WRITEUP GAME OF HACKERS (GOH) UNIKL 2023

USIM Team Name: sk1d_s3c

Flag format: gohunikl2023{}

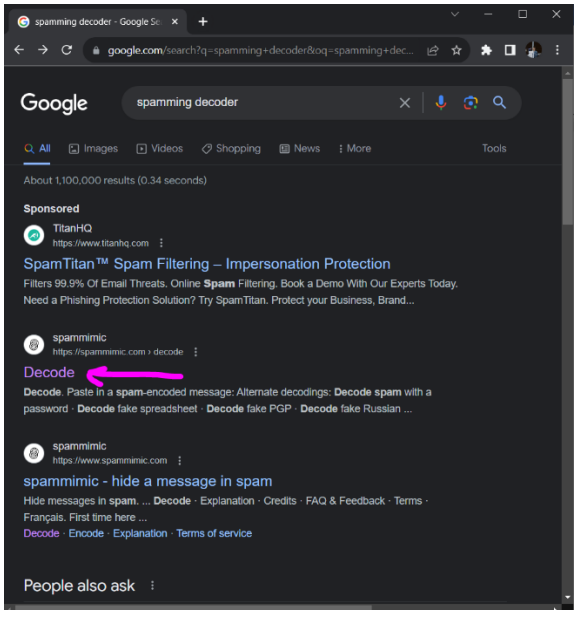
CRYPTOGRAPHY

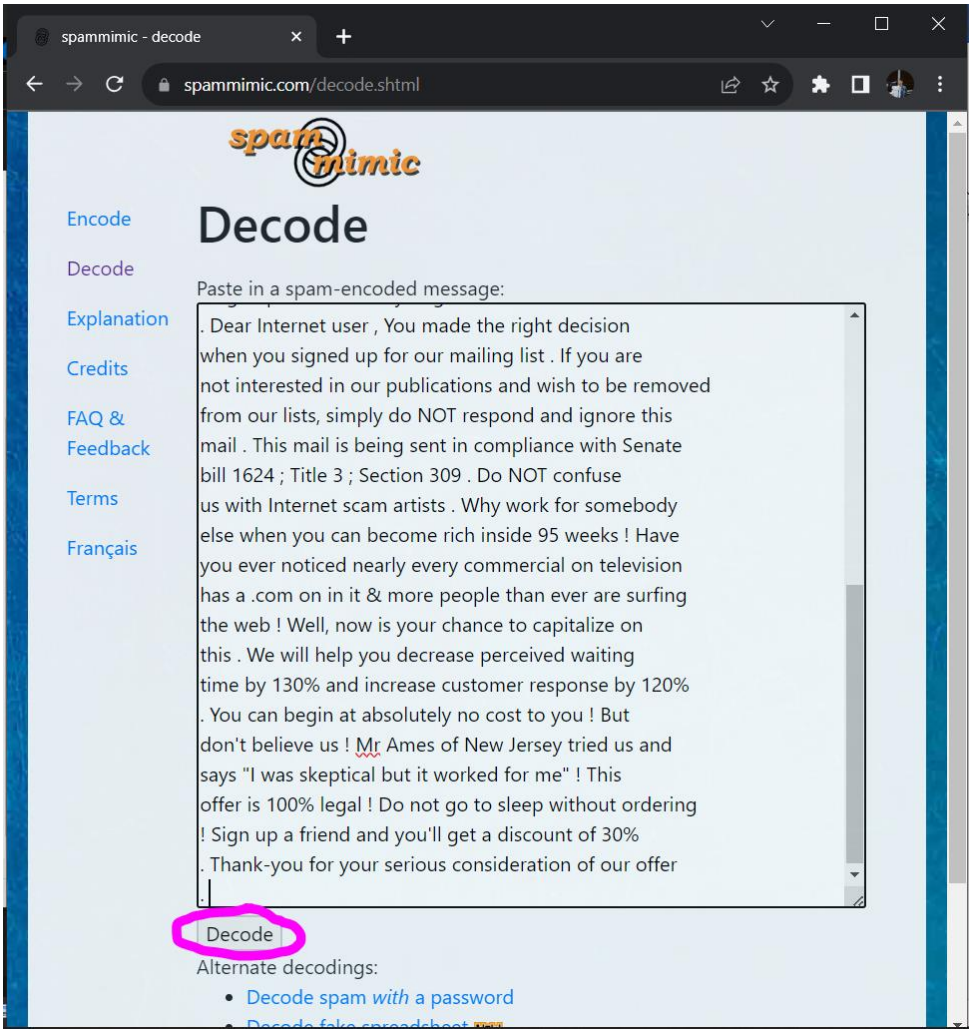
ReadMe (100 Points) Credit to: Danish

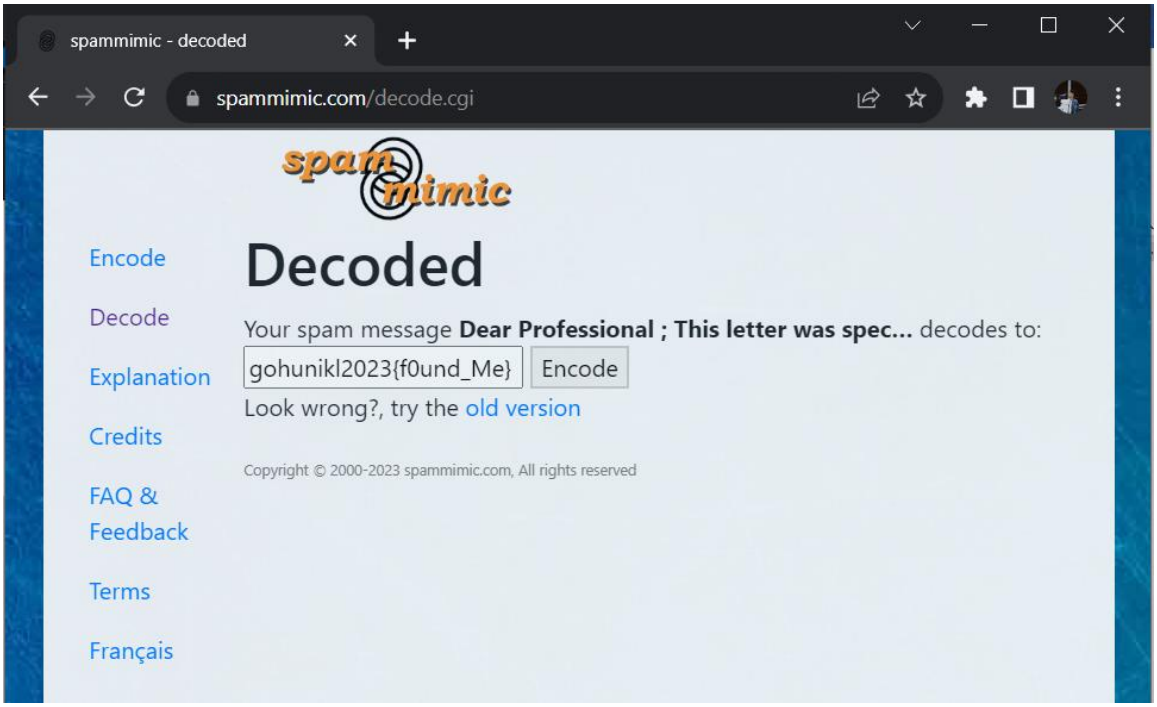
Given ReadMe.txt:

Dear Professional ; This letter was specially selected
to be sent to you . If you no longer wish to receive
our publications simply reply with a Subject: of "REMOVE"
and you will immediately be removed from our mailing
list . This mail is being sent in compliance with Senate
bill 2416 ; Title 5 , Section 301 ! THIS IS NOT A GET
RICH SCHEME ! Why work for somebody else when you can
become rich within 25 DAYS . Have you ever noticed
nobody is getting any younger and people are much more
likely to BUY with a credit card than cash ! Well,
now is your chance to capitalize on this ! WE will
help YOU deliver goods right to the customer's doorstep
and increase customer response by 120% ! You can begin
at absolutely no cost to you ! But don't believe us
. Mrs Simpson who resides in Idaho tried us and says
"I've been poor and I've been rich - rich is better"
. We are a BBB member in good standing . We urge you
to contact us today for your own future financial well-being
! Sign up a friend and you get half off ! God Bless
. Dear Internet user , You made the right decision
when you signed up for our mailing list . If you are
not interested in our publications and wish to be removed
from our lists, simply do NOT respond and ignore this
mail . This mail is being sent in compliance with Senate
bill 1624 ; Title 3 ; Section 309 . Do NOT confuse
us with Internet scam artists . Why work for somebody
else when you can become rich inside 95 weeks ! Have
you ever noticed nearly every commercial on television
has a .com on in it & more people than ever are surfing
the web ! Well, now is your chance to capitalize on
this . We will help you decrease perceived waiting
time by 130% and increase customer response by 120%
. You can begin at absolutely no cost to you ! But
don't believe us ! Mr Ames of New Jersey tried us and
says "I was skeptical but it worked for me" ! This
offer is 100% legal ! Do not go to sleep without ordering
! Sign up a friend and you'll get a discount of 30%
. Thank-you for your serious consideration of our offer
.

We tried the Acrostic Extractor, Trevanion Cipher and Cardan Grille but they do not end well. So we search spamming decoder on the Internet.







