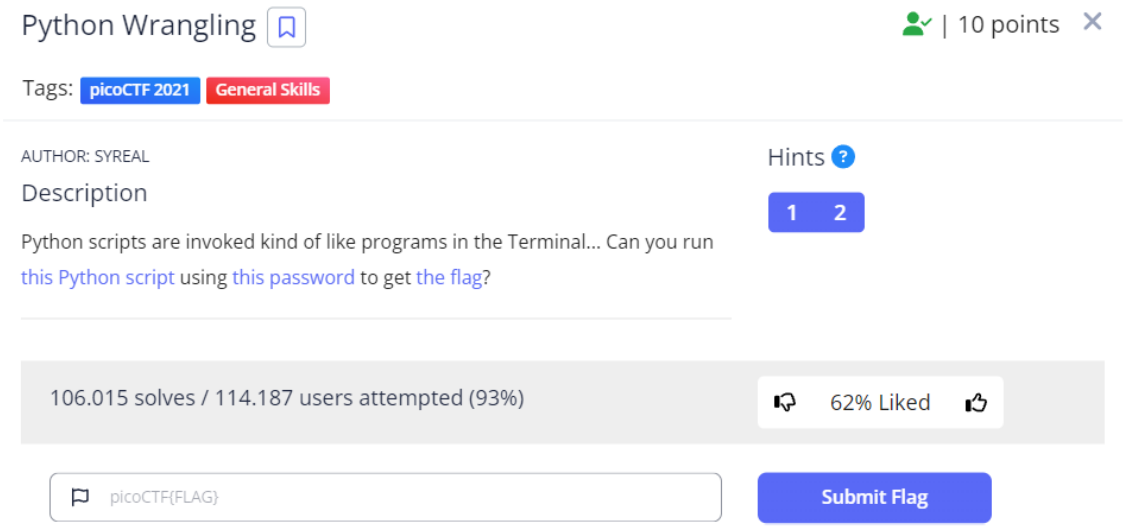

En este CTF, nos van a dar un script en Python, un fichero con una contraseña y un fichero con la flag.

```
┌──(kali㉿kali)-[~/picoCTF/Python_Wrangling]
└─$ ls
ende.py  flag.txt.en  pw.txt
```

A continuación vamos a ver el contenido de los ficheros flag.txt.en y pw.txt:

```
┌──(kali㉿kali)-[~/picoCTF/Python_Wrangling]
└─$ cat flag.txt.en
gAAAAABgUAIWsYfVayn4m1dKle5X91HrZW_MIRAW4ILPgf4gD6jalLF4PysYB5_YTpDwclcQPqw_0xTxanpJ_Urx5Vi6mTeBA_rWPA_WQLvVXXHp1mG3EpOgY8Na1_NIAfc9LceH_L2o
```

```
┌──(kali㉿kali)-[~/picoCTF/Python_Wrangling]
└─$ cat pw.txt
67c6cc9667c6cc9667c6cc9667c6cc96
```

Ahora vamos a examinar el código del script en Python, para visualizarlo, vamos a usar el comando “cat + nameFile”.

En esta captura, vemos que nos están explicando cómo debemos ejecutar el script desde el terminal y como meterle los argumentos, sabiendo que el argumento 0 es el nombre del fichero.py

```
┌──(kali㉿kali)-[~/picoCTF/Python_Wrangling]
└─$ cat ende.py

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"
```

En esta captura, se comprueba que le hemos pasado 2,3 o 4 argumentos, si no, se para la ejecución con un exit 1 (error).

```
if len(sys.argv) < 2 or len(sys.argv) > 4:
    print(usage_msg)
    sys.exit(1)
```

Si el segundo argumento que le pasamos es “-e”, nos metemos en el if. Primero comrueba que los argumentos pasados son menos de 4, en ese caso, tenemos que introducir la contraseña, en caso contrario, en la variable, se nos guarda el cuarto argumento que le pasamos.

A continuación, codifica en base64 la entrada y luego lo cifra con FERNET, creando así la clave secreta. Después de esto, se abre el fichero y se guarda en data (el fichero es el tercer argumento que le pasamos) y lo cifra usando la clave secreta ya creada.

```
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())
```

Si le hemos pasado en el segundo argumento “-d”, hacemos lo mismo, donde se nos pide introducir la contraseña, y en vez de cifrar el ultimo fichero que le pasamos como argumento, lo descifra.

```
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)
```

Para la resolución, vamos a ejecutar el script y como argumentos, le vamos a pasar -d (porque queremos descifrar) y luego el fichero flag.txt.en. Una vez que nos pida la contraseña, vamos a pasarle la contraseña del fichero pw.txt.

```
┌──(kali㉿kali)-[~/picoCTF/Python_Wrangling]
└─$ python ende.py  -d flag.txt.en
Please enter the password:67c6cc9667c6cc9667c6cc9667c6cc96
picoCTF{4p0110_1n_7h3_h0us3_67c6cc96}
```