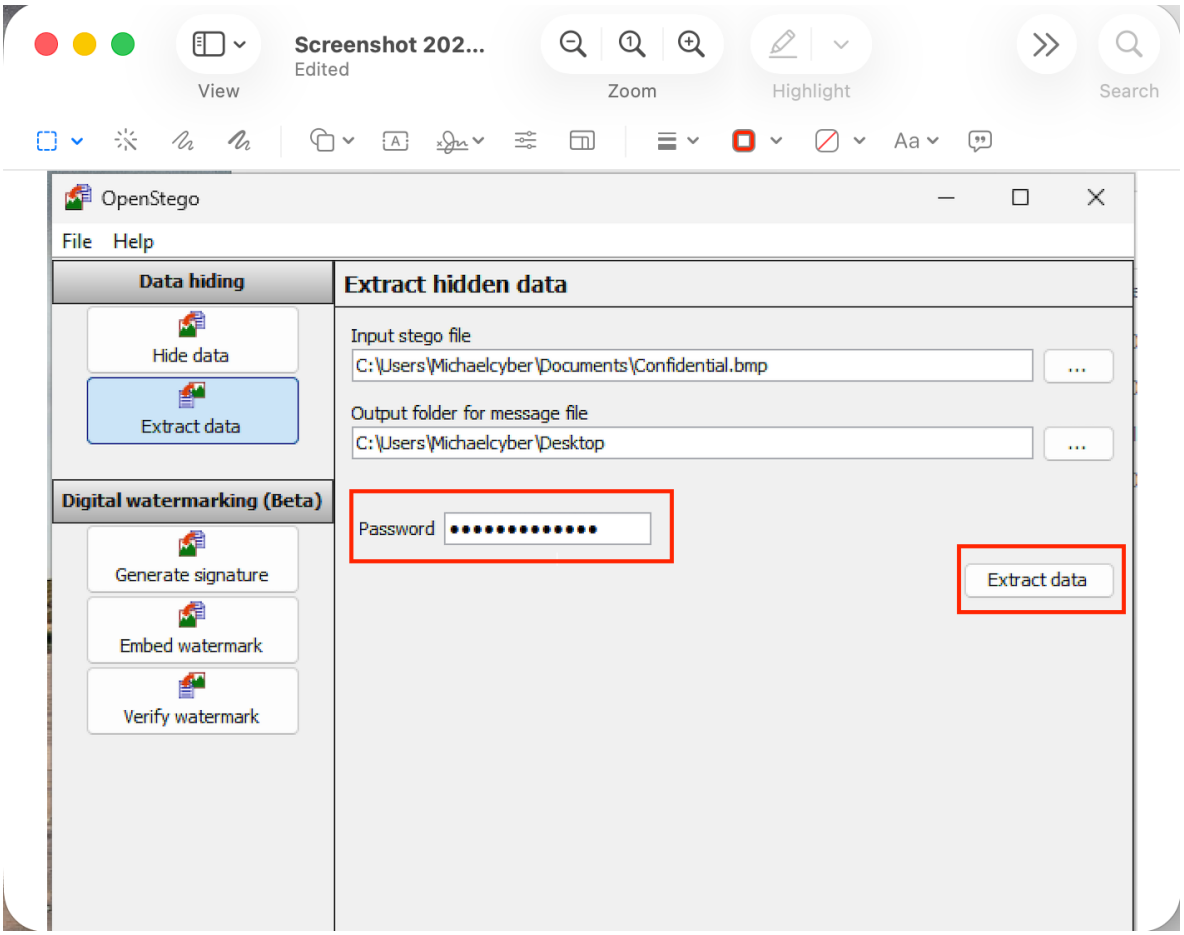


Step 4

- Under Password, I typed my password

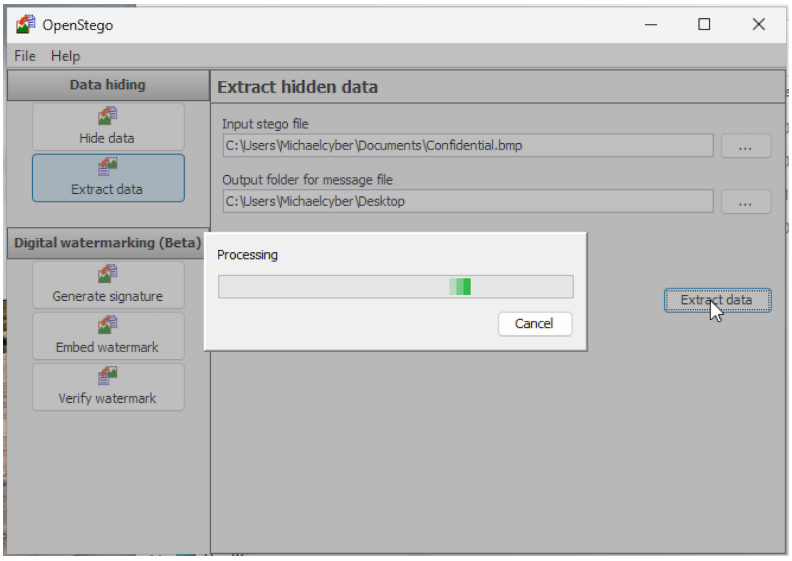
Step 5

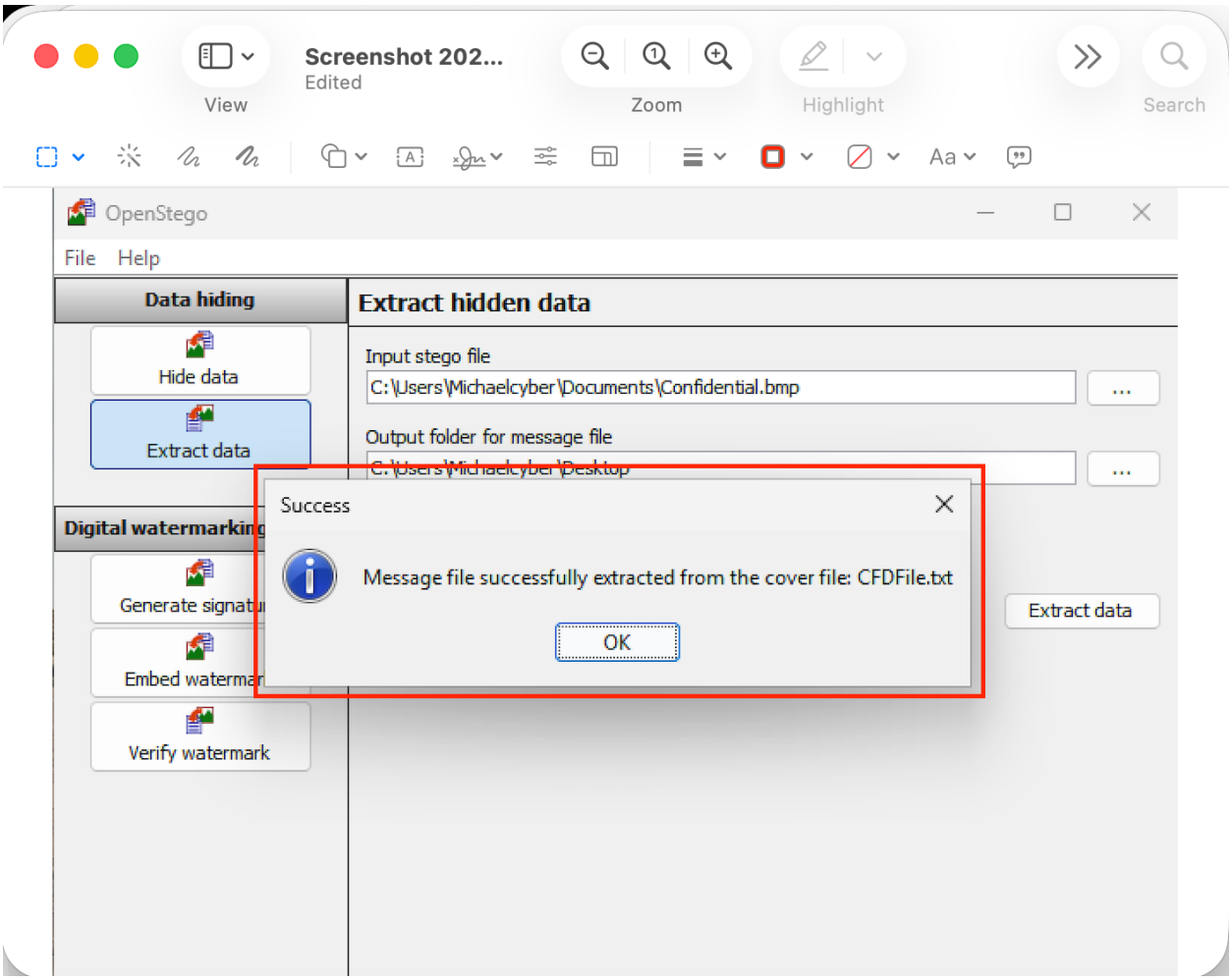
Then I Clicked “Extract Data”



Step 6

- When the Success prompt appears, I clicked OK and, on the **Desktop,** I double-clicked **CFDFile** to open it and observed the hidden text.





Lab Summary

In this project, I was able to conceal sensitive information within an innocuous-looking file. This experiment and the exploration of one of several features of **OpenStego** as a free and open source steganography software that allows users to hide secret messages inside digital images or audio files, equipped me with the knowledge and skills to perform steganography and understand the importance of using appropriate Cryptographic Solutions.