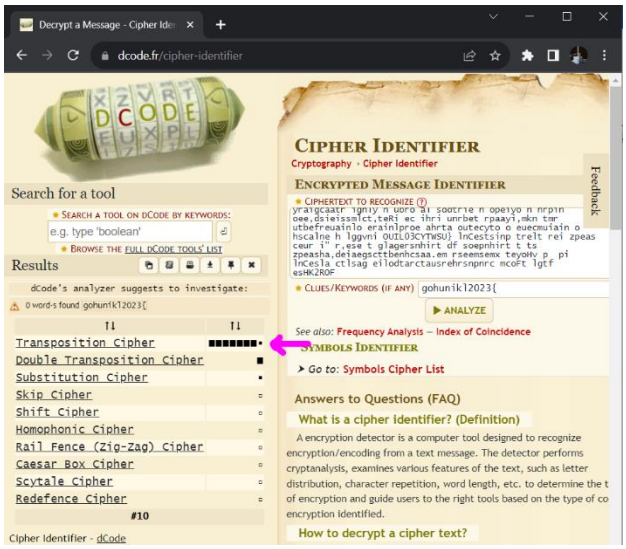
The flag is gohunikl2023{f0und_Me}

Cipher 101 (100 Points)

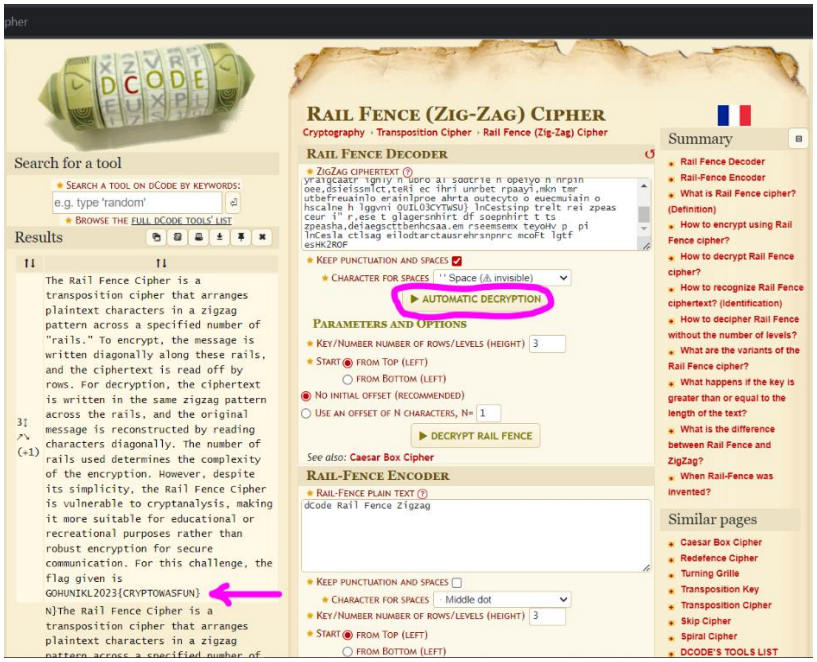
PS: About two teams solves

Given text.txt:

haFep aaoicehan ixhcs igtnr pf borsTnptmaireinyotei heieobo ri heirenemigtnr latoi s rnud d redolTnefiu entcltfect.wrettiiyhaFep veloyns iiosa co reo pst nb rifsront.rihe,eai GN2{PANTeRi ec ihri rnpsto ihrta ragspanetcaatr nazga atr cosaseiidnme f"al. oecyt h esg switndaoal ln hs al,adtecpetx sra f yrw.Frdcyto,tecpetx switni h aezga atr costeris n h rgnlmsaei eosrce yraigcaatr ignly h ubro al sddtrie h opeiyo h nrpin oee,dsieissmlct,teRi ec ihri unrbet rpaayi,mkn tmr utbefreuainlo erainlproe ahrta outecyto o euecmuiain o hscalne h lggvni OUIL03CYTWSU} lnCestsinp trelt rei zpeas ceur i" r,ese t glagersnhirt df soepnhirt t ts zpeasha,deiaegscttbenhcsaa.em rseemsemx teyoHv p pi lnCesla ctlsag eilodtarctausrehrsnpnrc mcoFt lgtf esHK2ROF



Although it said it is Transposition Cipher, but it does not end well too. So, Rail Fence Cipher is pretty common these days.



We need to hit the AUTOMATIC DECRYPTION until it shows up readable texts with flag.
The flag is GOUNIKL2023{CRYPTOWASFUN}

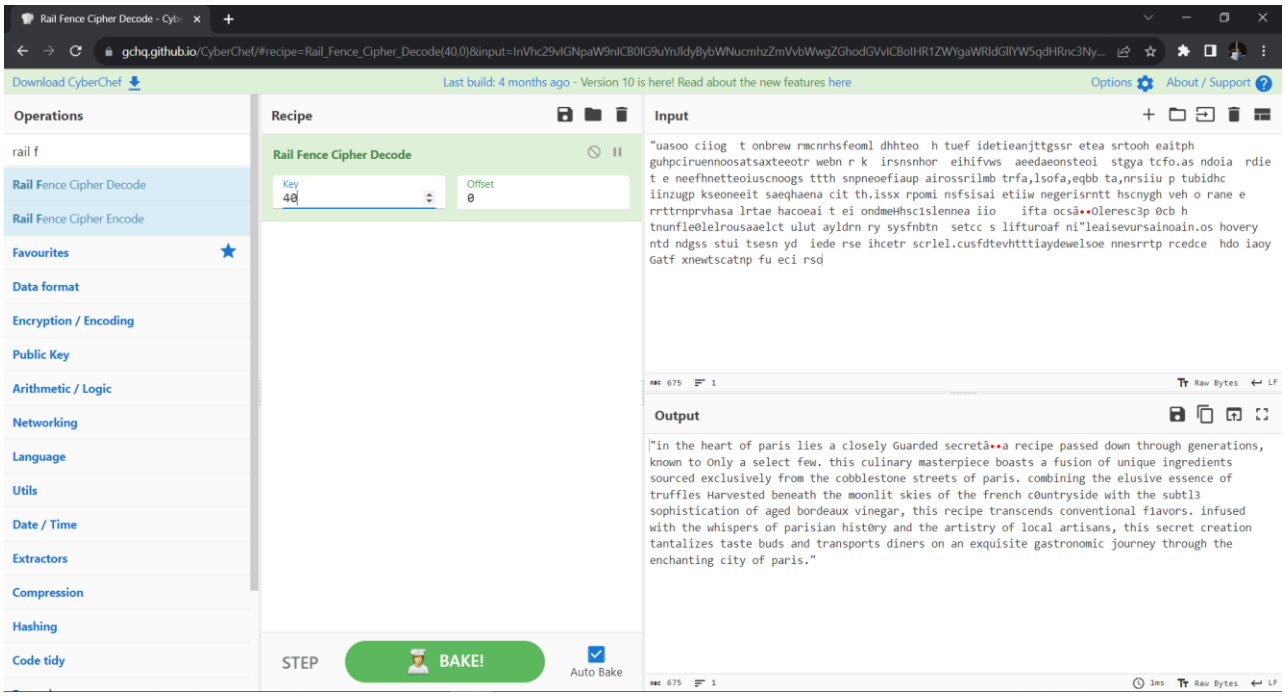
Train Of Bleu (150 Points)

PS: About three teams managed to get the flag

Given Secret_Recipe.zip (AES encryption password) and P4ssc0de.txt:

"uasoo ciiog t onbrew rmcnrhsfeoml dhhteo h tuef idetieanjttgssr etea srtooh eaitph guhpciruennoosatsaxteeotr webn r k irsnsnhor eihifvws aeedaeonsteoi stgya tcfo.as ndoia rdie t e neefhnetteoiuscnoogs ttth snpneoefiaup airossrilmb trfa,lsofa,eqbb ta,nrsiiu p tubidhc iinzugp kseoneeit saeqhaena cit th.issx rpomi nsfsisai etiiw negerisrntt hscnygh veh o rane e rrttrnprvhasa lrtae hacoeai t ei ondmeHhsc1slennea iio ifta ocsâ€ ” Oleresc3p 0cb h tnunfle0lelrousaaelct ulut ayldrn ry sysfnbtn setcc s lifturoaf ni"leaisevursainoain.os hovery ntd ndgss stui tsesn yd iede rse ihcetr scrlel.cusfdtevhtttiaydewelsoe nnesrrtp rcedce hdo iaoy Gatf xnewtscatnp fu eci rso

Cipher Identifier from dcode.fr and boxentriq cannot detect which cipher. Once again, we use the Rail Fence Cipher. This time, we need to use the Rail Fence Decoder in CyberChef. Why? Because given the hint number 40 from the file name itself, P4ssc0de.



Output retrieved:

"in the heart of paris lies a closely Guarded secret—a recipe passed down through generations, known to Only a select few. this culinary masterpiece boasts a fusion of unique ingredients sourced exclusively from the cobblestone streets of paris. combining the elusive essence of truffles Harvested beneath the moonlit skies of the french c0untryside with the subtl3 sophistication of aged bordeaux vinegar, this recipe transcends conventional f1avors. infused with the whispers of parisian hist0ry and the artistry of local artisans, this secret creation tantalizes taste buds and transports diners on an exquisite gastronomic journey through the enchanting city of paris."

