Abdelrhman

CTF Player, Digital Forensics, SOC

HOME
CATEGORIES
TAGS
ARCHIVES
ABOUT

Recently Updated
0xl4ugh2024
UofTCTF

Trending Tags
task schedule · backdoor · crypto · cve · discord · disk · firmware · network · pdf · trevorc2

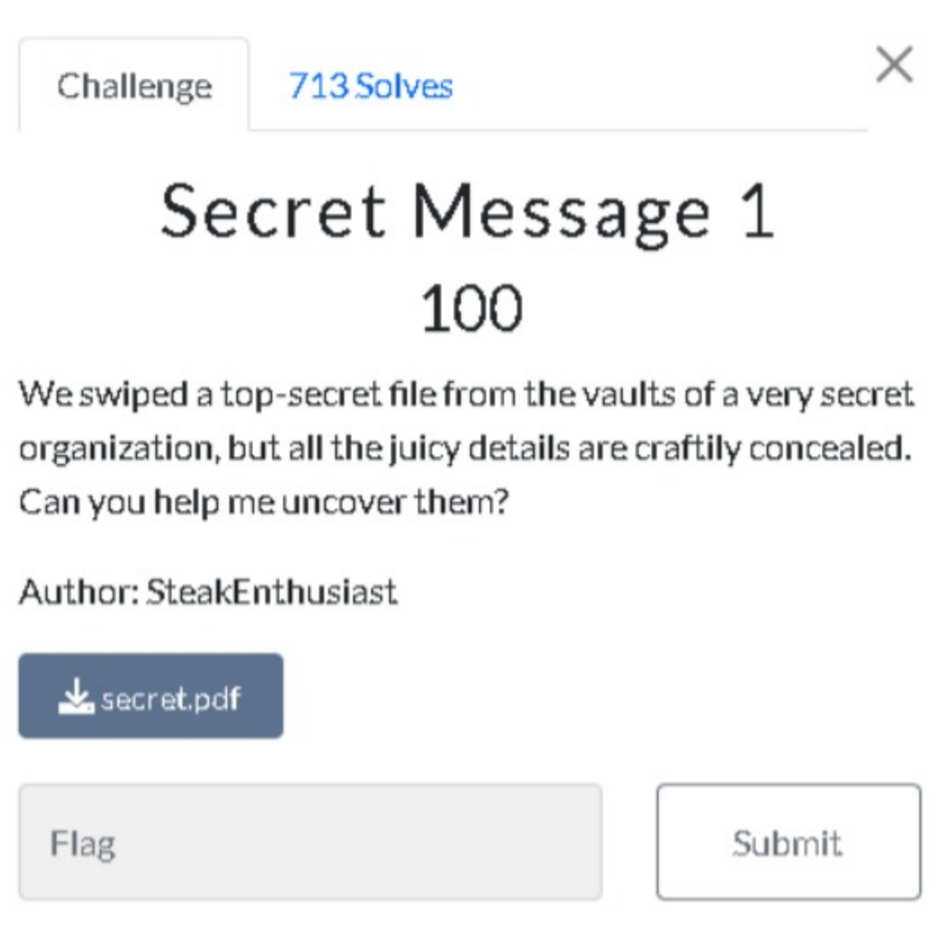
UofTCTF

Posted Jan 15, 2024

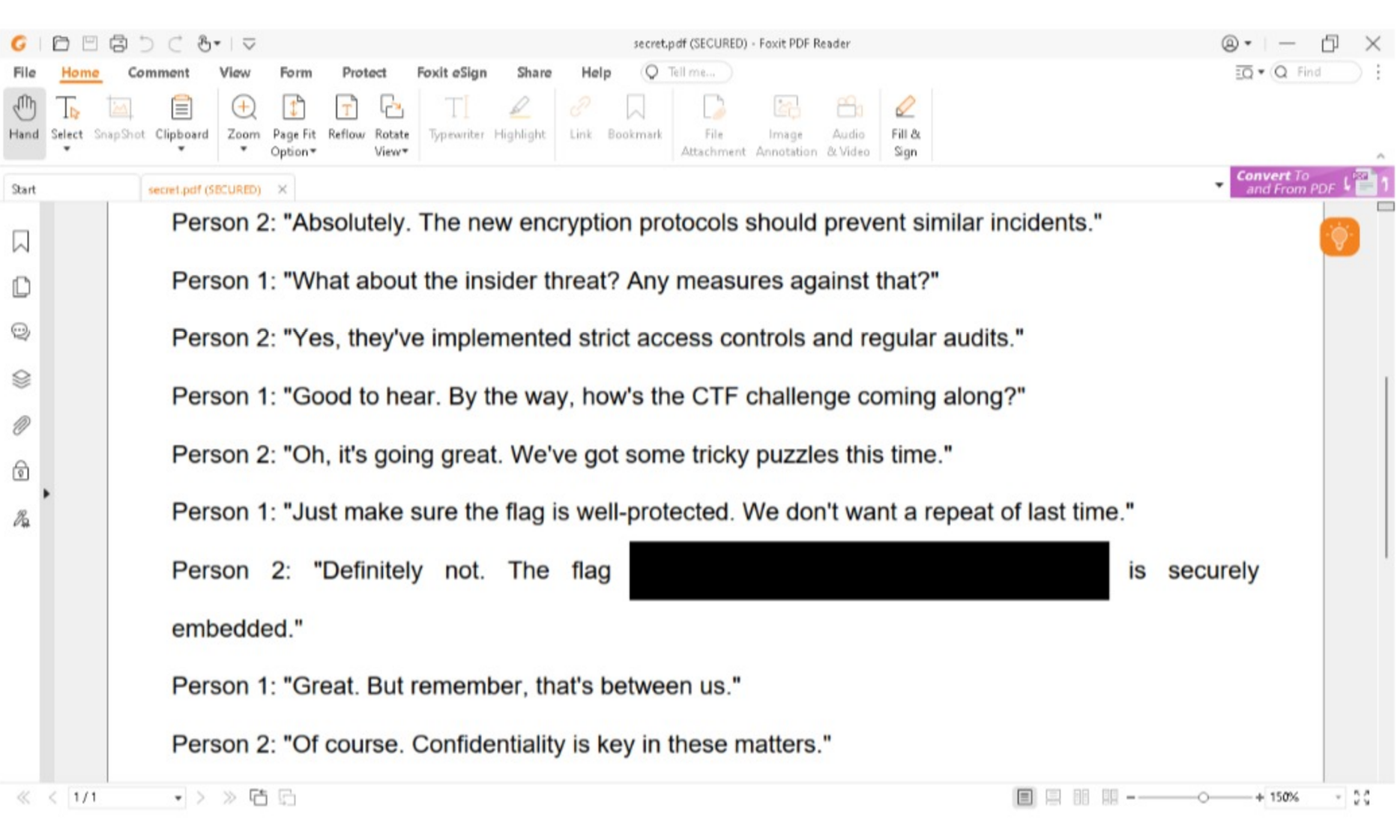
By Abdelrhman Mohamed 4 min read

I participated in the UofTCTF 2024 I solved all forensics challenge,and this is my write-up for most forensics challenge and challenge of IoT

