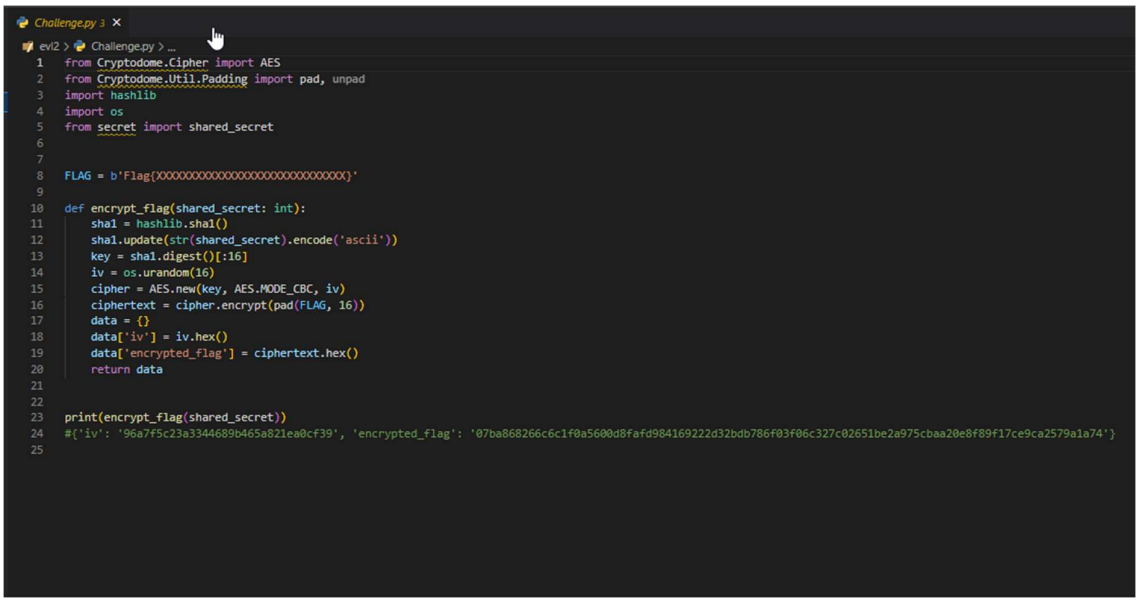
Ramadan CTF WriteUp

By: cicakberlari

Eventually (Cryptography)

- In this challenge, we got 2 files from extracted **evl.7z** file. First file is python file that contains code to encrypt the flag and the second file is txt file.

1. Challenge.py file



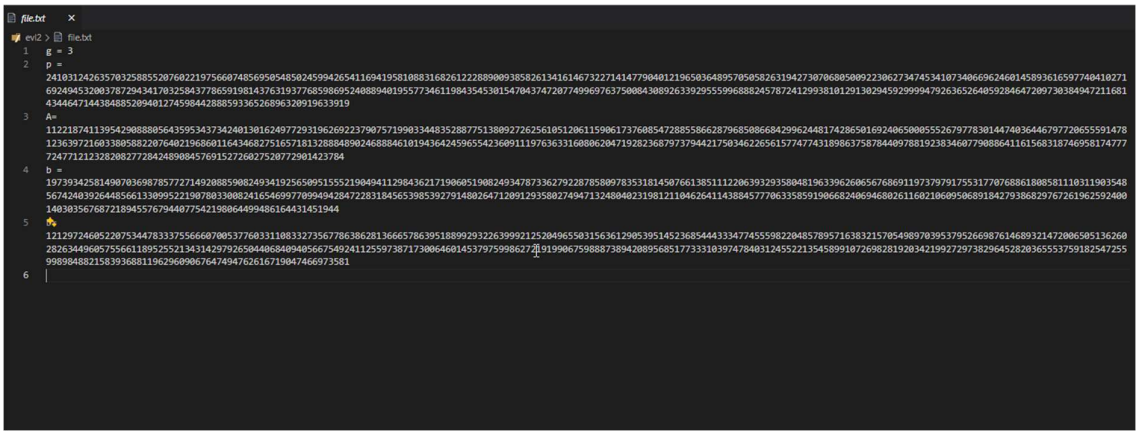
```python
from Cryptodome.Cipher import AES
from Cryptodome.Util.Padding import pad, unpad
import hashlib
import os
from secret import shared_secret


FLAG = b'Flag{XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX}'

def encrypt_flag(shared_secret: int):
    sha1 = hashlib.sha1()
    sha1.update(str(shared_secret).encode('ascii'))
    key = sha1.digest()[:16]
    iv = os.urandom(16)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    ciphertext = cipher.encrypt(pad(FLAG, 16))
    data = {}
    data['iv'] = iv.hex()
    data['encrypted_flag'] = ciphertext.hex()
    return data


print(encrypt_flag(shared_secret))
#{'iv': '96a7f5c23a3344689b465a821ea0cf39', 'encrypted_flag': '07ba868266c6c1f0a5600d8fafd984169222d32bdb786f03f06c327c02651be2a975cbaa20e8f89f17ce9ca2579a1a74'}
```