We noticed unusual capital letters and numbers, so we can take it as a clue to unlock the password-protected Secret_Recipe.zip file.
The clue is GOH0310 (password for Secret_Recipe.zip)

After unzipping the Secret_Recipe.zip, we obtained a Secret_Recipe.txt:

Parisian Secret Croissants

Ingredients:
2 1/4 cups all-purpose flour
1/4 cup granulated sugar
1 teaspoon salt

1 tablespoon active dry yeast
1/2 cup warm water
1 cup cold unsalted butter, sliced thinly
1/2 cup milk
1 egg, beaten (for egg wash)

Instructions:
1. Activate the Yeast: In a small bowl, dissolve the yeast in warm water. Let it sit for 5-10 minutes until frothy.

2. Prepare the Dough: In a large mixing bowl, combine flour, sugar, and salt. Add the activated yeast mixture and milk. Mix until a dough forms.

3. Incorporate the Butter: Roll out the dough on a floured surface into a rectangle. Place the thinly sliced cold butter evenly over two-thirds of the dough. Fold the unbuttered third over the buttered middle third, then fold the other buttered third over the top. This creates three layers.

4. Chill the Dough: Wrap the dough in plastic wrap and refrigerate for 30 minutes.

5. Roll and Fold: Remove the dough from the refrigerator and roll it out into a rectangle again. Fold it into thirds, as before. Chill for another 30 minutes.

6. Repeat Rolling and Folding: Repeat the rolling and folding process two more times, chilling the dough for 30 minutes between each step. This creates the flaky layers characteristic of croissants.

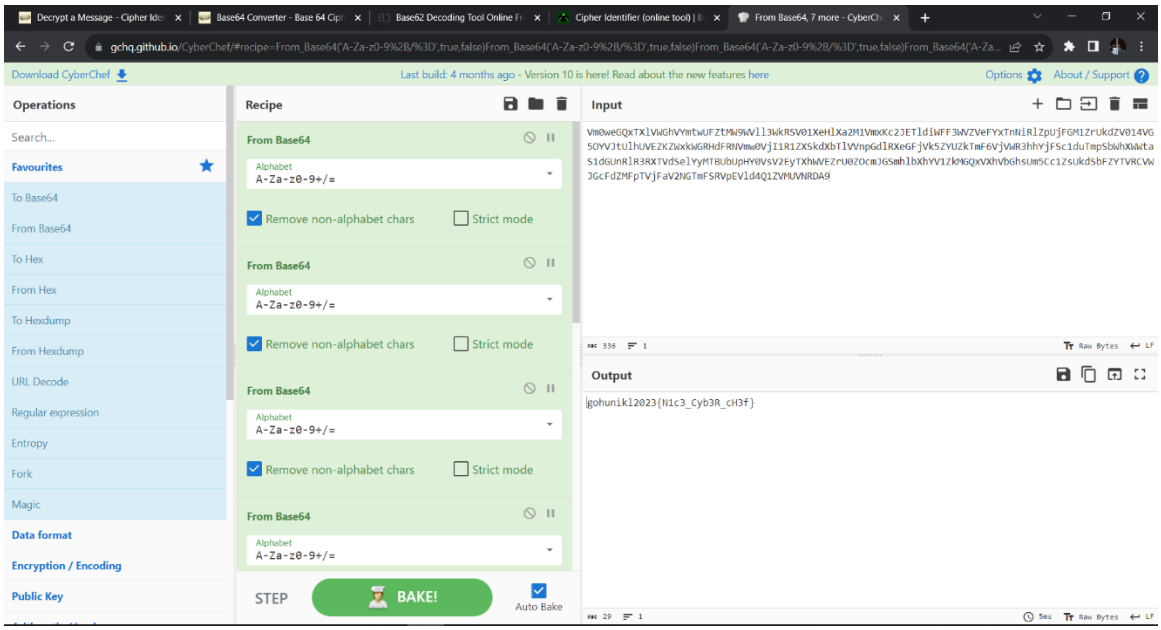
7. Shape the Croissants: Roll out the dough into a large rectangle and cut it into triangles. Roll each triangle starting from the wide end towards the tip to form the classic croissant shape.

8. Proofing: Place the shaped croissants on a baking sheet lined with parchment paper. Cover them loosely with a kitchen towel and let them proof in a warm place for about 1-2 hours, or until doubled in size.

9. Preheat and Bake: Preheat the oven to 375°F (190°C). Brush the croissants with beaten egg for a shiny finish. Bake for 15-20 minutes or until golden brown.

10. Enjoy the secret recipe from
Vm0weGQxTXlVWGhVYmtwUFZtMW9WVll3WkRSV01XeHlXa2M1VmxKc2JETldiWFF3WVZVeFYxTnNiRlZpUjFGM1ZrUkdZV014VG5OYVJtUlhUVEZKZWxkWGRHdFRNVmw0VjI1R1ZXSkdXbTlVVnpGdlRXeGFjVk5ZYUZkTmF6VjVWR3hhYjFSc1duTmpSbWhXWWtaS1dGUnRlR3RXTVdSelYyMTBUbUpHY0VsV2EyTXhWVEZrU0ZOcmJGSmhlbXhYV1ZkMGQxVXhVbGhsUm5Cc1ZsUkdSbFZYTVRCVWJGcFdZMFpTVjFaV2NGTmFSRVpEVld4Q1ZVMUVNRDA9.

This secret recipe holds the key to the quintessential Parisian croissant—crispy on the outside, tender and flaky on the inside, a true delight for the senses!

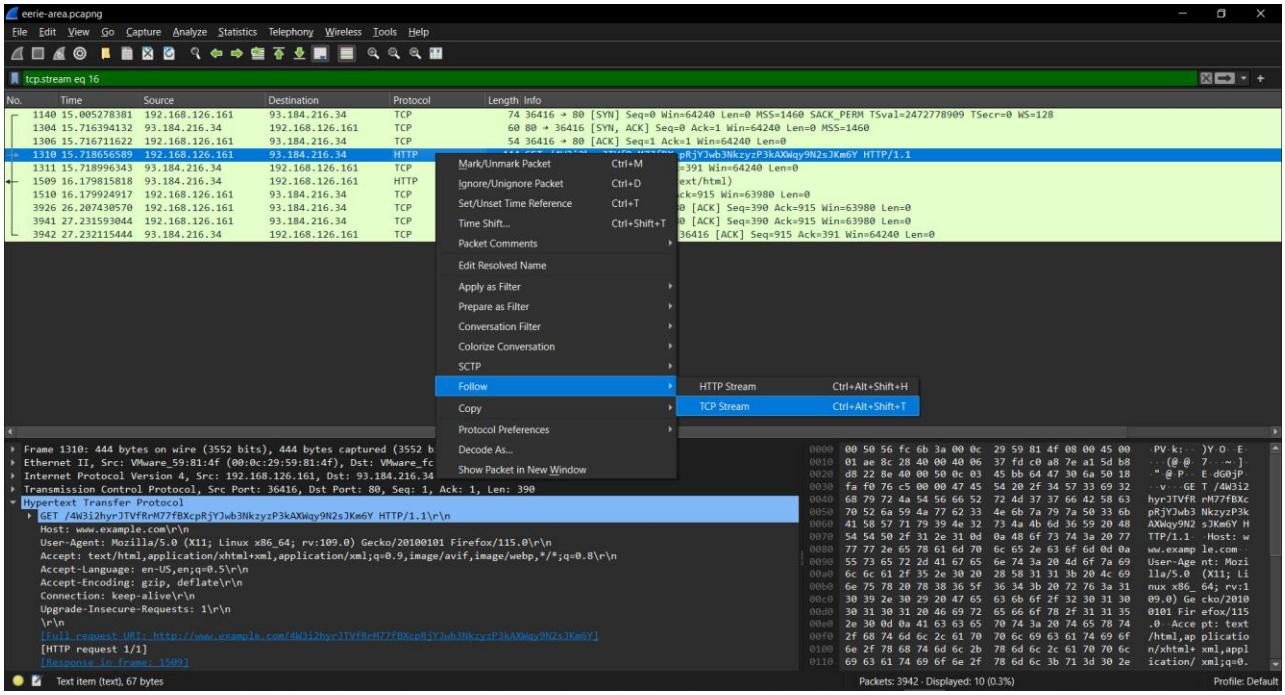


Use CyberChef again using From Base64 8 times to retrieve the flag: gohunikl2023{N1c3_Cyb3R_cH3f}

