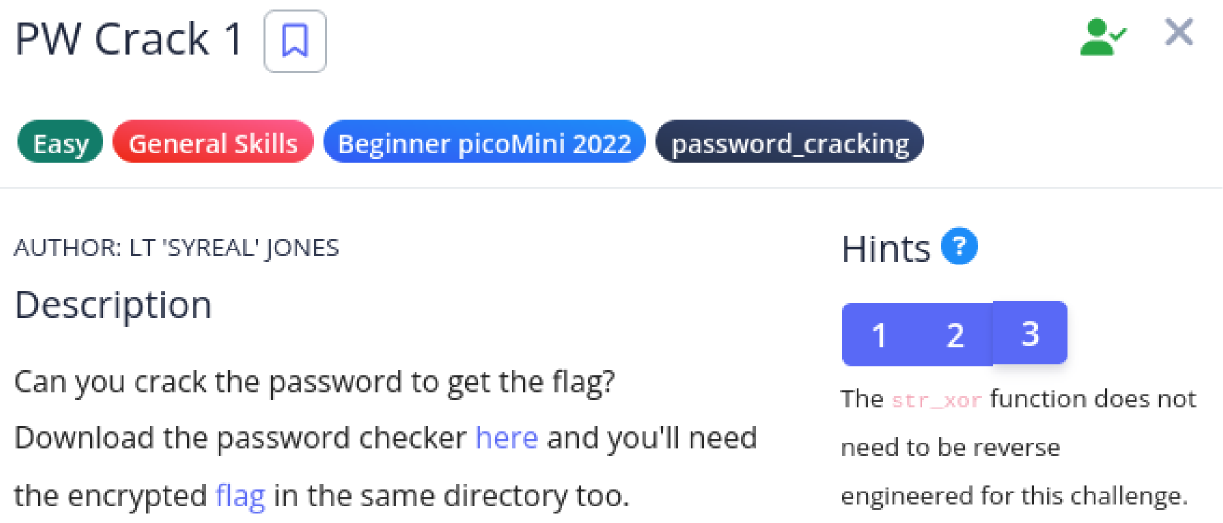
LinkedIn: Kelvin Kimotho



PW Crack 1

Easy General Skills Beginner picoMini 2022 password_cracking

AUTHOR: LT 'SYREAL' JONES

Description

Can you crack the password to get the flag?

Download the password checker here and you'll need the encrypted flag in the same directory too.

