Right There Solution

Matthew Burket (ISEAGE Lab, Iowa State University)

February 19, 2018

Step One: The Image

You were given this image, see Figure 1, of a poorly draw horse from IScorE.

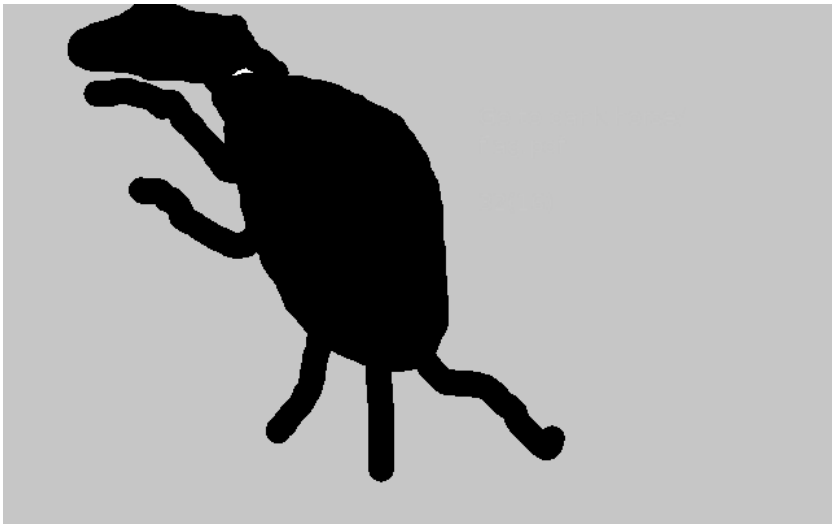

Figure 1: The Given Image

This part involves steganography. The text is off every slightly from the background. We can use StegSolve (direct link to the jar) to find the text. You could also use Photoshop or GIMP to play with the contrast and colors.

Using StegSolve we many images. I'm only including the ones help to get to our goal. See figures 2 and 3.

Step Two: The PDF

See can see the message "Go to dank.horse/flag.pdf" and then couple lines below we also see "32(16)". The first set line is useful. We will download the pdf and open it. Next, we will look at the properties of the PDF, see figure .

We see that there is a link to dank.horse/flag.zip. We will download and look at that.