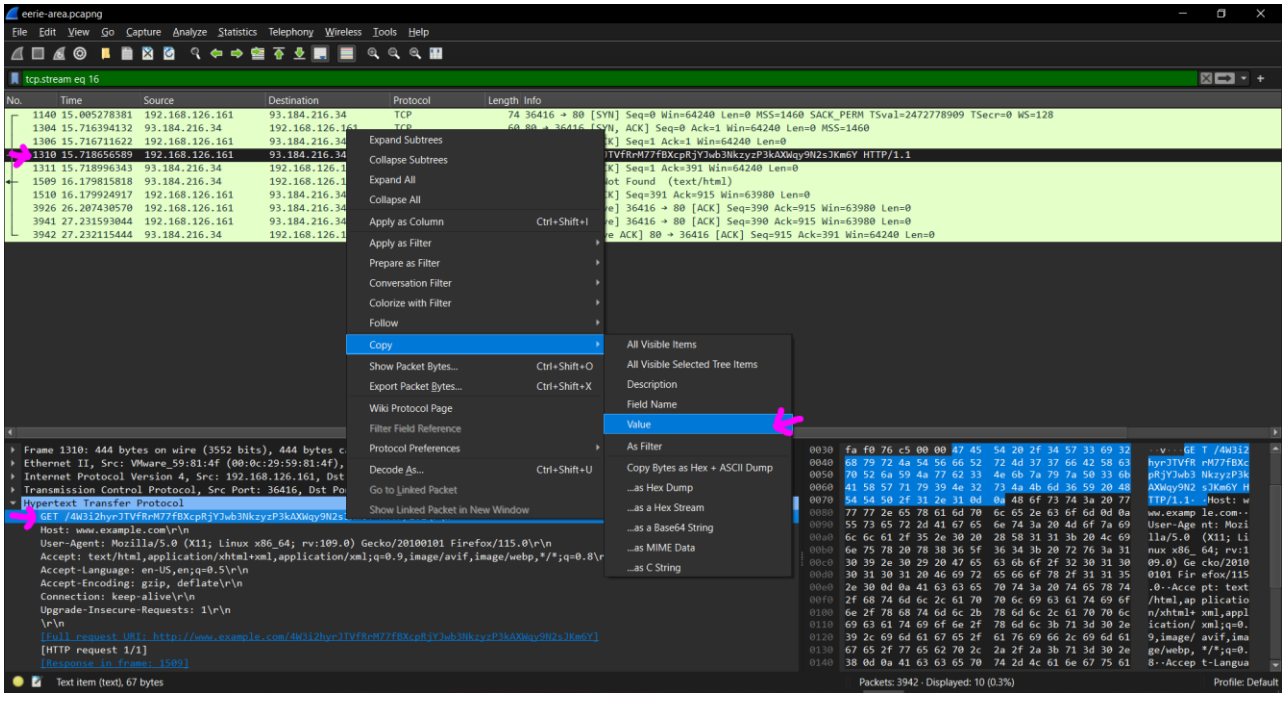
FORENSIC

The realm of Eerie-Area (150 Points) Credit to: Danish

Given eerie-area.pcapng

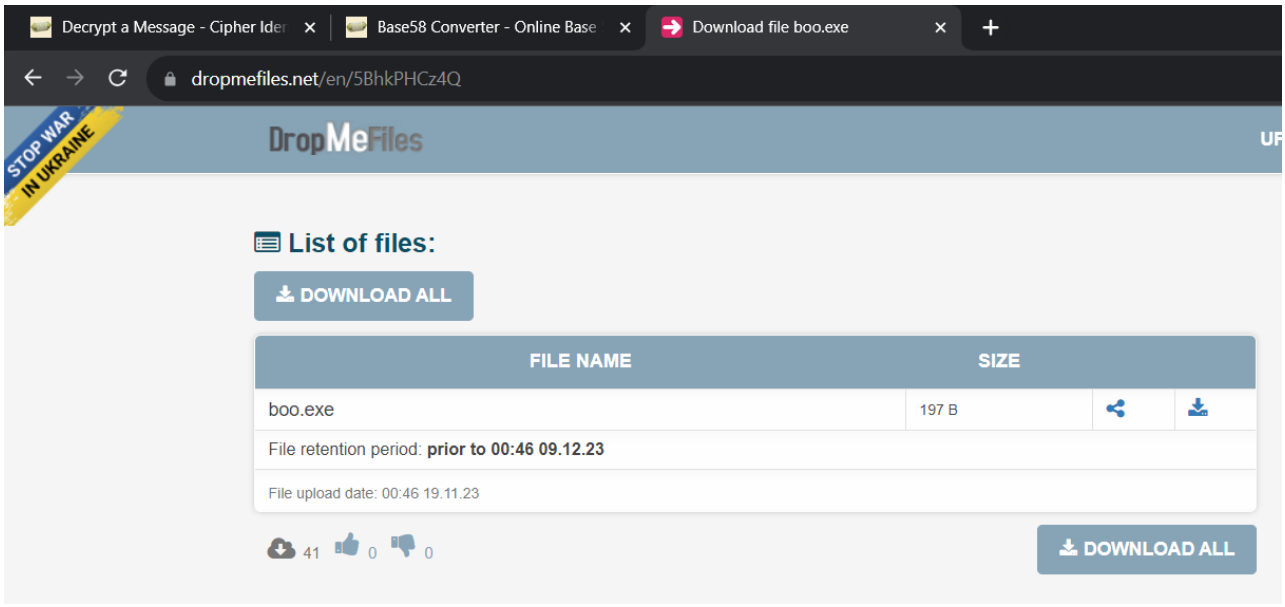


First, you need to right-click on anything with TCP Protocol. We found something unusual text which is a Base58 text. Copy the value of the HTTP.



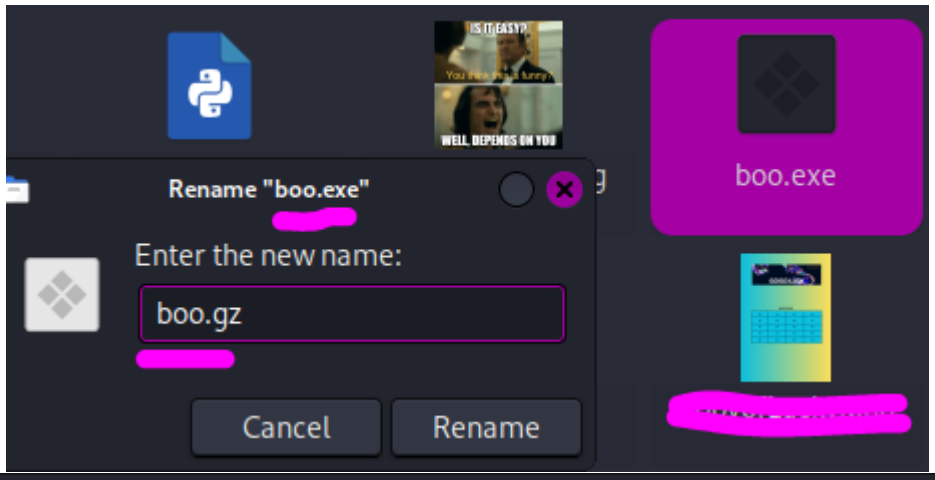
4W3i2hyrJTVfRrM77fBXcpRjYJwb3NkzyzP3kAXWqy9N2sJKm6Y

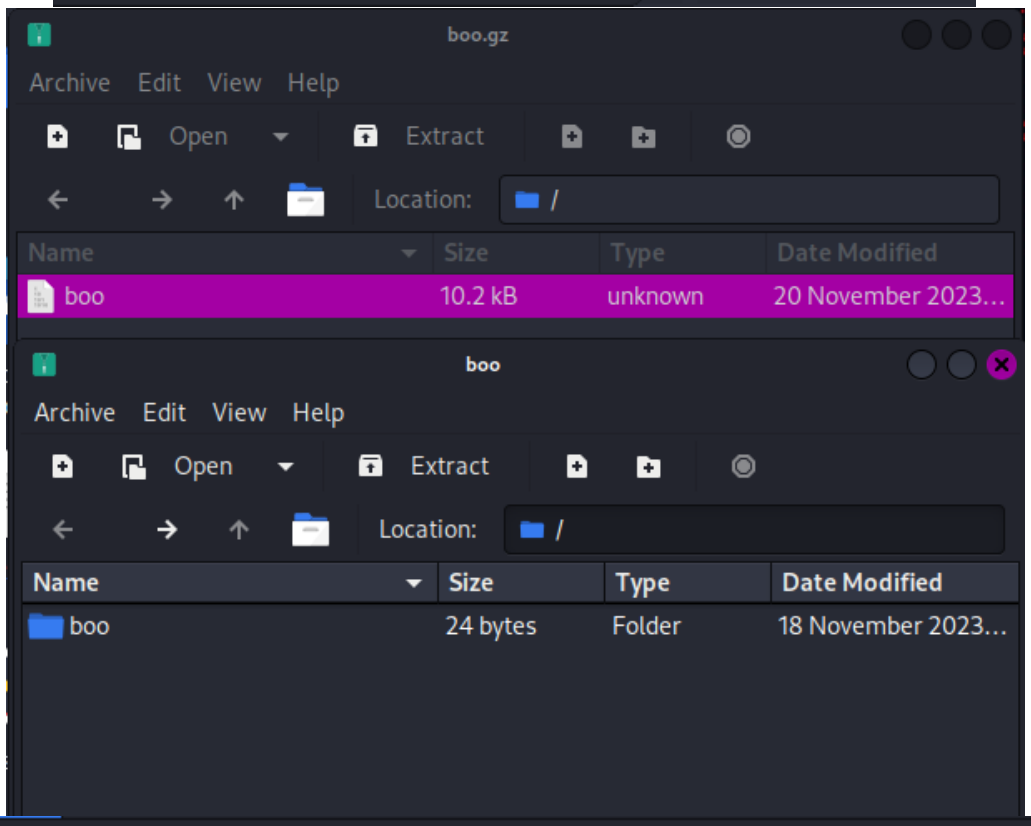
A website was retrieved : https://dropmefiles.net/en/5BhkPHCz4Q

