

Collection - 50 points

"a collection of 47 pictures in an art gallery"
definitely not stego

Here is the final art collection!

This is not stego since we've given you the answer in the hints and we don't want to frustrate anyone. However, this is very solvable without hints, so try avoiding hints first!

Download collection.zip below!

Being terribly incompetent at CTFs beyond OSINT challenges, I was thrilled to find out that this challenge gave away what I had to do through the hints. Needless to say, I clicked on them:

Hint



These images look awfully familiar. A single pixel has been modified in each of these images, changing the blue value (RGB) to a character in the flag.

Hint



The coordinate (x, y) represents a pixel on the image. All pixels were modified at $(0, y)$, where y represents the index of a character on the flag!

Hint

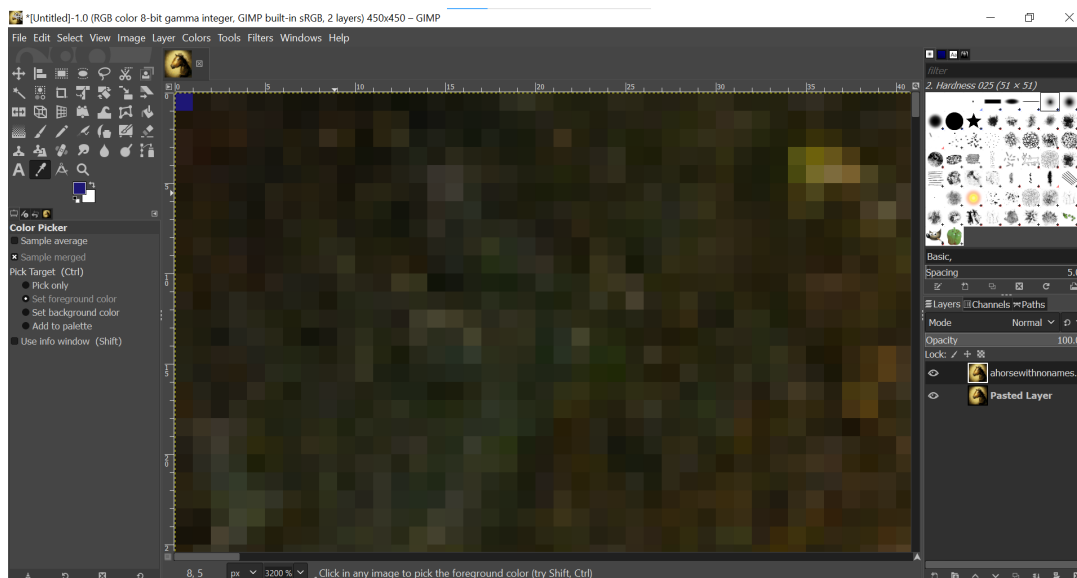


Create a script that successfully takes all the modified blue values after comparing to the original images on the CTFd, and arrange them based on the *y* value!

While I was distraught by the lack of “uiuctf{” in the hints, this was still highly valuable information. I learned that:

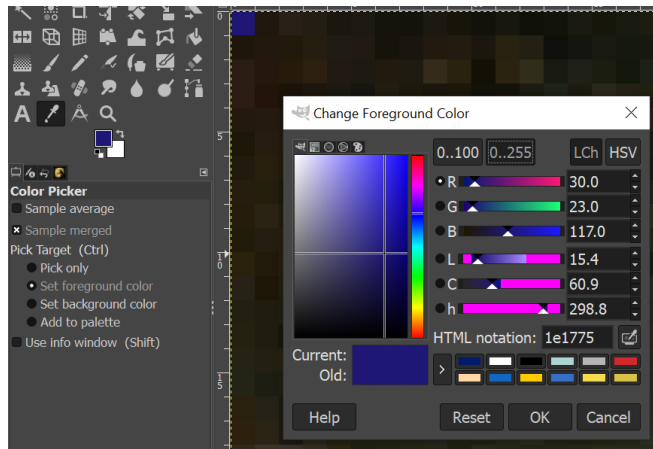
- Each image in the folder corresponded to a character in the flag
- Each character was somehow associated with a modified blue value from a pixel on the leftmost column of the image
- The order of the characters was directly correlated with the *y* value of each modified pixel, with 0 being the top left and going down as *y* increases

From here, a seasoned CTF participant would do what the third hint stated: running code to scrape the original images from the website, finding the pixels that differ between the folder and website images, and arranging the blue values of the folder images by ascending *y* value.



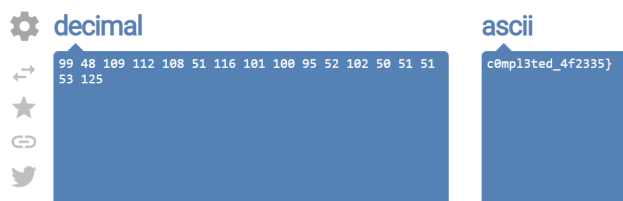
Unfortunately for me, I suffer from a rare condition known as “skill issue.”

Because I didn’t know how to script this kind of challenge, I manually pasted 47 original images into GIMP (cue agonized screaming from computer science majors) and overlapped them with the folder’s images to look for noticeably different pixels along (0, y). I found that the order of the pixels was luckily the same as the alphabetical order of the images, so I didn’t have to do any rearranging (this helped significantly for images where the color difference was practically unnoticeable). With each pixel identified, I used GIMP’s color picker tool to get the color and clicked on it to get the blue value, the case below being 117:



For the sake of organization, I made an Excel spreadsheet to record all my blue values, corresponding y values, and the name of each picture to help me reference back if I needed to. Excel functions made it convenient for me to append all the characters of the flag together after they were collected, and it also would have been easy to convert my blue values to hexadecimal or binary in case it was necessary. After going through every image, I was left with y values going from 0 to 46 with blue values corresponding to them.

I was initially confused how the blue values could construct a flag while working through the images, but realized that each of them neatly corresponded to an ASCII character. I recalled there being 256 ASCII characters, and there were also 256 possible values for blue in decimal form. Since I didn’t know my ASCII characters, [a website](#) allowed me to do the conversion piece by piece whenever I got tired of looking at GIMP. I’ve provided the last conversion I made below:



Converting all my blue values to ASCII gave me characters that were more typical of a flag, and unsurprisingly, Excel gave me a flag after appending everything together:

H1 ✕ ✓ fx =CONCAT(C2:C48)												
	A	B	C	D	E	F	G	H	I	J	K	L
1	y	blue_dec	ascii	pic			flag	uiuctf{Th1s_c0llect10n_is_n0w_c0mpl3ted_4f2335}				
2	0	117	u	ahorsewithnonames								
3	1	105	i	ahorsewithnoneighs								
4	2	117	u	arpwny								
5	3	99	c	asr								
6	4	116	t	blackboxwarrior								
7	5	102	f	brokeyn								
8	6	123	{	commandnotfound								
9	7	84	T	cnc								

Looking back, I could have done this much more efficiently by keeping all my blue values in the ASCII converter and getting the full flag as an output. However, I still found my spreadsheet a good way to keep everything in one place and would have been easier to adjust in case I got the flag wrong.

Overall, this was a surprisingly fun challenge to do despite my disturbing lack of proficiency. I appreciate SIGPwny for providing me with nice images to work through, and especially appreciate the person who wrote the hints, whoever they may be.