Secret Message 1



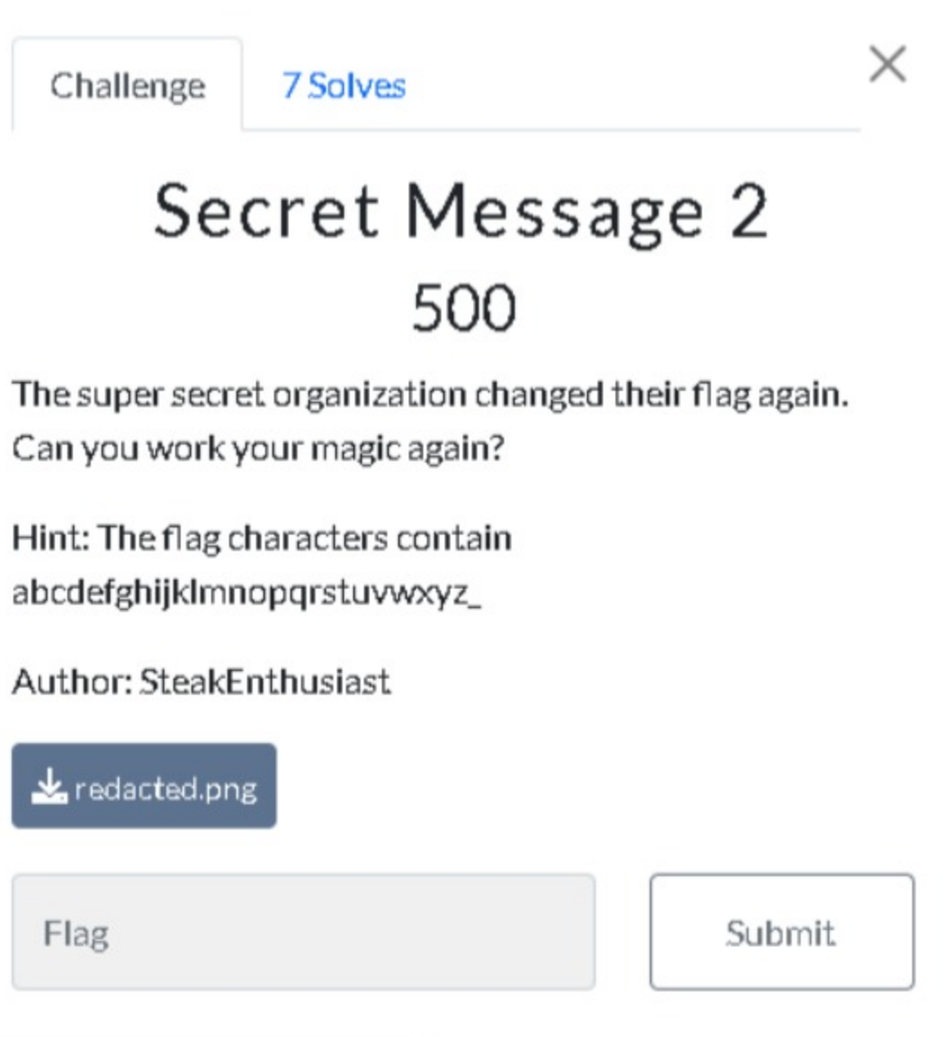
it's pdf file so we open it we saw this