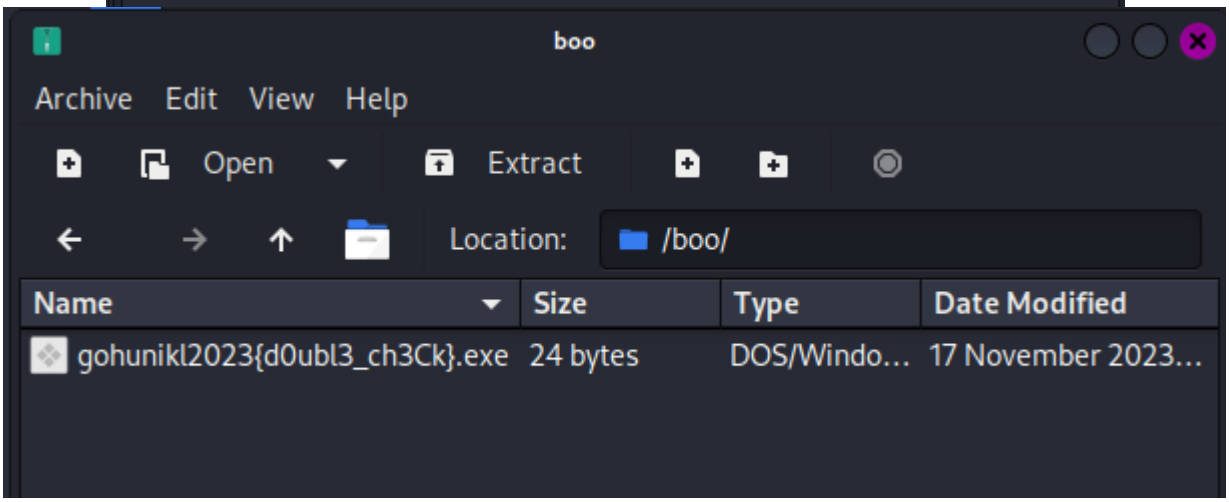


boo.exe retrieved that can't be executed in Windows. Every time we obtained a suspicious file. Use Kali Linux to use command "file" to check the exact file extension.

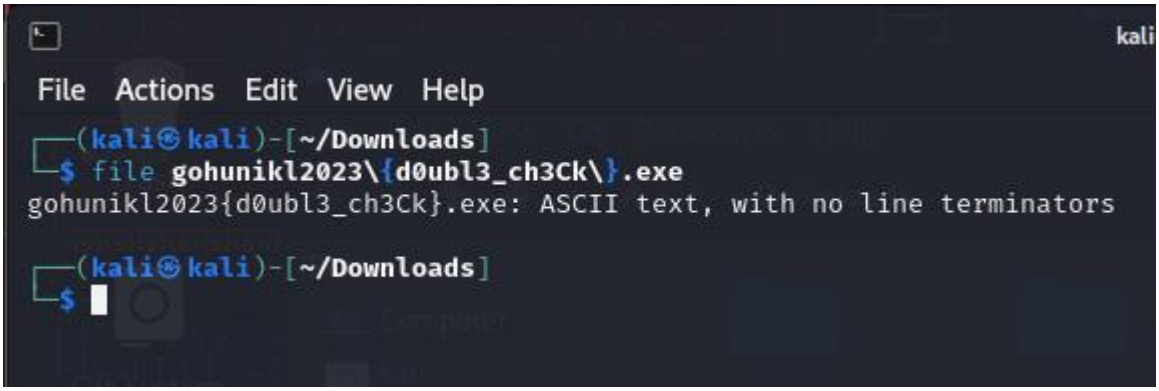
So it should be in .gz file format so rename it with .gz extension. Then, unzip it to retrieve an object named boo.



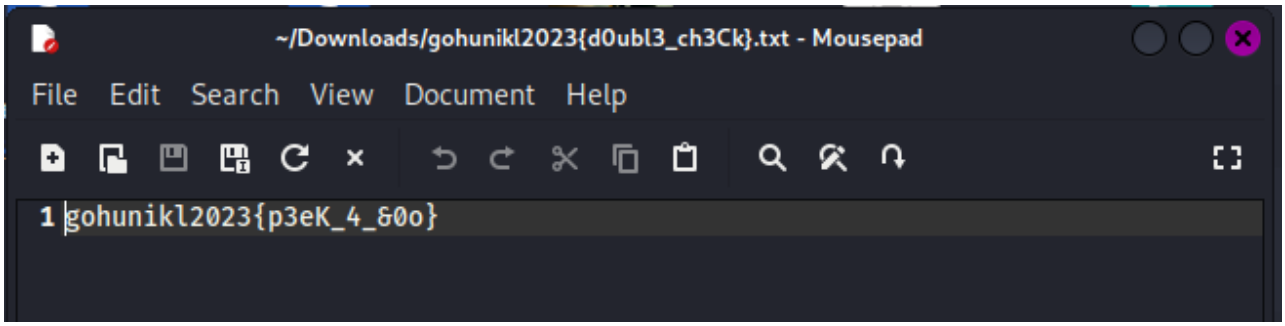




So this is not the flag yet. Double check means you have to use the ‘file’ Kali Linux command again.



Rename it to .txt

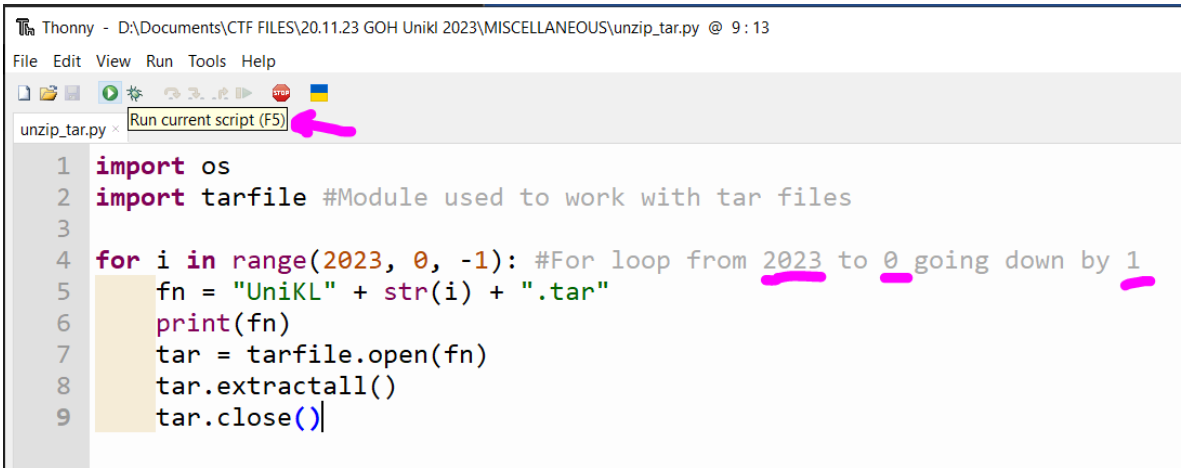


The flag is gohunikl2023{p3eK_4_&0o}

MISCELLANEOUS

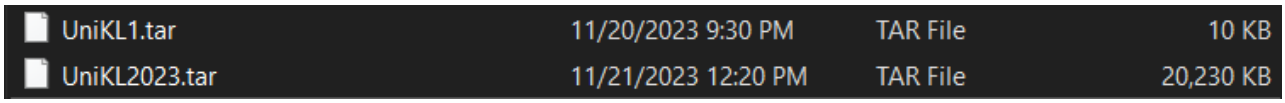
Just Unzip (100 Points)

Given UniKL2023.tar (a compressed file that can be extracted). First extract will get us UniKL2022.tar
So we need to craft a small Python script for it in order to unzip it until it gets UniKL1.tar (make sure the Python script in the same folder with UniKL2023.tar



```
import os
import tarfile #Module used to work with tar files

for i in range(2023, 0, -1): #For loop from 2023 to 0 going down by 1
      fn = "UniKL" + str(i) + ".tar"
      print(fn)
      tar = tarfile.open(fn)
      tar.extractall()
      tar.close()
```

PS: From 20mb to about 15GB of total files XD



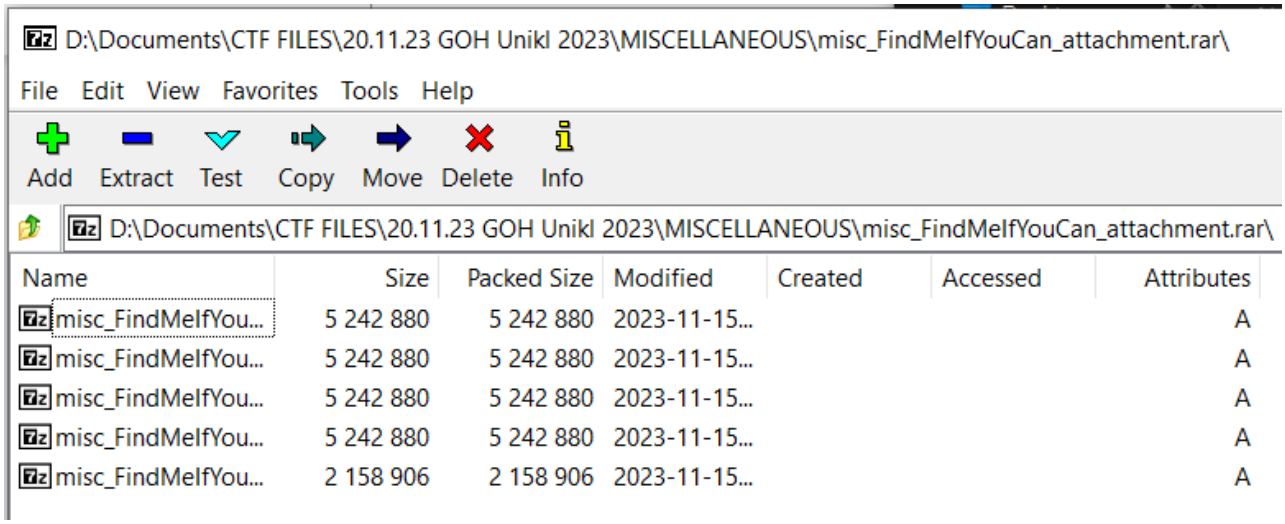
unzip the UniKL1.tar and we will obtain a picture of flag.png