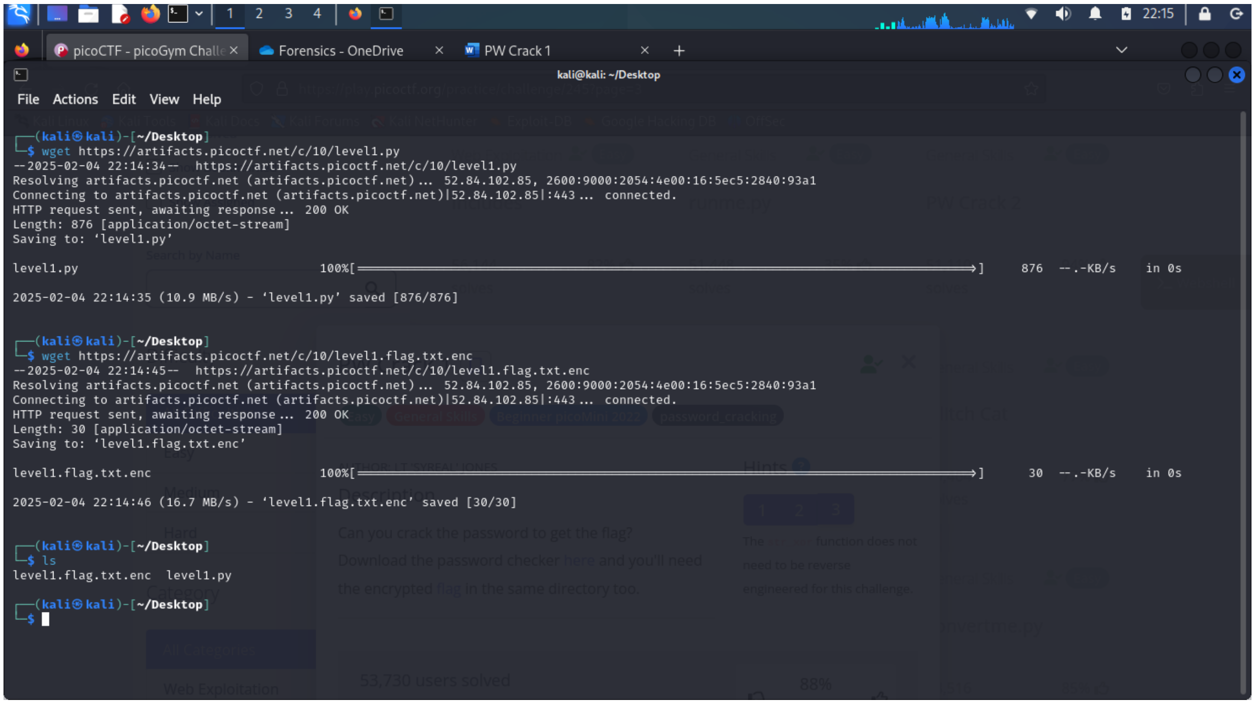
Hints

1 2 3

The `str_xor` function does not need to be reverse engineered for this challenge.

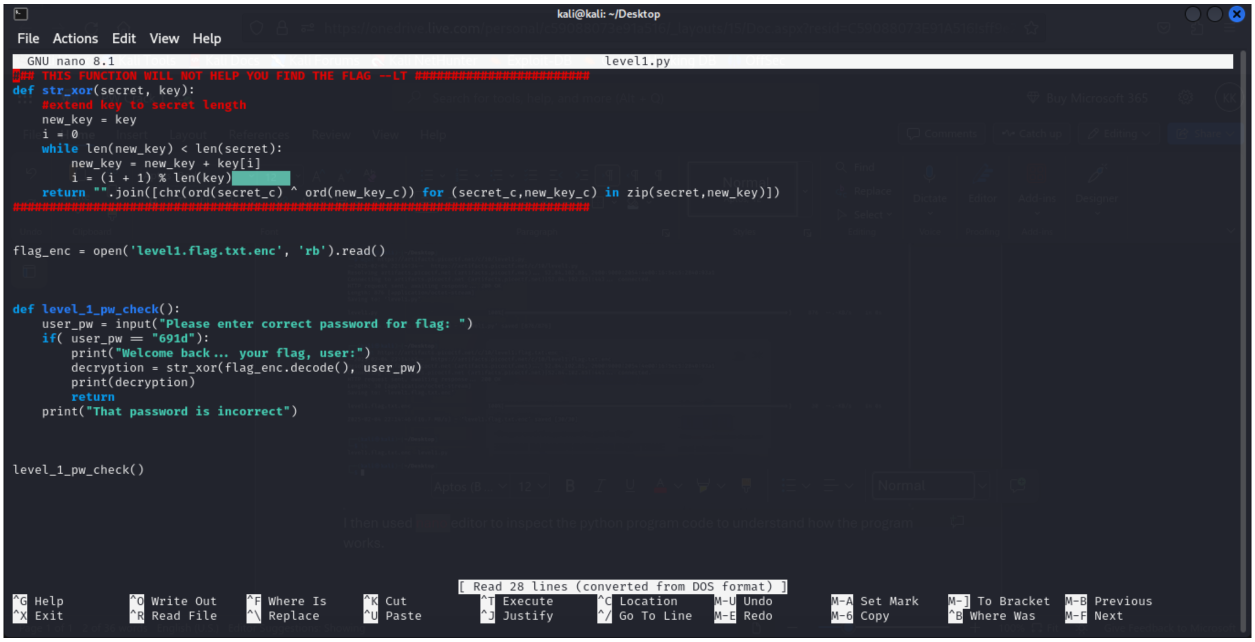
Solution

I downloaded the password checker python file and an encrypted flag file using wget tool.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ wget https://artifacts.picoctf.net/c/10/level1.py
--2025-02-04 22:14:34--  https://artifacts.picoctf.net/c/10/level1.py
Resolving artifacts.picoctf.net (artifacts.picoctf.net)... 52.84.102.85, 2600:9000:2054:4e00:16:5ec5:2840:93a1
Connecting to artifacts.picoctf.net (artifacts.picoctf.net)|52.84.102.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 876 [application/octet-stream]
Saving to: 'level1.py'

level1.py                 100%[===================================================>]     876  --.-KB/s    in 0s

