Steganography Library

Created by Paul Mbusa – Updated 5/13/16

# Description

This library was created for my Computer Networks course as my final project. The assignment had very few restrictions as the only requirements were teacher approval and use of networking.

# Background

The inspiration for this project came from ta real world need of mine. I was part of an organization whose main means of communication was through Groupme. Many confidential things were discussed in the chatroom and at some point we had an issue where somebody was leaking information to people who weren’t authorized to possess said information. So the problem at hand was finding out who did it. Rats are very fast, sly creatures; if you have ever tried to chase and catch a rat, you know it’s impossible. However, it is very much within the realm of reality to set a trap and let the rat come to us, that’s what this library was created to do.

# Description

This proof-of-concept project uses the Spark framework to run a webserver which hosts a simple website. Every page on the website was to be generated on the fly, as such each time the page was loaded it could be different. During the page generation, the IP address of the client is pulled from the HTTP request and encoded into the image. Commonly used by spies, Steganography is the art of hiding something in plain sight. This sort of image post-processing doesn’t look any different from the source material but under the right light, reveals far more than at first glance

A pixel is most easily represented as a 3-tuple of the red, green and blue light values in the range of 0-255; this is the range of a 2 digit hexadecimal number. By modifying the least significant digit of the color value for each pixel, a message can be hidden in the image without too much distortion to be noticeable. The clients IP address is stored in this image and can be retrieved at a later time using a decoder.

My algorithm also has a randomization element to it so that even if two identical strings are stored in the same