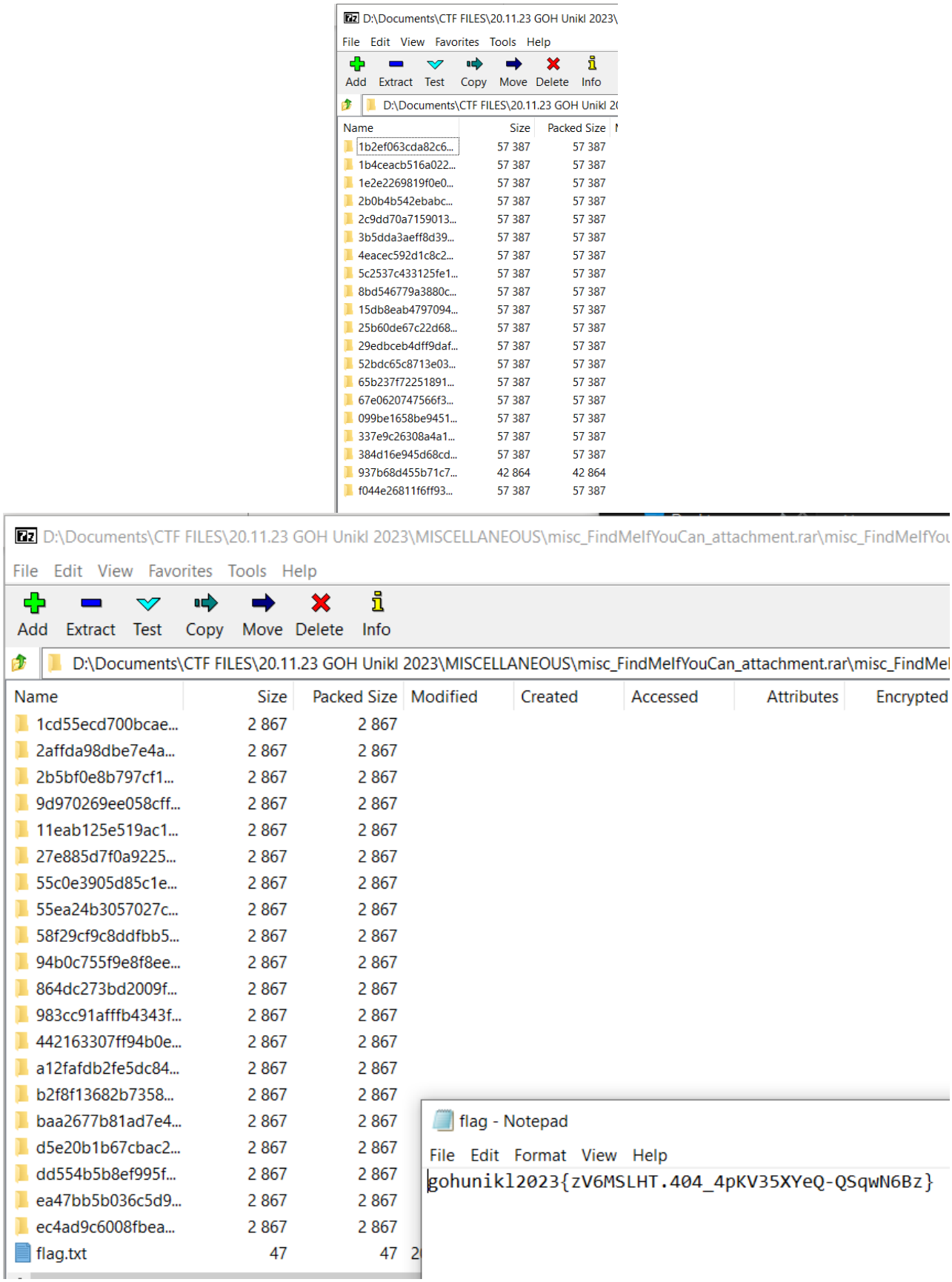
The flag is gohunikl2023{1_H0p3_U_D0nt_d0_1t_2023_TiMe}

FindMeIfYouCan (100 Points)

PS: About 8 teams found the flag

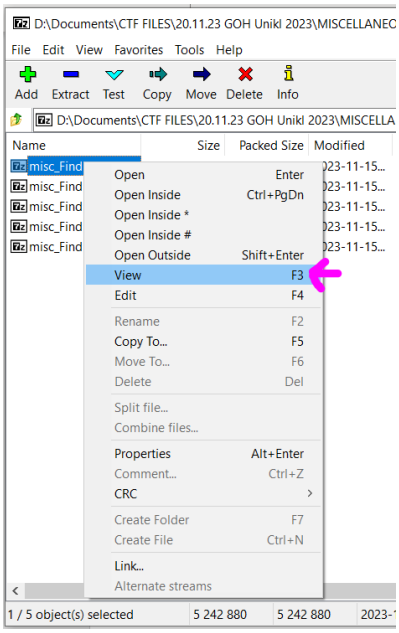
From misc_FindMeIfYouCan_attachment.rar , we need to find flag





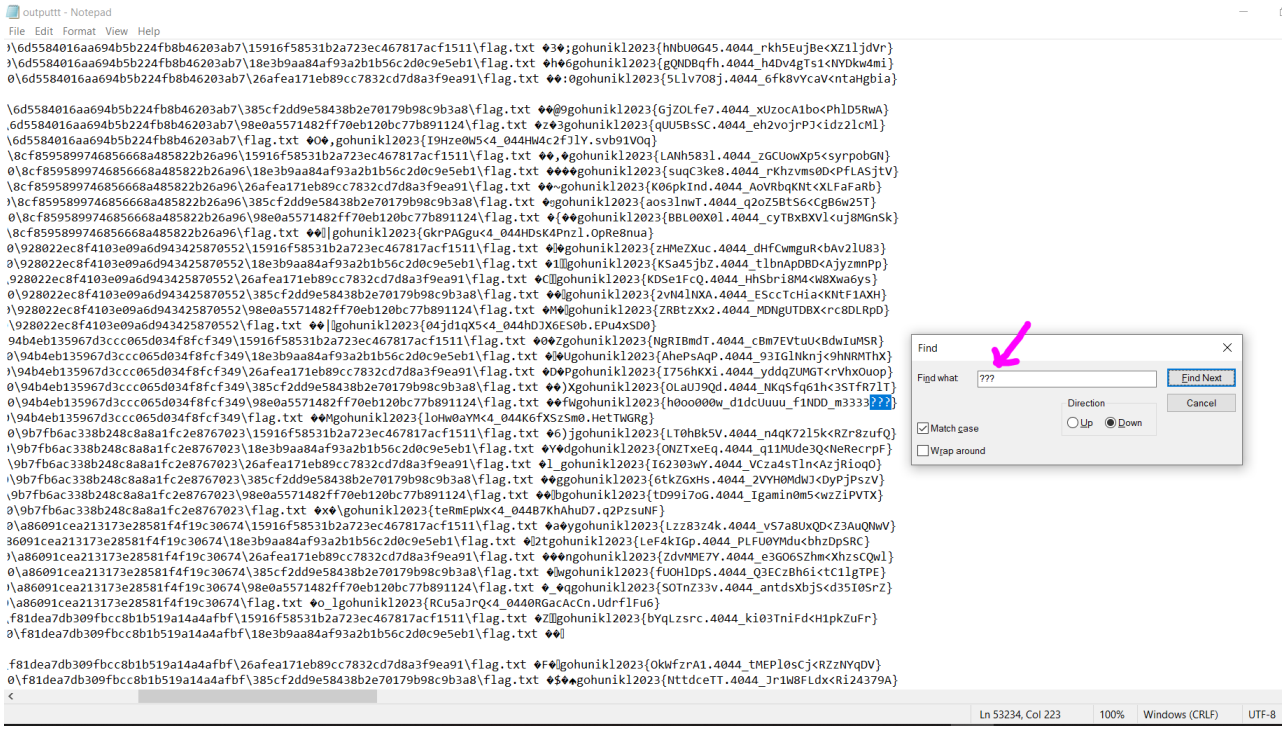
As you can see, there are a lot of folders inside folders, and sometimes you will obtain the flag.txt containing fake flags (totally a lot, like every folder has one each)

However, there's a tool that ease your flag finding by using this step:



This output window will be shown:

You can view another 4 .rar files too and you will notice that there are no flag in the 4th and 5th rar. Thus, things get easier.
Copy and paste all 3 viewing output into a new .txt file.



