PicoCTF is a fun set of challenges that are geared toward beginners and high school students. Annually, a new challenge is issued. This year a 2-week challenge will be held in March. In order to prepare for the challenges teams or individuals can work through the PicoGym challenges. Below are the first 5 challenges that appear at: https://play.picoctf.org/practice
My comments are highlighted in yellow. In all challenges, if a download is required, I am created a folder with the name of the challenge for the downloads.

OBEDIENT CAT

This challenge has you download and read a file. This is a pretty straight forward challenge. PicoCTF provides a Webshell that you can use if you would like. The challenges have you download information for solving. For this reason, I am using the Ubuntu App on a Windows machine.

It is important to note the format of the Flags that we are looking for. Once found, the flags have to be entered in the correct format to be accepted by PicoCTF. The format is picoCTF{XXXXXXXXXXXXXXXXXXXX}

```
XXXXXX@Cyber-PC:~/picogym/obedient_cat$ sudo wget
https://mercury.picoctf.net/static/0e428b2db9788d31189329bed089ce98/flag
--2022-03-14 13:46:31--
https://mercury.picoctf.net/static/0e428b2db9788d31189329bed089ce98/flag
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 34 [application/octet-stream]
Saving to: ‘flag’

flag                          100%[=============================================>]
34  --.-KB/s    in 0s

2022-03-14 13:46:31 (994 KB/s) - ‘flag’ saved [34/34]

XXXXXX@Cyber-PC:~/picogym/obedient_cat$ ls
flag
XXXXXX@Cyber-PC:~/picogym/obedient_cat$ cat flag
picoCTF{s4n1ty_v3r1f13d_2fd6ed29}
XXXXXX@Cyber-PC:~/picogym/obedient_cat$
```

MOD26

The challenge asks if you know what ROT13 is. And the string they give you (below) is clearly encoded.

cvpbPGS{arkg_gvzr_V'yy_gel_2_ebhaqf_bs_ebg13_hyLicInt}

Below is a ROT13 decoder, so we pass it in the encoded string we were given to find the flag.

https://cryptii.com/pipes/rot13-decoder

picoCTF{next_time_I'll_try_2_rounds_of_rot13_ulYvpVag}

PYTHON WRANGLING

In this challenge we know there are three files to download and that python is a scripting language. Endy.py is a python program, pw.txt is a password that needs to be used somewhere, and flag.txt.en is an encoded string that we need to decode.

```
XXXXXX@Cyber-PC:~/picogym/python_wrangling$ sudo wget
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/ende.py
--2022-03-14 13:54:22--
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/ende.py
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1328 (1.3K) [application/octet-stream]
Saving to: 'ende.py'
ende.py                      100%[=================================================>]
1.30K  --.-KB/s    in 0s

2022-03-14 13:54:22 (22.4 MB/s) - 'ende.py' saved [1328/1328]

XXXXXX@Cyber-PC:~/picogym/python_wrangling$ ls
ende.py
XXXXXX@Cyber-PC:~/picogym/python_wrangling$ python3 ende.py
Usage: ende.py (-e/-d) [file]
XXXXXX@Cyber-PC:~/picogym/python_wrangling$ sudo wget
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/pw.txt
--2022-03-14 13:55:26--
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/pw.txt
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33 [application/octet-stream]
Saving to: 'pw.txt'
pw.txt                       100%[=================================================>]
33  --.-KB/s    in 0s

2022-03-14 13:55:26 (663 KB/s) - 'pw.txt' saved [33/33]

XXXXXX@Cyber-PC:~/picogym/python_wrangling$ cat pw.txt
ac9bd0ffac9bd0ffac9bd0ffac9bd0ff
```

This is our password, we just need to figure out how to use it.

```
XXXXXX@Cyber-PC:~/picogym/python_wrangling$ cat ende.py
```

When we look at the code for ende.py we see that the program is used to -e (encode) or -d (decode) a [file]. We know we have a flag.txt.en that needs to be decoded and we have a file that can do the decoding. We can also see in the code that the program will ask us for a password. So, we now have all the pieces, we just need to put them all together.

```
import sys
import base64
from cryptography.fernet import Fernet

usage_msg = "Usage: "+ sys.argv[0] +" (-e/-d) [file]"
help_msg = usage_msg + "\n" +\
        "Examples:\n" +\
        "  To decrypt a file named 'pole.txt', do: " +\
        "'$ python "+ sys.argv[0] +" -d pole.txt'\n"

if len(sys.argv) < 2 or len(sys.argv) > 4:
    print(usage_msg)
```

```
    sys.exit(1)

if sys.argv[1] == "-e":
    if len(sys.argv) < 4:
        sim_sala_bim = input("Please enter the password:")
    else:
        sim_sala_bim = sys.argv[3]

    ssb_b64 = base64.b64encode(sim_sala_bim.encode())
    c = Fernet(ssb_b64)

    with open(sys.argv[2], "rb") as f:
        data = f.read()
        data_c = c.encrypt(data)
        sys.stdout.write(data_c.decode())

elif sys.argv[1] == "-d":
    if len(sys.argv) < 4:
        sim_sala_bim = input("Please enter the password:")
    else:
        sim_sala_bim = sys.argv[3]

    ssb_b64 = base64.b64encode(sim_sala_bim.encode())
    c = Fernet(ssb_b64)

    with open(sys.argv[2], "r") as f:
        data = f.read()
        data_c = c.decrypt(data.encode())
        sys.stdout.buffer.write(data_c)

elif sys.argv[1] == "-h" or sys.argv[1] == "--help":
    print(help_msg)
    sys.exit(1)

else:
    print("Unrecognized first argument: "+ sys.argv[1])
    print("Please use '-e', '-d', or '-h'.")

XXXXXX@Cyber-PC:~/picogym/python_wrangling$ sudo wget
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/flag.txt.en
--2022-03-14 13:57:13--
https://mercury.picoctf.net/static/325a52d249be0bd3811421eacd2c877a/flag.txt.en
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 140 [application/octet-stream]
Saving to: ‘flag.txt.en’

flag.txt.en                   100%[=================================================>]
140  --.-KB/s    in 0s

2022-03-14 13:57:13 (2.81 MB/s) - ‘flag.txt.en’ saved [140/140]

Here is the command to use the program.  We tell it to decode the file using the ende.py
program.  Then we pass it the password we downloaded to get the flag.
XXXXXX@Cyber-PC:~/picogym/python_wrangling$ python3 ende.py -d flag.txt.en
Please enter the password:ac9bd0ffac9bd0ffac9bd0ffac9bd0ff
picoCTF{4p0110_1n_7h3_h0us3_ac9bd0ff}
```