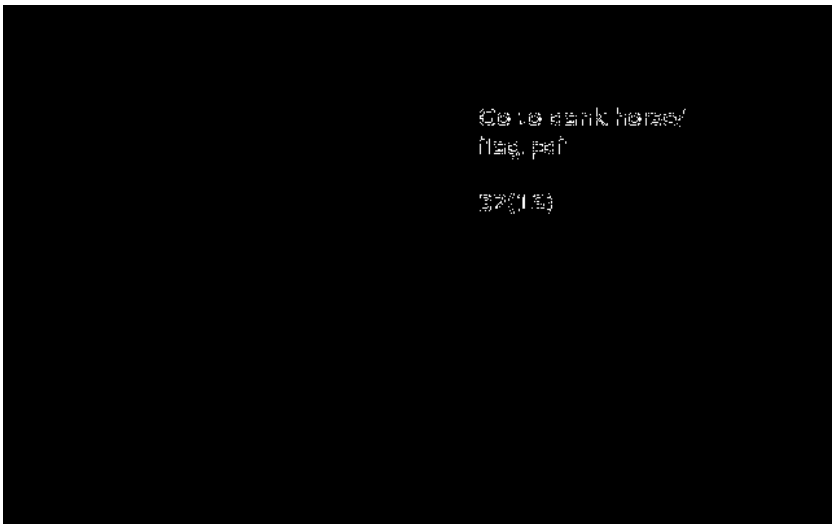

Figure 2: Solution to Stego 1

Figure 3: Solution to Stego 2

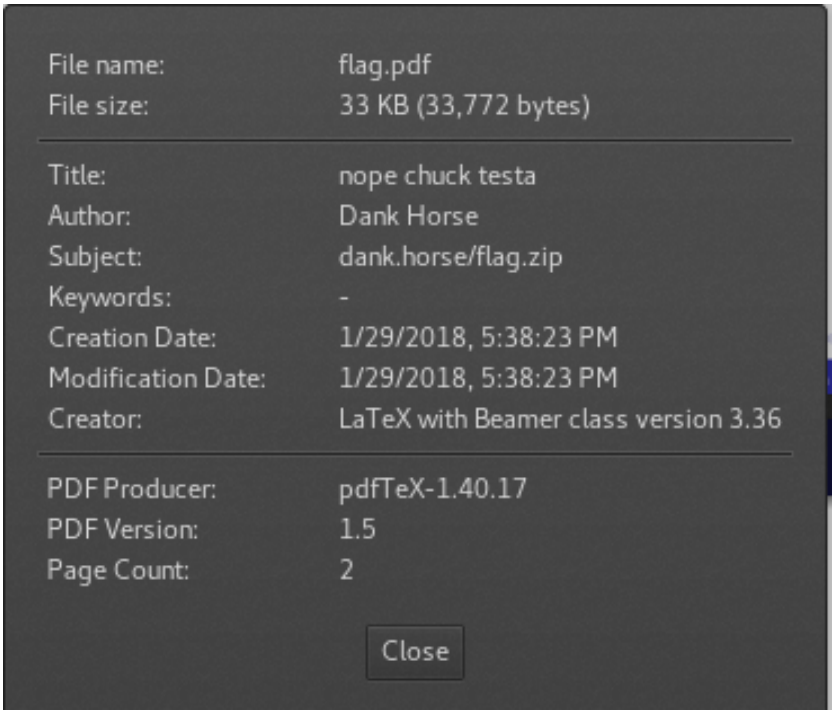

Figure 4: Proprieties of flag.pdf

Step Tree: The Zip

We will download that. If we see what files are in we see that there "flag.pdf". It is the same as the flag.pdf as the first step. So we must look harder. So a fun fact about zip files is that they don't care about stuff that comes after them. So let use `cat` on flag.zip. You could also use `strings`. See Figure 5.

We see something that might of interest at the end of the file. The after the last tilde. So we see that is not in the CDC flag form. However, it looks like it might be encoded. We recall the other text from the first step "32(16)". What if that is how the flag is encoded. Let's try that. So first we will base32 decode on the flag. I will use the python shell. First, we will type `import base64`. Then we will type `b32 = base64.b32decode(flag)`. Then we will type `base64.b16decode(b32)` and the flag appears. See Figure 6.

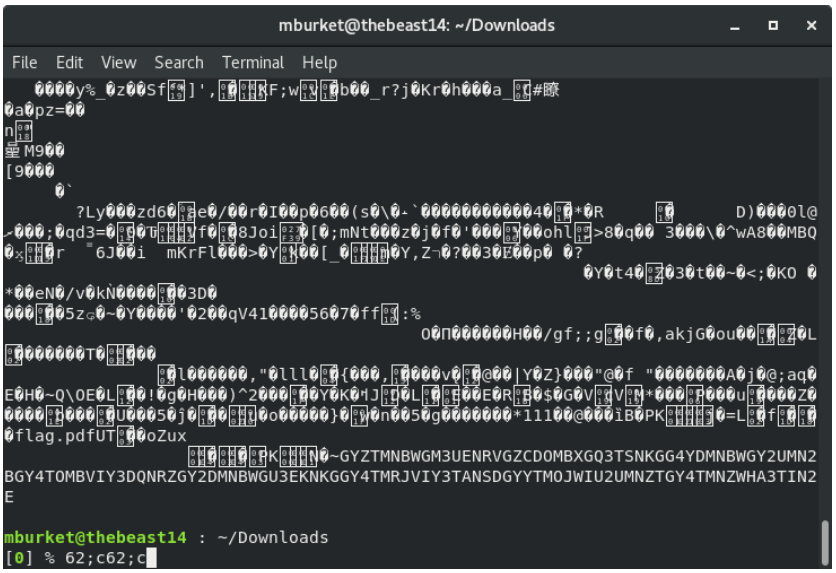

Figure 5: Using `cat` on flag.zip

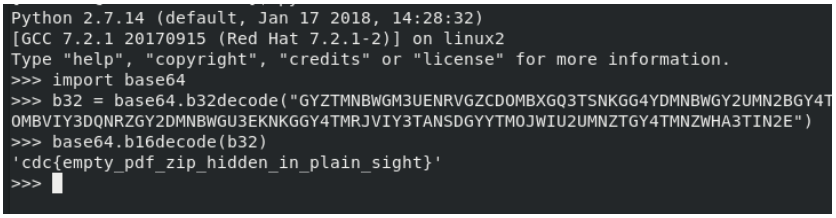

Figure 6: Python Flag Decode