There are 2 ways to find the flag:

1) Find unique symbols such as '???', '!' and try your luck.
2) Bring this to the Kali Linux, use the grep command to filter out 4044,404,40,4_ (observed the flag pattern) by using the command: grep 'gohunikl2023' output.txt | grep -Ev '4044|404|40|4_' (recommended)

Reference Link: https://osxdaily.com/2018/04/05/how-exclude-word-grep-command-line/

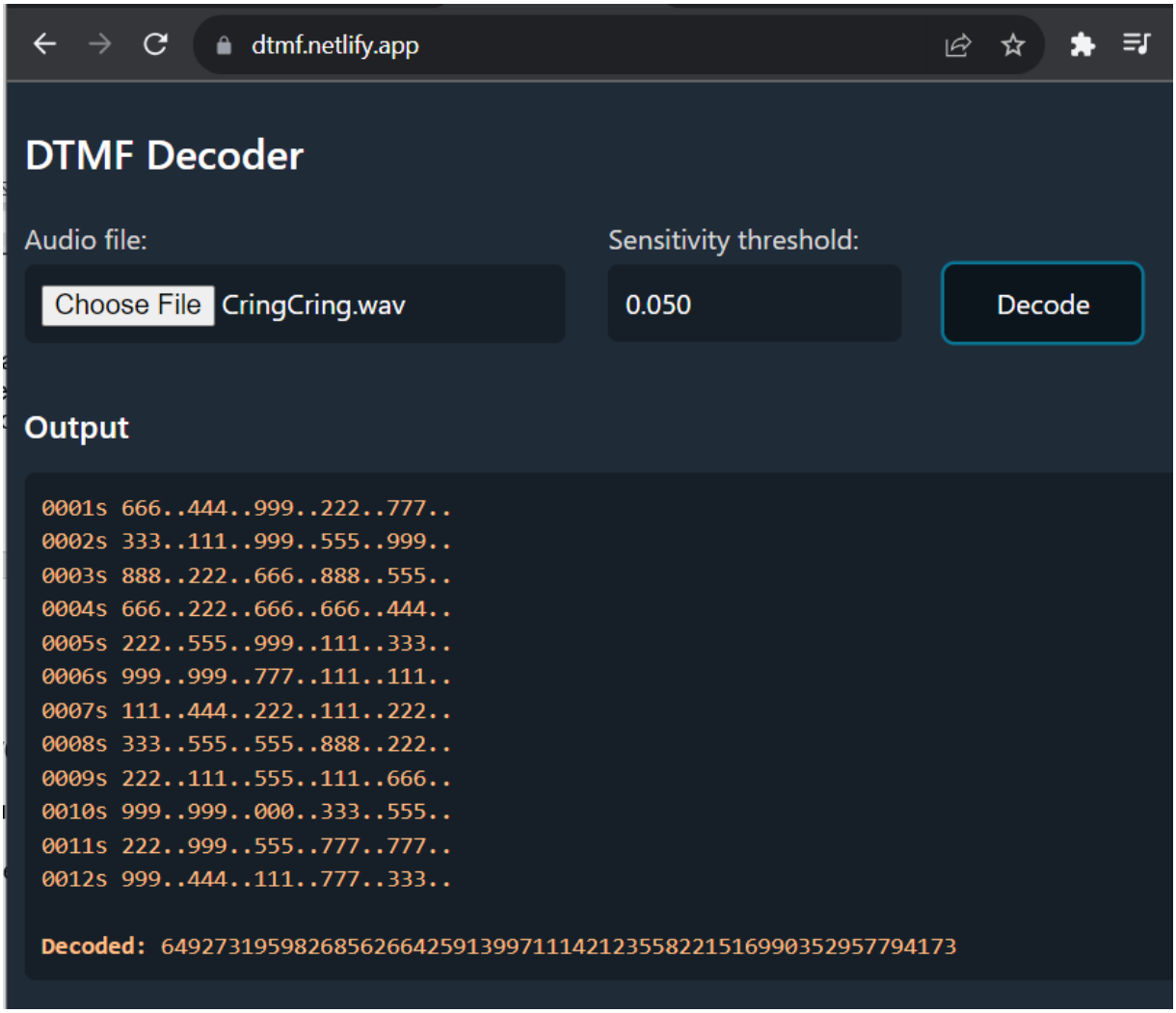
The flag is gohunikl2023{h0oo000w_d1dcUuuu_f1NDD_m3333???}

STEGANOGRAPHY

Cring (50 Points) Credit to: Danish

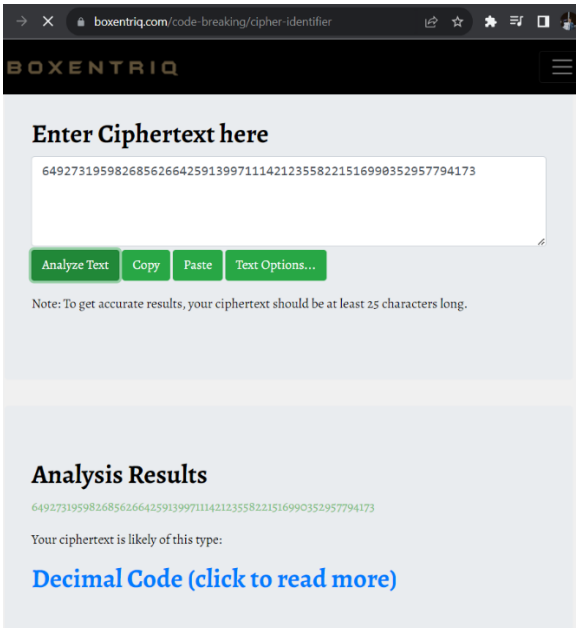
Given CringCring.wav file with short old phone sounds (12 seconds of dialling sounds)

Every audio that relates to telephone, we can use the DTMF Decoder tool.