2025-02-04 22:14:35 (10.9 MB/s) - 'level1.py' saved [876/876]


┌──(kali㉿kali)-[~/Desktop]
└─$ wget https://artifacts.picoctf.net/c/10/level1.flag.txt.enc
--2025-02-04 22:14:45--  https://artifacts.picoctf.net/c/10/level1.flag.txt.enc
Resolving artifacts.picoctf.net (artifacts.picoctf.net)... 52.84.102.85, 2600:9000:2054:4e00:16:5ec5:2840:93a1
Connecting to artifacts.picoctf.net (artifacts.picoctf.net)|52.84.102.85|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 30 [application/octet-stream]
Saving to: 'level1.flag.txt.enc'

level1.flag.txt.enc       100%[===================================================>]      30  --.-KB/s    in 0s

2025-02-04 22:14:46 (16.7 MB/s) - 'level1.flag.txt.enc' saved [30/30]


┌──(kali㉿kali)-[~/Desktop]
└─$ ls
level1.flag.txt.enc  level1.py

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```

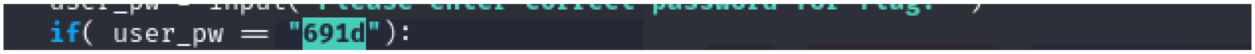
I then used nano editor to inspect the python program code to understand how the program works.



```
### THIS FUNCTION WILL NOT HELP YOU FIND THE FLAG --LT ########################
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])
###############################################################################


flag_enc = open('level1.flag.txt.enc', 'rb').read()



def level_1_pw_check():
    user_pw = input("Please enter correct password for flag: ")
    if( user_pw == "691d"):
        print("Welcome back... your flag, user:")
        decryption = str_xor(flag_enc.decode(), user_pw)
        print(decryption)
        return
    print("That password is incorrect")



level_1_pw_check()
```

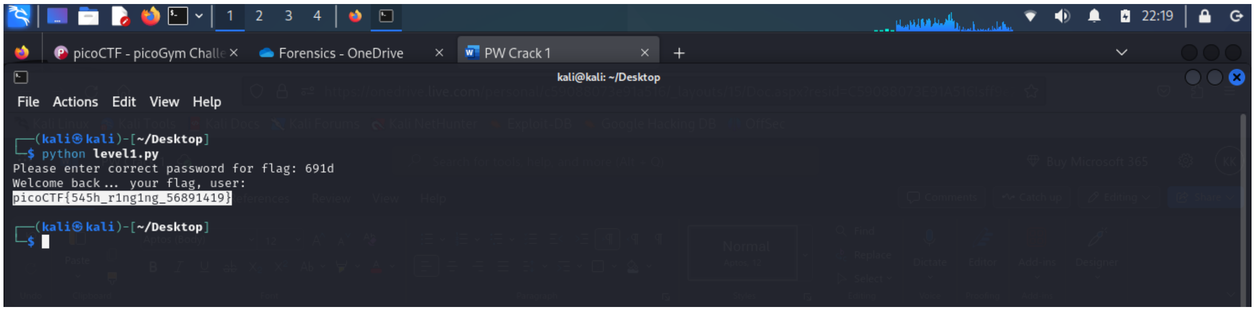
```
[ Read 28 lines (converted from DOS format) ]
```

The program expects the user to enter a password for the program to decrypt the encrypted flag file and print out the flag. User input was compared to 691d meaning that this was the password required.



```
    if( user_pw == "691d"):
```

I went ahead and ran the program passing the password i found and that's how i retrieved the hidden flag as picoCTF{545h_r1ng1ng_56891419}.



```
┌──(kali㉿kali)-[~/Desktop]
└─$ python level1.py
Please enter correct password for flag: 691d
Welcome back... your flag, user:
picoCTF{545h_r1ng1ng_56891419}

┌──(kali㉿kali)-[~/Desktop]
└─$ 
```