WAVE A FLAG

In this challenge we have to download a file and figure out what to do with it.

```
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ sudo wget
https://mercury.picoctf.net/static/f95b1ee9f29d631d99073e34703a2826/warm
--2022-03-14 14:01:22--
https://mercury.picoctf.net/static/f95b1ee9f29d631d99073e34703a2826/warm
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10936 (11K) [application/octet-stream]
Saving to: 'warm'

warm                          100%[=============================================>]
10.68K  --.-KB/s    in 0s

2022-03-14 14:01:22 (30.5 MB/s) - 'warm' saved [10936/10936]

XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ ls
warm
```

The first thing we notice is that we don't have permissions to do anything with the file we just downloaded. So we give ourselves execute permission.

```
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ ls -alps
total 1036
   0 drwxr-xr-x 1 root root   512 Mar 14 14:01 ./
   0 drwxr-xr-x 1 root root   512 Mar 14 14:00 ../
1036 -rw-r--r-- 1 root root 10936 Mar 15  2021 warm
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ sudo chmod u=rwx warm
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ ls -alps
total 1036
   0 drwxr-xr-x 1 root root   512 Mar 14 14:01 ./
   0 drwxr-xr-x 1 root root   512 Mar 14 14:00 ../
1036 -rwxr--r-- 1 root root 10936 Mar 15  2021 warm
```

Now we run the file as a program.

```
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ sudo ./warm
Hello user! Pass me a -h to learn what I can do!
```

It wants us to pass it a switch (-h)

```
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$ sudo ./warm -h
Oh, help? I actually don't do much, but I do have this flag here:
picoCTF{b1scu1ts_4nd_gr4vy_f0668f62}
XXXXXX@Cyber-PC:~/picogym/wave_a_flag$
```

INFORMATION

We are given a .jpg file and told to find a flag. This flag could be put into a picture using stenography (spoiler alert: it's not) or it could be in the meta data (it is).

```
XXXXXX@Cyber-PC:~/picogym/information$ sudo wget
https://mercury.picoctf.net/static/e5825f58ef798fdd1af3f6013592a971/cat.jpg
--2022-03-14 14:06:39--
https://mercury.picoctf.net/static/e5825f58ef798fdd1af3f6013592a971/cat.jpg
Resolving mercury.picoctf.net (mercury.picoctf.net)... 18.189.209.142
Connecting to mercury.picoctf.net (mercury.picoctf.net)|18.189.209.142|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 878136 (858K) [application/octet-stream]
Saving to: 'cat.jpg'

cat.jpg                       100%[=============================================>]
857.55K  1.83MB/s    in 0.5s
```

```
2022-03-14 14:06:40 (1.83 MB/s) - ‘cat.jpg’ saved [878136/878136]
```

Exiftool is a great program that will help us read the metadata of cat.jpg

```
XXXXXX@Cyber-PC:~/picogym/information$
XXXXXX@Cyber-PC:~/picogym/information$ exiftool cat.jpg
ExifTool Version Number         : 11.88
File Name                       : cat.jpg
Directory                       : .
File Size                       : 858 kB
File Modification Date/Time     : 2021:03:15 12:24:46-06:00
File Access Date/Time           : 2022:03:14 14:27:31-06:00
File Inode Change Date/Time     : 2022:03:14 14:26:33-06:00
File Permissions                : rwxr--r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.02
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
Current IPTC Digest             : 7a78f3d9cfb1ce42ab5a3aa30573d617
Copyright Notice                : PicoCTF
Application Record Version      : 4
XMP Toolkit                     : Image::ExifTool 10.80
License                         : cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9
Rights                          : PicoCTF
Image Width                     : 2560
Image Height                    : 1598
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 2560x1598
Megapixels                      : 4.1
XXXXXX@Cyber-PC:~/picogym/information$
```

There are two strings that look like they could be flags:

```
7a78f3d9cfb1ce42ab5a3aa30573d617
cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9
```

To figure out which one is the flag we can put them into CyberChef and see if they are encoded. https://gchq.github.io/CyberChef/

Use CyberChef to convert cGljb0NURnt0aGVfbTN0YWRhdGFfMXNfbW9kaWZpZWR9 from Base64
This gives us: picoCTF{the_m3tadata_1s_modified}