Tyler Jackson

4/27/2015

Project 4 : Steganography

This project was a lot more difficult than I thought it would be. Initially, because the concept of steganography is somewhat simple to understand I assumed in practice it would be similar, but this was naive. I figured finding the passwords would be as simple as looking at the image and trying to find an irregularity and then finding that spot in the file and then decoding it to get the password. Unfortunately, these images all seemed normal to me. I opened them in an open source image editor called Gimp. Using Gimp I adjusted tons of the image settings including hue, saturation, brightness, opacity, etc. I tried to convert the image to gray scaled, or to view the image as layers and remove extra layers, but none of these seemed to lead anywhere. Fortunately, we knew the format of the passwords, so I figured I would try to open the bitmap file in a text editor and see if maybe I could see an irregularity from that perspective. Unfortunately these files were all relatively big or extremely large, so it wasn’t as trivial as just scrolling through the file. I spent a while trying to write something to extract the least significant bits from the bitmap, as I figured you would probably have hidden at least one of the passwords in those bits, but when I tried to read in the bits from memory they seemed to come in as special character even if they were just hex in the file. This was confusing, and I couldn’t figure out how to read in the hex values as they were shown in the bitmap file.

I then spent a while looking into the various file formats of bmp files and was able to determine a little more information about the moon landing bmp file. The bmp file has various fields set in the first bytes that tell us information about the type of bmp file we are dealing with. The first 2 bytes are 0x42 0x4D to identify the bmp file. The number of bits per pixel are in byte 28, and the compression method used is in byte 30. The number of bits per pixel is 0x18 or 24. The compression method used is 0x0000 or no compression. After spending a while on this file I decided to try some of the others.

The baby feet image also perplexed me. I tried to change as many of the image characteristics as I could, but nothing seemed to reveal anything useful about the photo. Initially I thought the apple would be the easiest as the colors are all very uniform which I thought would make any adjustments to the color components very obvious, but alas I still wasn’t able to figure it out.

One thing that really surprised me about this project was how much documentation there was online about steganography in general, but how little documentation there was in regards to tutorials and actual walk-throughs.

For my own hidden message I hid the word Obama in the least significant bits of a bmp file. I started by finding an image with a lot of detail and then looking for a section where the color components were non-uniform. Then I looked at the least-significant bit of each byte and either changed it up or down 1 to match the next bit in my hidden message. Since OBAMA takes 40 bits in binary I changed the least significant byte of 40 bits. The 40 original bytes in my image bmp file were

1011 1312 1313 1313

1515 1513 1414 1412 1312 1110 1211 110f

110d 0f0f 0d0d 0d0c 0d0b 0806 0503 0404

and I changed them to

1011 1212 1313 1313

1415 1414 1414 1512 1213 1210 1212 100f

100d 0e0e 0d0d 0e0d 0c0b 0806 0402 0403

OBAMA in binary is

01001111 01000010 01000001 01001101 01000001.

ORIGINAL ALTERED

 