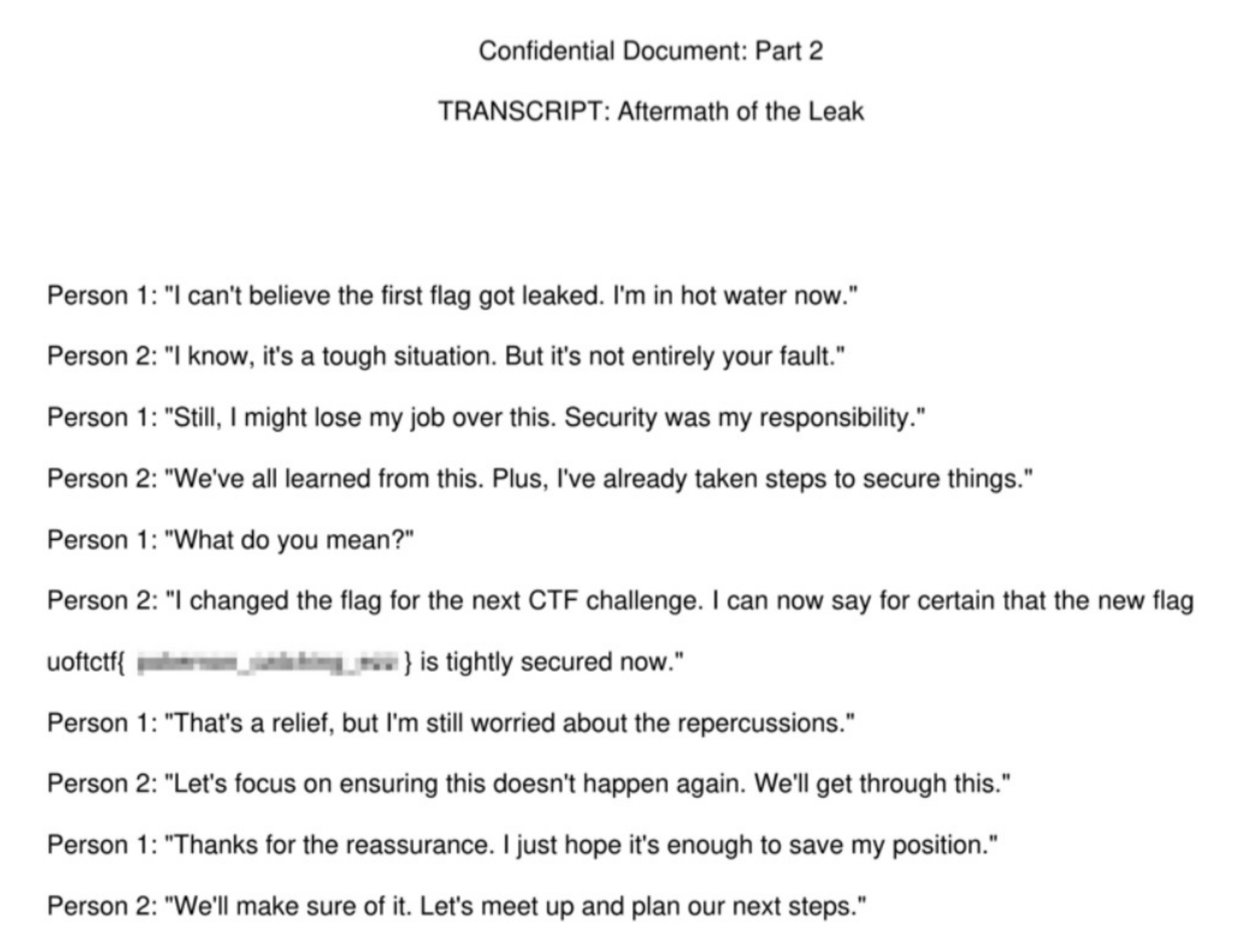

```shell
pdftotext secret.pdf
cat secret.txt |grep uof
```

