

Roy Soldin 204542179
Noa Levy 308277763

Mini Project – Network Security

Using steganography in order to hide files inside a PNG image.

Tool Objective:

In this project we implemented a script which allows to hide files inside png images, using a well known steganographic method: hide the data in the least significant bits of an image pixels. This produces little changes to the image that usually aren't noticed by just looking at the image.

Project Challenges

While working on this project we encountered a number of difficulties that we had to deal with, the main one was handling media files - which we did not deal with in the past and in addition how to decode information from a collection of bits. We also had to practice on performing bitwise actions, from within the file and into the file. In order to deal with the image file media we had to look for a suitable modules that we could work with in a convenient and efficient way.

Weakness and Strength:

Strength:

- Encrypting a file using our tool is very simple for the user as well as extraction. The interface is conveniently built and friendly (CLI interface).
- Identifying the encryption is almost impossible and therefore meets our main goal we set for ourselves in our work planning.

- The tool does not overwrite the original image, but produces a copy, thus enabling reusing of the original image.

Weakness

- Very long messages cannot be encrypted.
- Our tool supports png images only.

Description of The Service work and components

Our tool is composed from 4 main components:

1. **Main class** : in charge of all user interface and general flow of the program.
2. **Encode Decoder** : responsible for encoding and decoding logics.
3. **File Handler** : responsible for reading, writing, breaking and assembling file bytes of our secret files.
4. **Image Handler** : responsible for reading and writing image files

Proof of Concept

1. The option `--help`(or `-h`) to learn how to use the tool:

```
C:\Study\4 year\netSec\Steganography-project>LSB_Steganography.py -h
usage: LSB_Steganography.py [-h] [--encode | --decode] [--image IMAGE]
                             [--file FILE] [--output OUTPUT]

Hide files inside a PNG image and extract them back

optional arguments:
  -h, --help            show this help message and exit
  --encode              encode the data into the image
  --decode              decode the data from the image
  --image IMAGE          Path to an image to use for hiding or file extraction
  --file FILE            Path to the file with the data to hide or to extract
  --output OUTPUT        Path where to save the encoded image.
```

2. Encoding data inside an image:

Inside the project we have a folder with image and data file for testing. This is the text in the data file:

```
C:\Study\4 year\netSec\Steganography-project>type ImgAndData\data.txt
This is a test
```

Now we are going to use the tool with the encode option:

```
C:\Study\4 year\netSec\Steganography-project>LSB_Steganography.py --encode --image
ImgAndData\original.png --file ImgAndData\data.txt --output ImgAndData\out.png
```

Now we got the two images, the original and the encoded:

Original:



Encoded:



3. Decoding the data from the encoded Image to a new file:

```
C:\Study\4 year\netSec\Steganography-project>LSB_Steganography.py --decode --image  
ImgAndData\out.png --file ImgAndData\outdata.txt
```

Now if we check the outdata.txt file we will get:

```
C:\Study\4 year\netSec\Steganography-project>type ImgAndData\outdata.txt  
This is a test
```