Evlz CTF

Sanity Check II

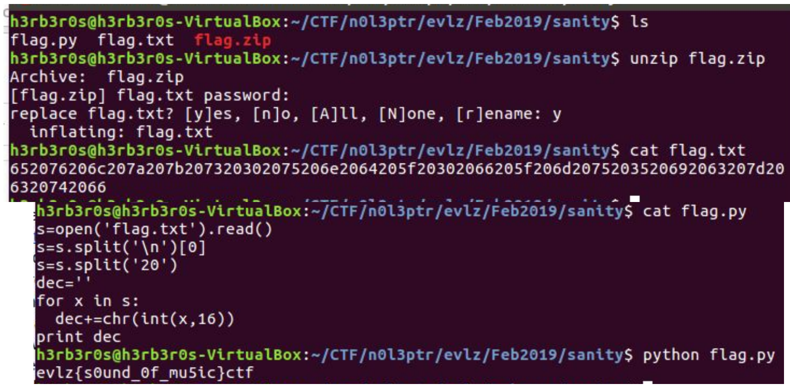
Problem: No one deserving should go ever go empty handed. Image might take some time to load

Analysis: The download for this problem is a zip file.

Solution: Begin by unzipping the zip file. Using the *unzip* command reveals that the file is password protected. There isn't much information on what the password could possibly be, so we are left with either using *fcrackzip* to brute force the password or we could use the free online tool at https://passwordrecovery.io/zip-file-password-removal/ to crack the zip archive. This online tool is extremely fast and reveals the password as **!!!0mc3t.**

Unzipping the archive gets us *flag.txt* which is a string of ascii characters. Close analysis of the string reveals a repetition of '20' every 2 characters. This is common in URL encoding as 0x20 is the hex byte for a space. Meaning that the string is a hex string. From there I wrote a python script to convert the string the from hex to ascii and found the flag.

Flag: evlz{s0und_0f_mu5ic}ctf



```
h3rb3r0s@h3rb3r0s-VirtualBox:~/CTF/n0l3ptr/evlz/Feb2019/sanity$ ls
flag.py  flag.txt  flag.zip
h3rb3r0s@h3rb3r0s-VirtualBox:~/CTF/n0l3ptr/evlz/Feb2019/sanity$ unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.txt password:
replace flag.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: flag.txt
h3rb3r0s@h3rb3r0s-VirtualBox:~/CTF/n0l3ptr/evlz/Feb2019/sanity$ cat flag.txt
652076206c207a207b20732030207520 6e2064205f20302066205f206d20752035206920632 07d20
```

```
652076206c207a207b207320302075206e2064205f20302066205f206d207520352069206320 7d20
6320742066
```

```
h3rb3r0s@h3rb3r0s-VirtualBox:~/CTF/n0l3ptr/evlz/Feb2019/sanity$ cat flag.py
s=open('flag.txt').read()
s=s.split('\n')[0]
s=s.split('20')
dec=''
for x in s:
  dec+=chr(int(x,16))
print dec
h3rb3r0s@h3rb3r0s-VirtualBox:~/CTF/n0l3ptr/evlz/Feb2019/sanity$ python flag.py
evlz{s0und_0f_mu5ic}ctf
```

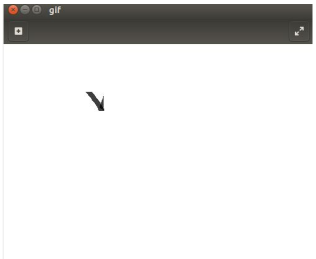
Don't Blink

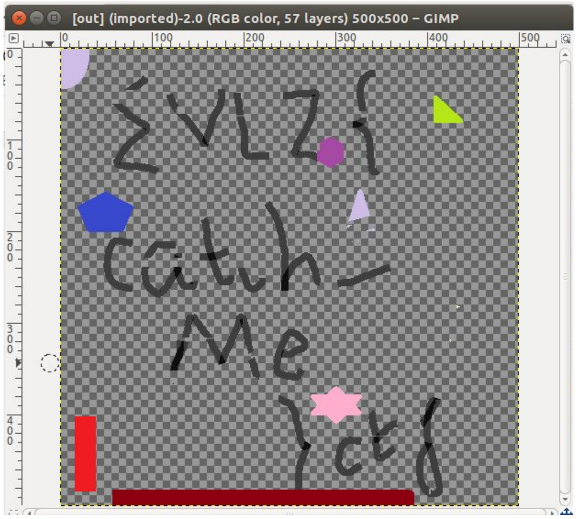
Problem: Do you have persistence of vision? Well try it out with this file

Analysis: The download for this problem is a gif file. There are a bunch of small markings moving through the gif on a white background, some of them overlapping each other on different frames. If we were to compress the frames we may find something.

Solution: We could open the gif up in *gimp* and manually erase all of the white background to compress the layers, or we could use the *convert* command to do it for us. The *convert* command is as easy as *convert gif -transparent white out.gif*. From there open up the new file with *gimp out.gif* and you will be able to see the message.

Flag: evlz{catch_Me}ctf




[out] (imported)-2.0 (RGB color, 57 layers) 500x500 – GIMP
0
100
200
300
400
500
evlz{
catch_
Me
}ctf