flag: uoftctf{fired_for_leaking_secrets_in_a_pdf}

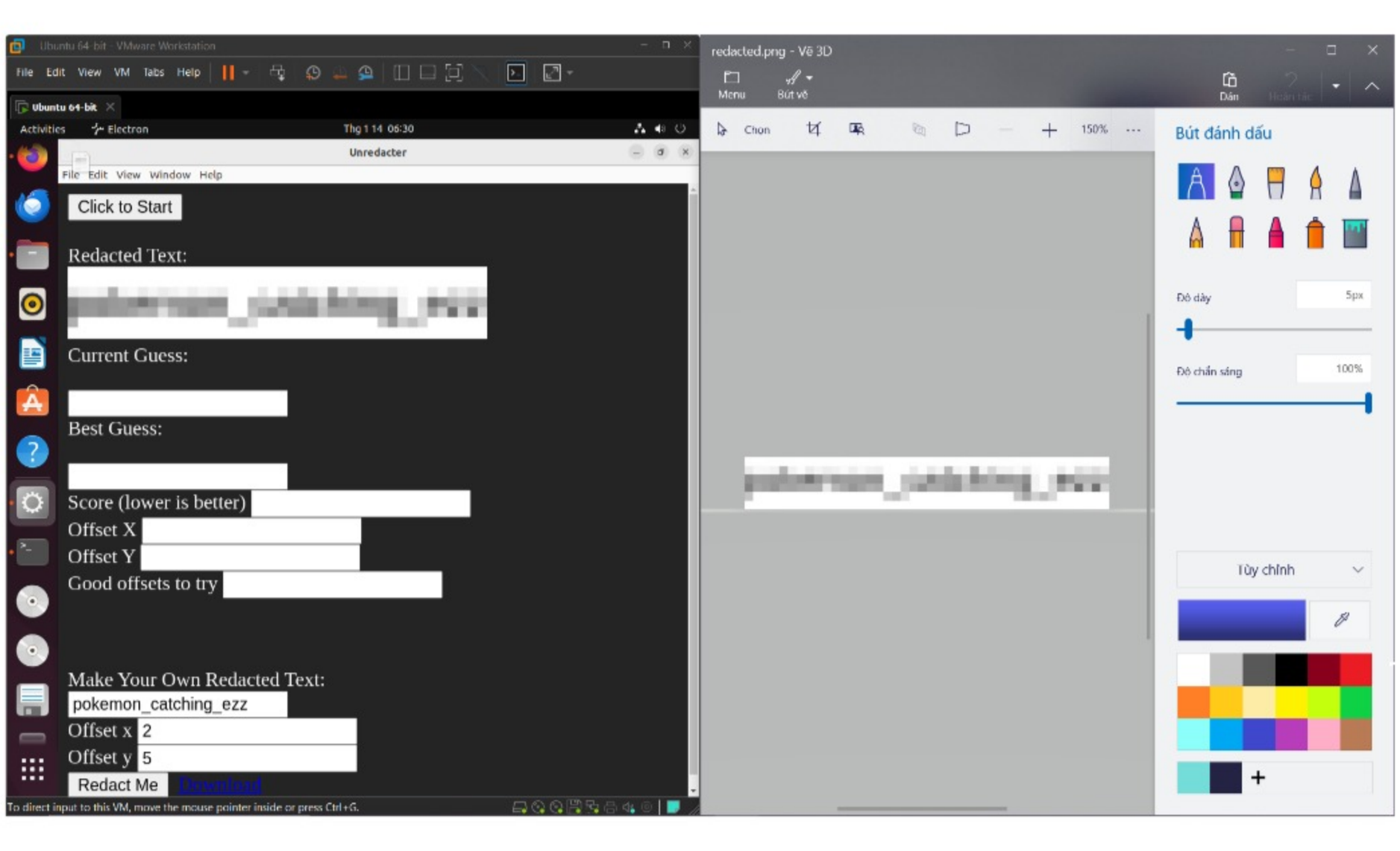
Secret Message 2



it's png so we open it we saw this

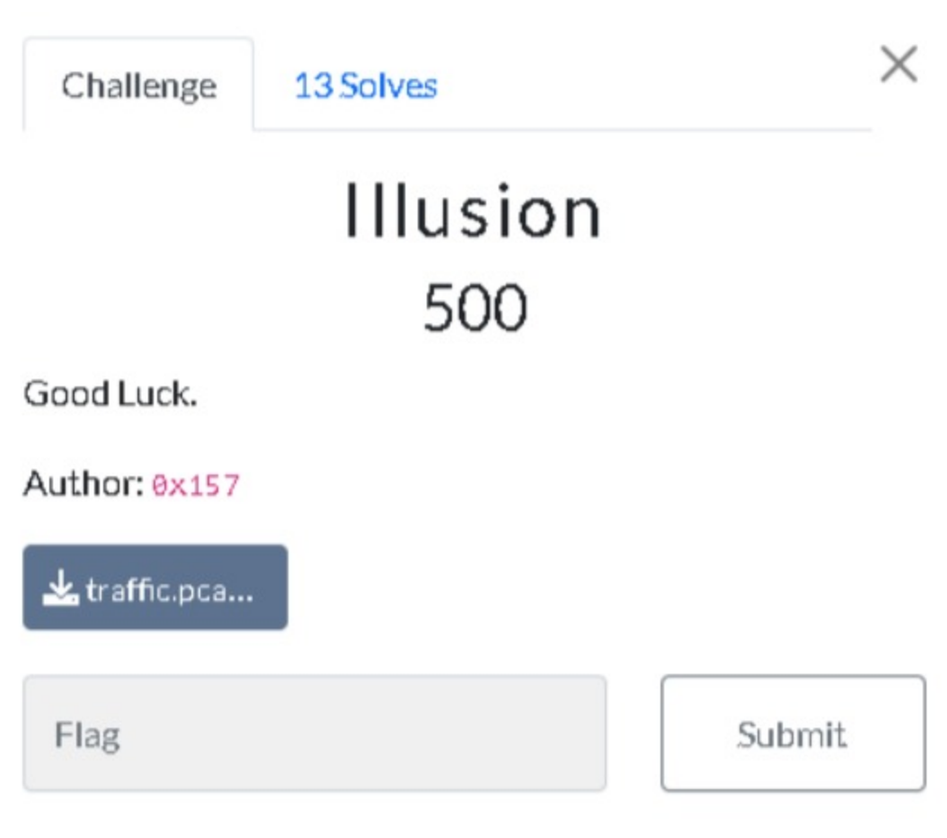


so i searched how to fix the flag so found this article and this tool i replaced secret.png with challenge photo i make him to try automatic to figure out the picture but it gave me wrong flag so my friend did it manually by Redact me and got the flag

