Cuprite-CTF: Image Steganography
50.020 Security

1 Objective

- Write a simple python script to extract text information within Portable Network Graphics (PNG) files via the Least Significant Bit (LSB) algorithm.
- Manipulate output from the Portable Network Graphics files to decode information from substitution cipher.

2 Background

You are an invigilator for a security exam and whilst the exam was taking place, you realised that the students were passing images to each other via airdrop. Being very curious about what these images contain, you decided to not flag them out and attempt to crack the hidden message behind those images after the exam. The first thought that came to your mind was image steganography.

- In order to embark on your quest, you have to first read up on steganography, steganalysis and the LSB algorithm. These are highly recommended as they will give you an overview of how steganography works in general, with the basics in LSB steganography, assuming that LSB steganography would be the only way to go since the students were taught only that in class:
    a. https://en.wikipedia.org/wiki/Steganalysis
    b. https://en.wikipedia.org/wiki/Steganography
    c. http://www.aaronmiller.in/thesis/

- Read up on binary operations and file manipulation as they will greatly help in this exercise:
    a. http://www.tutorialspoint.com/python/bitwise_operators_example.htm
    b. https://docs.python.org/2/tutorial/inputoutput.html

3 Pre-requisites

You would also require an external library for images. We recommend the use of the python PIL library, for more information: **http://www.pythonware.com/products/pil/**

4 Part 1: Implementing LSB to Reveal Information

While sniffing out suspicious high traffic on the network, you realised that a unique set of 5 images were constantly being sent around by your students. Among the 5 images, only 1 image seems to contain the actual information and the rest are decoys.

- Use Python script LSB.py as a template to implement least significant bit extraction on the given image files, with the link below describing the method of embedding in “Image Based Steganography”:
    a. http://www.ehacking.net/2012/09/steganography-tutorial-image-concept.html
    b. http://www.codeproject.com/Articles/5524/Hiding-a-text-file-in-a-bmp-file

- Implement LSB extraction on the 5 RGB PNG images provided to identify the image that has been manipulated and the corresponding extracted information.

- Hint: LSB is considered a general approach to extract/embed data. For this scenario, the students have seemed to have used a pattern in hiding bits of the information. Each pixel (8 bits * 3 for RGB) has at least 1 bit of useful information to be extracted by LSB extraction.

  Given a pre-analysis with the professor, he mentions that the embedded information can be found from the first byte of the first pixel onwards and this information is spread across the image by a fixed number of bits in each interval.

6 Part 2: Extraction of Flag

- Through the extraction procedure, the information is written in binary and appears to be a .mystery file. This is because the extraction only reveals the information in the file but not its type.

- Hint: Discussing with the professor, he mentions most binary data should have some sort of recognizable text at the top of the file that hints towards the format of the binary data or the binary file name itself.

- Try to tinker with the output binary data to finally retrieve the information. You’re are allowed to use any tools required to achieve your goals.
- Hint: Highly recommended to use multiple computers to process the file if required.

7 Hand In

- Submit the captured flag message.