

Stegasploit Assignment Questions

You should use a Linux Machine for this assignment. Ubuntu or Kali with Firefox will be sufficient. I am sure many groups will be providing a Kali image so I will not be providing the image (you can simply install the image from the official website anyways) and those of you who have Ubuntu installed on your machine, you can use that. You could try using Windows but the commands and instructions below will be different and I will not answer any Windows Specific questions as I have no access to a Windows Machine.

Assignment Original Document (If you are looking at this, you have a copy anyways. But I may update the assignment page)

<https://docs.google.com/document/d/1u-82tvRmDRlvTzdI0OZMdwuuBMb-lpCvgkMFRi3FybQ/edit?usp=sharing>

Assignment files should be provided by Arnold. If not, please contact me:

juhong.kim@mail.utoronto.ca

Files Provided:

- members.txt
- answers.txt
- ilir_polyglot.png
- cat.png
- cat.jpg
- pocorgtfo08.pdf

Files To Submit:

- answers.txt
- layer7.jpg
- layer3.jpg
- encode.png
- polyglot.png
- decoder_cve_2014_0282.html

If you need HELP:

You can email Kim: juhong.kim@mail.utoronto.ca, Kim should respond within a day. Or you can ask on Teams, Albert and Kim will be available to answer your questions.

Note: The server runs using Python 2.7 and not Python 3

Part 1: Encoding Conceptual Questions

Here are some warm up questions to start before working on the toolkit. Answer **2/5** questions (You can answer more but only if you have time)

1. How many passes does PNG require during iterative encoding and why
2. Iterative Encoding: On slide 24, we say $\Delta = M - M'$
What does that actually mean? Please explain in words. In addition how do we know when $\Delta = 0$ on the toolkit (the stegosploit toolkit) when we run the iterative encoding. (Hint: Try running iterative encoding on a jpg using the stegosploit toolkit (instructions in part 2) or look at slide 25 and observe what information we know when iteration completes)
3. What is a good indication that the iterative encoding will converge (hint look at slide 25 or just think what does it mean to converge)
4. Is JPG encoding and decoding cross browser support? And why?
5. Why would a large code exploit may not be able to be encoded into a small image?

From tutorial in **Part 2**, you needed to extract the toolkit, you will be using the toolkit for the next following questions

In case you do not have the toolkit, the pdf has been provided. Here are some steps:

```
unzip pocorgtfo08.pdf stegosploit_tool.png  
mv stegosploit_tool.png stegosploit_tool.html
```

Open firefox and open the file

Click on the image and you should be able to download the toolkit

Part 2: Encoding

To start off, you will need to start the server to run any of the tools

Make sure you are in the folder of **stegosploit-tools/stegosploit**

```
python -m SimpleHTTPServer
```

Go to the folder from the assignment named **provided** and copy all the provided images to **stegosploit-tools/stegosploit**

```
$ pwd
/home/zaku/Downloads/stegosploit/stego_assignment/provided/
cp cat.jpg stegosploit-tools/stegosploit
cp cat.png stegosploit-tools/stegosploit
cp ilir_polyglot.png stegosploit-tools/stegosploit
```

Here's an example of me running the commands

```
stegosploit_kim_albert_assignment$ cp cat.jpg stegosploit-tools/stegosploit
stegosploit_kim_albert_assignment$ cp cat.png stegosploit-tools/stegosploit
stegosploit_kim_albert_assignment$ cp ilir_polyglot.png
stegosploit-tools/stegosploit
```

You should have all the three images in the :

```
$ pwd
stegosploit/stego_assignment/provided/stegosploit-tools/stegosploit
$ ls | grep -E "jpg|png"
cat.jpg
cat.png
ilir_polyglot.png
```

Step 1: Encode in different layers (JPG)

On your web browser, visit localhost:8000/stego/iterative_encoding.html

Note: You will need to write the path of the image. The path is relative to the web server

1. Load the provided image (.././cat.jpg)

Encode Image Data on JPG/PNG

Input file:

2. Encode **cat.jpg** with **layer 7**, **grid size** of 3 in the **Blue Channel** (there's no specific reason for choosing this channel)
3. Choose the exploit code to be IE C-Input Exploit
4. Click on the **process** button

Resolution: 960x420 Bit Layer (0-7): 7 JPG Quality (0-1): 1 Grid: 3 ☐ R ☐ G ☒ B ☐ All