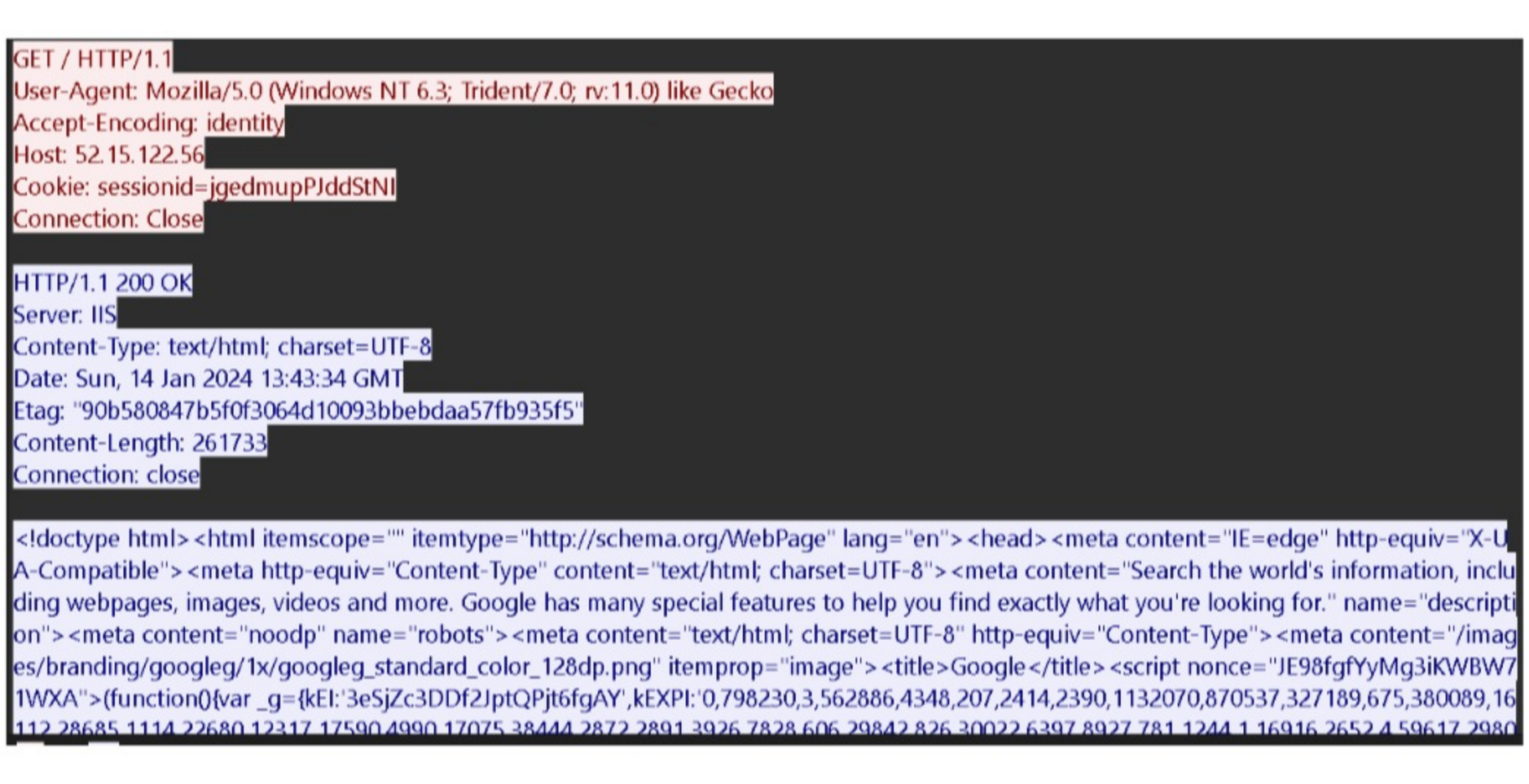


flag: uoftctf{pokemon_catching_ezz}

Illusion



For this challenge we are given a packet capture file, upon inspecting traffic ive noticed a lot of http traffic coming to and from a rogue IP address