2. File.txt



```
g = 3
p =
241031242635703258855207602219756607485695054850245994265411694195810883168261222889009385826134161467322714147790401219650364895705058263194273070680500922306273474534107340669624601458936165977404102716924945320037872943417032584377865919814376319377685986952408894019557734611984354530154704374720774996976375008430892633929555996888245787241299381012913029459299994792636526405928464720973038494721168143446471443848852094012745984428885933652689632091633919
A=
112218741139542908880564359534373424013016249772931962692237907571990334483528877513809272625610512061159061737608547288558662879685086684299624481742865016924065000555267977830147403644679772065559147812363972160338058822076402196860116434682751657181328884890246888461019436424596554236091119763633160806204719282368797379442175034622656157747431898637587844097881923834607790886411615683318746958174777724771212328208277284248908457691527260275207729014237 84
b =
197393425814907036987857727149208859082493419256509515552190494112984362171906051908249347873362792287858097835318145076613851112206393293580481963396260656768691197379791755317707688618085811103119035485674240392644856613309952219078033082416546997709949428472283184565398539279148026471209129358027494713248040231981211046264114388457770633585919066824869468026116821060950689184279386829767261962592400140303567687218945576794407754219806449948616443145194 4
A=
121297246052207534478333755666070053776033110833273567786386281366657863951889929322639992125204965503156361290539514523685444333477455598220485789571638321570549897039537952669876146893214720065051362602826344960575566118952552134314297926504406840940566754924112559738717300646014537975998627219199067598887389420895685177331039747840312455221354589910726982819203421992729738296452820365553759182547255998984882158393688119629609067647494762616719047466973581
```

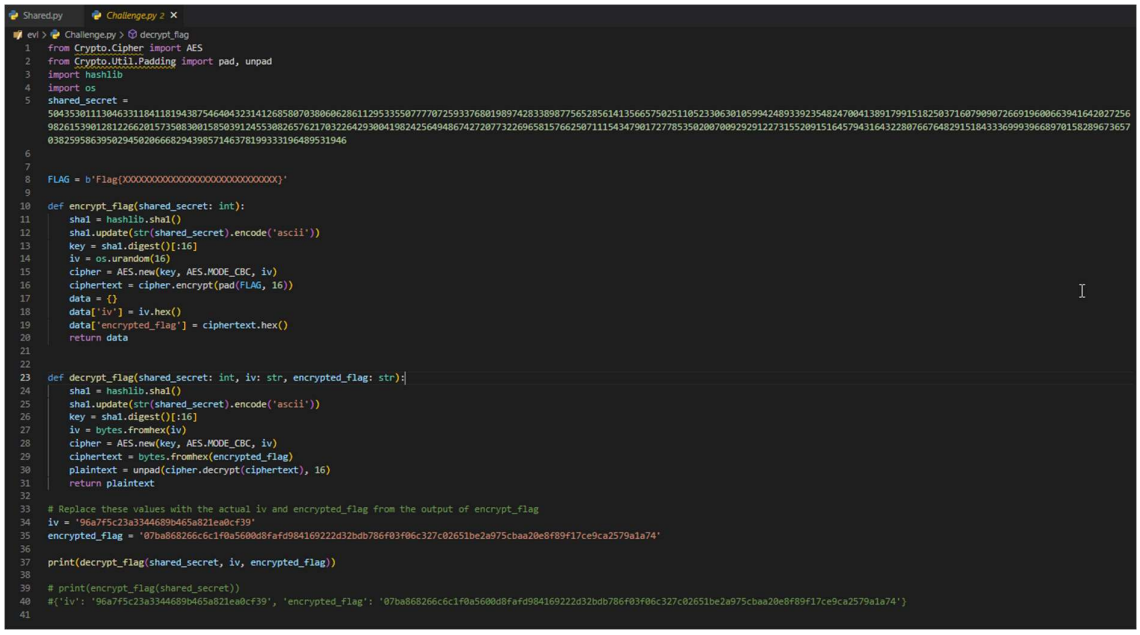
3. By doing some research on google, we know that file.txt are some variables that related to Diffie-Hellman key exchange protocol.

4. We code a simple python code to get the shared secret(to use inside *Challenge.py)*.

5. Run the *Shared.py* and we will get the shared secret.

6. Now moving to python file *Challenge.py* we modify some of the code :-
    a. Change the import from Cryptodome to Crypto
    b. Add some import from hashlib and os
    c. Add shared_secret variables that we got from *Shared.py*
    d. Add some function to decrypt the flag

    Here the modified code :



7. Now we just run the file and then we got the decrypted flag.