Ready to use exploits: IE Cinput Use-After-Free calc v

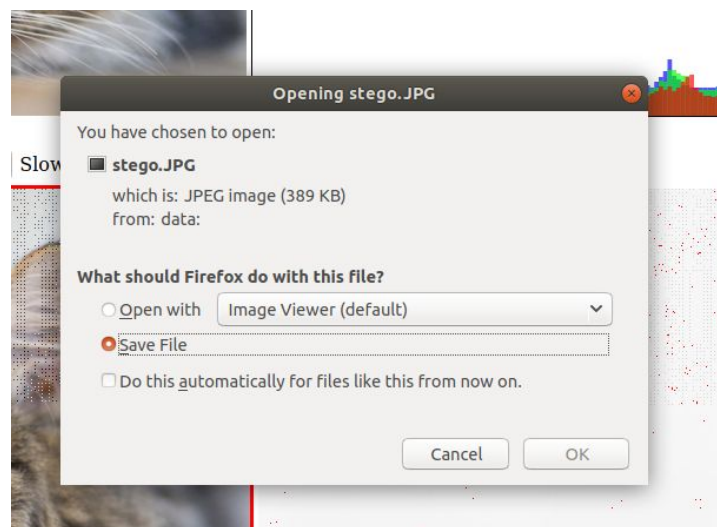
Or supply your own code:

```
function H5(){this.d=[];this.m=new Array();this.f=new Array();H5.prototype.flatten=function(){for(var f=0;f<this.d.length;f++){var n=this.d[f];if(typeof(n)=='number'){var c=n.toString(16);while(c.length<8){c='0'+c}var l=function(a){return(parseInt(c.substr(a,2),16))};var g=l(6),h=l(4),k=l(2),m=l(0);this.f.push(g);this.f.push(h);this.f.push(k);this.f.push(m)}if(typeof(n)=='string'){for(var d=0;d<n.length;d++){this.f.push(n.charCodeAt(d))}};H5.prototype.fill=function(a){for(var c=0,b=0;c<a.data.length;c++,b++){if(b>=8192){b=0;a.data[c]=(b<this.f.length)?this.f[b]:255}};H5.prototype.spray=function(d){this.flatten();for(var b=0;b<d;b++){var c=document.createElement('canvas');c.width=131072;c.height=1;var a=c.getContext('2d').createImageData(c.width,c.height);
```

MD5: 516c9def8b7207299f939b617d3f788f

process

5. Choose the option for slow and click on the **iterate** button
Look at the right side of the image where it shows the deviation (this is just to see visually the encoding process)
6. Save the image




7. Rename the image as **layer7.jpg**
The image is saved to your downloads folder

Here's an example of what I did:

```
kimju10@dh2026pc09:~/Downloads$ cp stego.JPG stegosplit_kim_albert_assignment/layer7.jpg
```

8. Write down the number of passes it went through to encode

The orange dot contains the number of iterations. **Write the answer to answers.txt**

iterate Pass:  Delta: 4712/403200 (1.1686507936507935%) stop Slow Motion: ☐

Notice how the delta is quite high for layer 7.

9. **Refresh** the page
10. Repeat steps 1-5 but on a layer 3 instead and call the file **layer3.jpg**

Step 2: Encode a PNG File

1. On the same page as part 1, encode a png file on a layer that doesn't have any noticeable distortion. **Ensure to refresh the page before starting.**

2. Save the image as **encode.png**
3. Write down the number of iteration into **answers.txt**

Part 3: Add Decoder to the encoded PNG image

YOU WILL NEED TO FINISH STEP 2 PART 2 TO WORK ON PART 3

1. Modify **exploits/decoder_cve_2014_0282.html** and ensure the following variables are set correctly as those you set when you did encoding for **encode.png**:

bL: the bit layer where the code was encoded to the image

eC: is the channel we are decoding from

0: red

1: green

2: Blue

3: all channels (use this for grey scale images)

gr: the grid size

YOU WILL BE SUBMITTING THE **decoder_cve_2014_0282.html**

2. Run **html_in_png.pl** under **imajs** directory to add the decoder to the image **encode.png**. Here's an **example** of how to run the command.

```
perl ./imajs/html_in_png.pl exploits/decoder_cve_2014_0282.html  
../../../../encode.png ../../polyglot.png
```

3. View the file using vim and observe that you can see the decoder in the image but not the malicious code. We will be checking that the decoder is in **polyglot.png**

Part 4: Find Code Encoded in Image

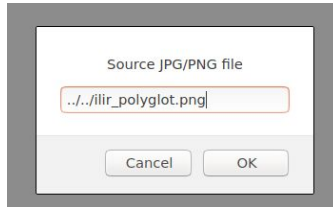
Since running the exploit on a Windows VM is a lot of work, I'll have you see a "decoder" that simply extracts the message from **ilir_polyglot.png**.

Story: You are trying to find proof that Kim has committed an AO with Artem. You view all the logs collected their interaction in social media, email, and sms and the only information that was passed between the two are the following:

- **ilir_polyglot.png**.
- Bit Layer: 3
- Grid Size: 3
- Channel: Blue

Knowing the fact they did a presentation on steganography on images, you are suspicious about the image. Find if there are any code embedded in the image (there is).

1. Go to **localhost:8000/stego/decodedecoder.html**
2. Load the image: **../..ilir_polyglot.png**



3. Bit Layer: 3
4. Color Channel: Blue (which is 2)
5. Grid: 3
6. Click on **decode** button

Decode a Steganographic message from an Image



7. Write the hidden code to **answers.txt**

Answers will be posted after the submission deadline for the assignment.