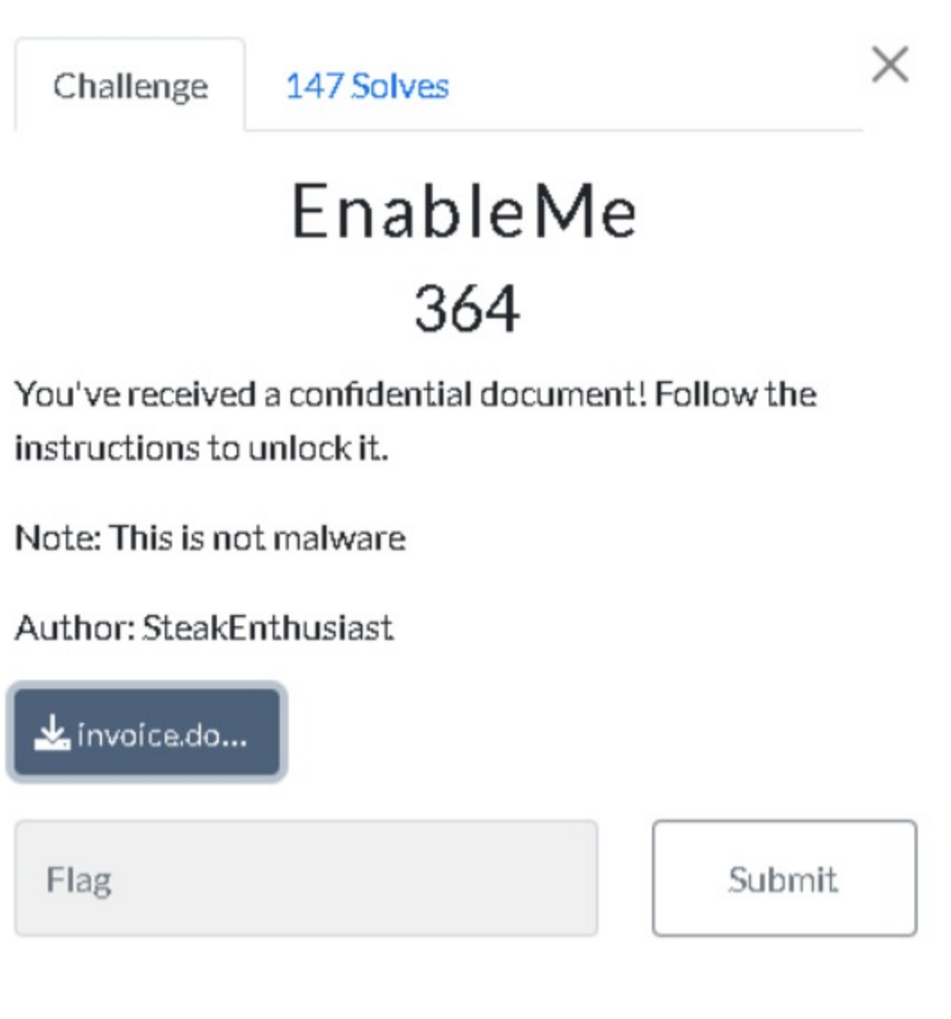


Once i viewed the streams of the packets ive noticed a weird *oldcss=* string and reported that to my teammates, a few hours later my teammate found some interesting github trevorc2 so from trevorc2_server.py I began dissecting its functionality and recognized that it was indeed following the correct trajectory. I comprehended its operational mechanisms and endeavored to decrypt the commands transmitted through the covert HTTP channel.

```python
CIPHER = ("Tr3v0rC2R0x@nd1s@w350m3#TrevorForget")
import random
import base64
import time
import subprocess
import hashlib
from Cryptodome import Random
from Cryptodome.Cipher import AES
import sys
import platform


class AESCipher(object):
    """
    A classical AES Cipher. Can use any size of data and any size of password thanks to padding.
    Also ensure the coherence and the type of the data with a unicode to byte converter.
    """

    def __init__(self, key):
        self.bs = 16
        self.key = hashlib.sha256(AESCipher.str_to_bytes(key)).digest()

    @staticmethod
    def str_to_bytes(data):
        u_type = type(b''.decode('utf8'))
        if isinstance(data, u_type):
            return data.encode('utf8')
        return data

    def _pad(self, s):
        return s + (self.bs - len(s) % self.bs) * AESCipher.str_to_bytes(chr(self.bs - len(s) % self.bs

    @staticmethod
    def _unpad(s):
        return s[:-ord(s[len(s)-1:])]

    def encrypt(self, raw):
        raw = self._pad(AESCipher.str_to_bytes(raw))
        iv = Random.new().read(AES.block_size)
        cipher = AES.new(self.key, AES.MODE_CBC, iv)
        return base64.b64encode(iv + cipher.encrypt(raw)).decode('utf-8')

    def decrypt(self, enc):
        enc = base64.b64decode(enc)
        iv = enc[:AES.block_size]
        cipher = AES.new(self.key, AES.MODE_CBC, iv)
        return self._unpad(cipher.decrypt(enc[AES.block_size:])).decode('utf-8')
Abdelrhman = AESCipher(CIPHER)
encrypted_text = "C9XqWpYeqCIn8Dk8gCVtpdg47vm8e8peFqkfQJ6WVbUvL7ucvQ0ayWnKRBF2GI+1tFBWNMa+wawqeuvFK61RG
decrypted_text = Abdelrhman.decrypt(encrypted_text)
print(decrypted_text)
```

flag: uoftctf{Tr3V0r_C2_1s_H4rd_T0_D3t3c7}

EnableMe



we have docm file so i tried olevba on it

```shell
olevba invoice.docm
```

and i got this vba

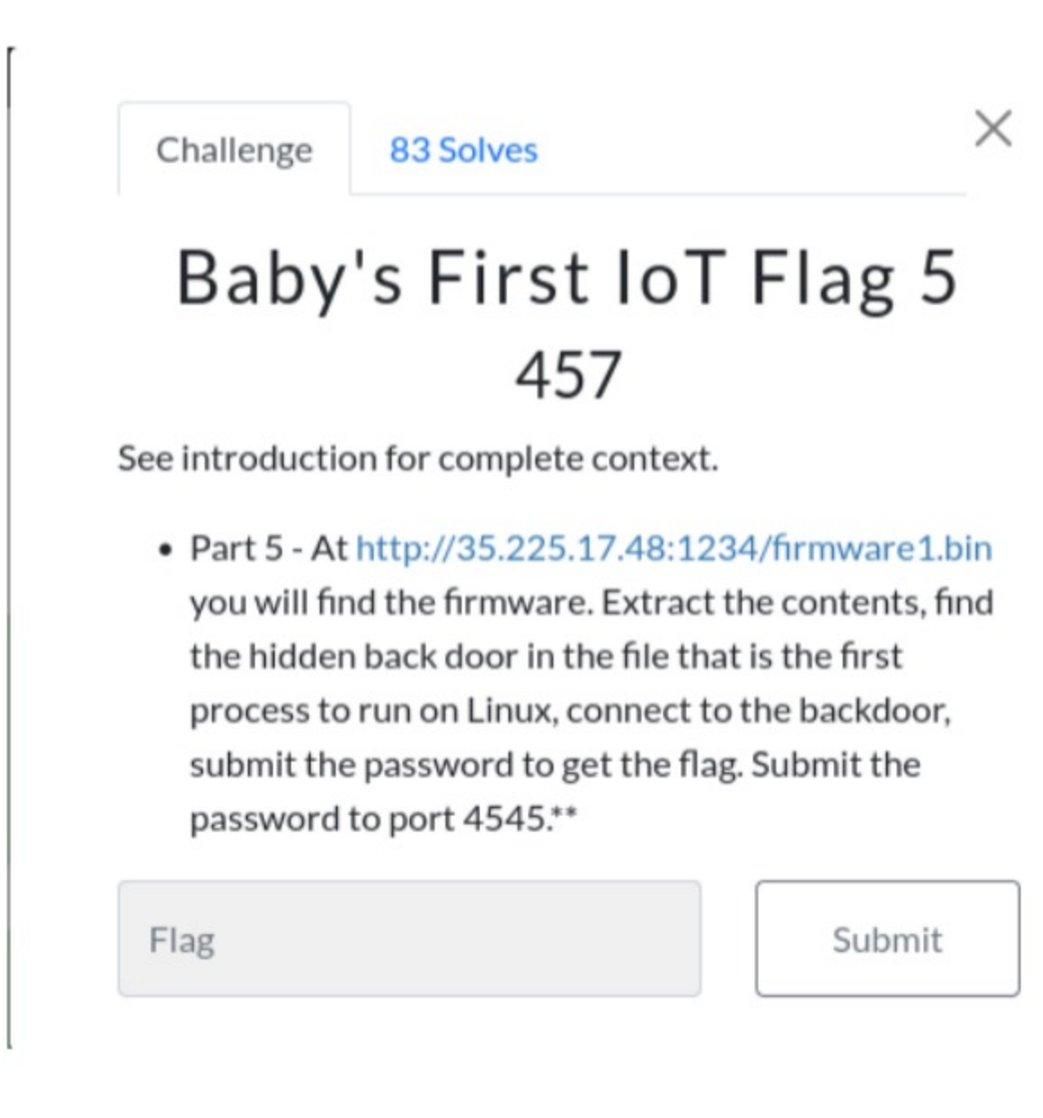
```
Sub AutoOpen()
    Dim v6 As Variant, v7 As Variant
    v6 = Array(98, 120, 113, 99, 116, 99, 113, 108, 115, 39, 116, 111, 72, 113, 38, 123, 36, 34, 72, 116,
    v7 = Array(44, 32, 51, 84, 43, 53, 48, 62, 68, 114, 38, 61, 17, 70, 121, 45, 112, 126, 26, 39, 21, 78

    Dim v8 As Integer: v8 = 23

    Dim v9 As String, v10 As String, v4 As String, i As Integer
    v9 = ""
    For i = 0 To UBound(v6)
        v9 = v9 & Chr(v6(i) Xor Asc(Mid(Chr(v8), (i Mod Len(Chr(v8))) + 1, 1)))
    Next i

    v10 = ""
    For i = 0 To UBound(v7)
        v10 = v10 & Chr(v7(i) Xor Asc(Mid(v9, (i Mod Len(v9)) + 1, 1)))
    Next i

    MsgBox v10
End Sub
```

so i convert it to python code and i run it

```python
v6 = [98, 120, 113, 99, 116, 99, 113, 108, 115, 39, 116, 111, 72, 113, 38, 123, 36, 34, 72, 116, 35, 12
v7 = [44, 32, 51, 84, 43, 53, 48, 62, 68, 114, 38, 61, 17, 70, 121, 45, 112, 126, 26, 39, 21, 78, 21, 7

v8 = 23

v9 = ''.join(chr(v6[i] ^ ord(chr(v8)[i % len(chr(v8))])) for i in range(len(v6)))
v10 = ''.join(chr(v7[i] ^ ord(v9[i % len(v9)])) for i in range(len(v7)))
print(v9)
print(v10)
```

flag: uoftctf{d0cx_f1l35_c4n_run_c0de_t000}

Baby's First IoT Flag 4 (IoT)

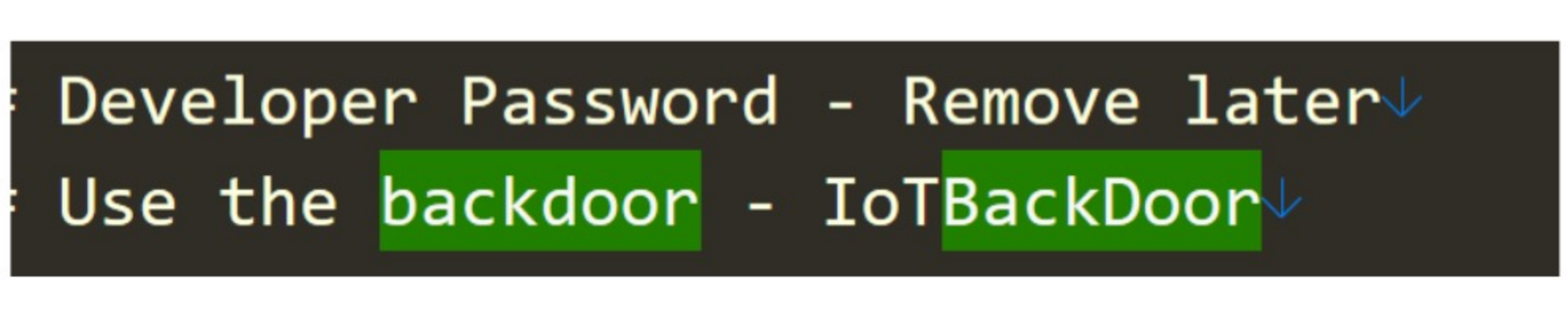


it's firmware.bin so we extract the content of it

```shell
binwalk -extract --dd=".*" firmware1.bin
cd _firmware1.bin.extracted
```

we saw many files but we know there is backdoor file so we try to grep it

```shell
grep -r backdoor
```



so we know have password let's get the flag

```
printf 'IoTBackDoor\n\0' | nc 35.225.17.48 4545
```

flag: uoftctf{Develper_BackDoor}

Forensics, IoT

backdoor · task schedule · trevorc2 · firmware · crypto · network · wireshark · pdf · unredacted

 Share:

Further Reading

Feb 10, 2024
0xl4ugh2024
I participated in the 0xl4ugh2024 I solved all forensics challenge,and th…

OLDER
-

NEWER
0xl4ugh2024