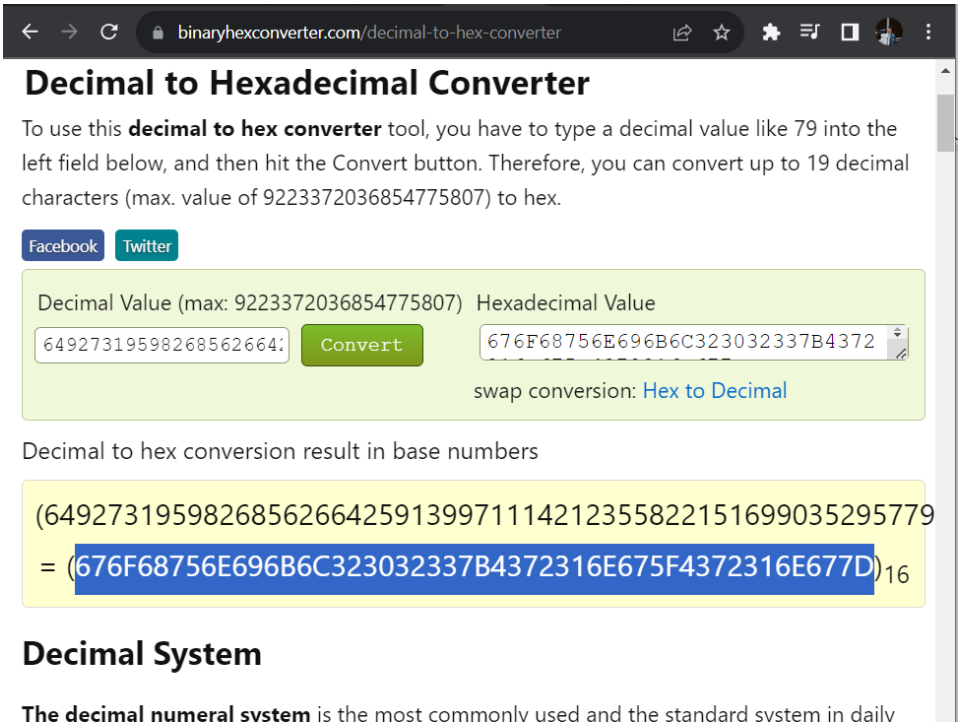


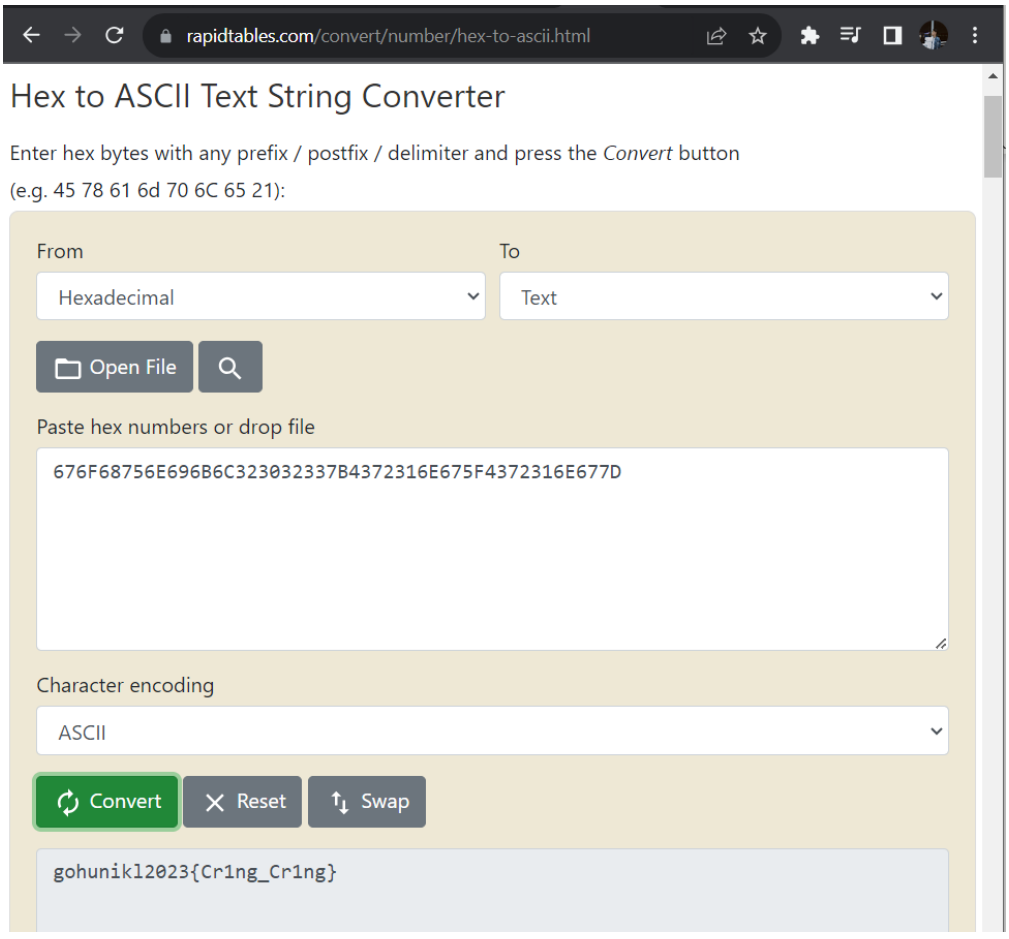
Decoded: 649273195982685626642591399711142123558221516990352957794173

We brought the decoded output of numbers to Cipher Identifier Boxentriq.



In order to decode the decimal code, you must convert it to hex first.
Then, convert from hex to text (ascii).





The flag is gohunikl2023{Cr1ng_Cr1ng}

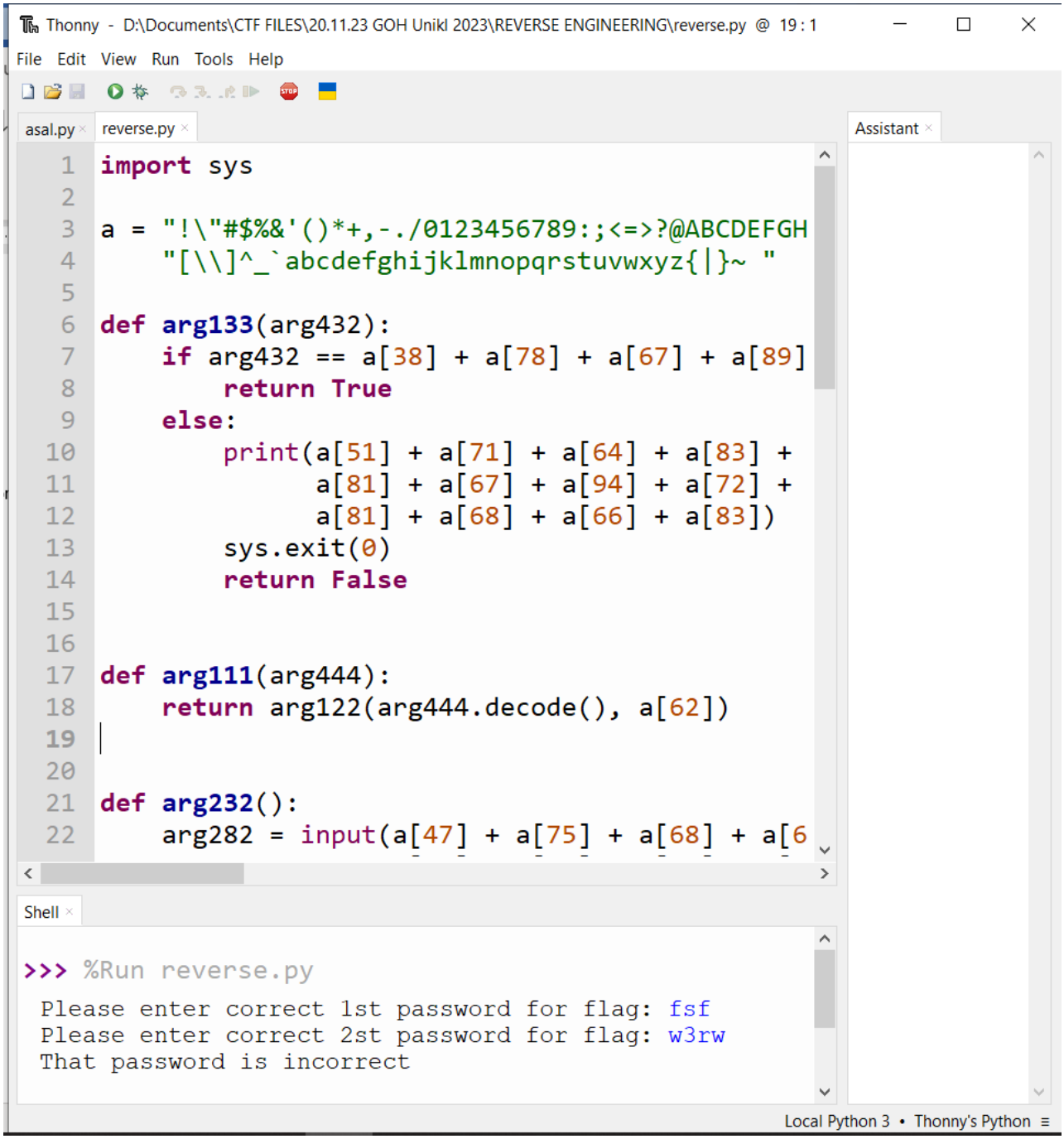
REVERSE ENGINEERING

Reverse Script (50 Points) Credit to: Haziq

Given flag.txt.enc and a Python script reverse.py:

```
import sys

a = "!\"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ" + \
    "[\\]^_`abcdefghijklmnopqrstuvwxyz{|}~ "

def arg133(arg432):
    if arg432 == a[38] + a[78] + a[67] + a[89] + a[72] + a[43] + a[75] + a[31] + a[53] + a[50] + a[42] + a[16] + a[77] + a[70]
+ a[38] + a[71] + a[16] + a[67] + a[15] + a[81] + a[19] + a[39]:
        return True
    else:
        print(a[51] + a[71] + a[64] + a[83] + a[94] + a[79] + a[64] + a[82] + a[82] + a[86] + a[78] +
              a[81] + a[67] + a[94] + a[72] + a[82] + a[94] + a[72] + a[77] + a[66] + a[78] + a[81] +
              a[81] + a[68] + a[66] + a[83])
        sys.exit(0)
        return False


def arg111(arg444):
    return arg122(arg444.decode(), a[62])


def arg232():
    arg282 = input(a[47] + a[75] + a[68] + a[64] + a[82] + a[68] + a[94] + a[68] + a[77] + a[83] +
                   a[68] + a[81] + a[94] + a[66] + a[78] + a[81] + a[81] + a[68] + a[66] + a[83] +
                   a[94] + a[16] + a[82] + a[83] + a[94] + a[79] + a[64] + a[82] + a[82] + a[86] +
                   a[78] + a[81] + a[67] + a[94] + a[69] + a[78] + a[81] + a[94] + a[69] + a[75] +
                   a[64] + a[70] + a[25] + a[94])

    if arg282 == '':
        print("Password 1 is empty")
        sys.exit(0)

    arg383 = input(a[47] + a[75] + a[68] + a[64] + a[82] + a[68] + a[94] + a[68] + a[77] + a[83] +
                   a[68] + a[81] + a[94] + a[66] + a[78] + a[81] + a[81] + a[68] + a[66] + a[83] +
                   a[94] + a[17] + a[82] + a[83] + a[94] + a[79] + a[64] + a[82] + a[82] + a[86] +
                   a[78] + a[81] + a[67] + a[94] + a[69] + a[78] + a[81] + a[94] + a[69] + a[75] +
                   a[64] + a[70] + a[25] + a[94])

    if arg383 == '':
        print("Password 2 is empty")
        sys.exit(0)

    if arg282 is not None and arg383 is not None:
        arg888 = arg282 + arg383
        return arg888


def arg132():
    return open('flag.txt.enc', 'rb').read()


def arg112():
    print(a[54] + a[68] + a[75] + a[66] + a[78] + a[76] + a[68] + a[94] + a[65] + a[64] + a[66] +
          a[74] + a[13] + a[13] + a[13] + a[94] + a[88] + a[78] + a[84] + a[81] + a[94] + a[69] +
          a[75] + a[64] + a[70] + a[11] + a[94] + a[84] + a[82] + a[68] + a[81] + a[25])
```

```
def arg122(arg432, arg423):
    arg433 = arg423
    i = 0
    while len(arg433) < len(arg432):
        arg433 = arg433 + arg423[i]
        i = (i + 1) % len(arg423)
    return "".join([chr(ord(arg422) ^ ord(arg442)) for (arg422, arg442) in zip(arg432, arg433)])

arg444 = arg132()
arg432 = arg232()
arg133(arg432)
arg112()
arg423 = arg111(arg444)
print(arg423)
sys.exit(0)
```



First, execute the script to analyze the code. After a session with Blackbox AI, it said:

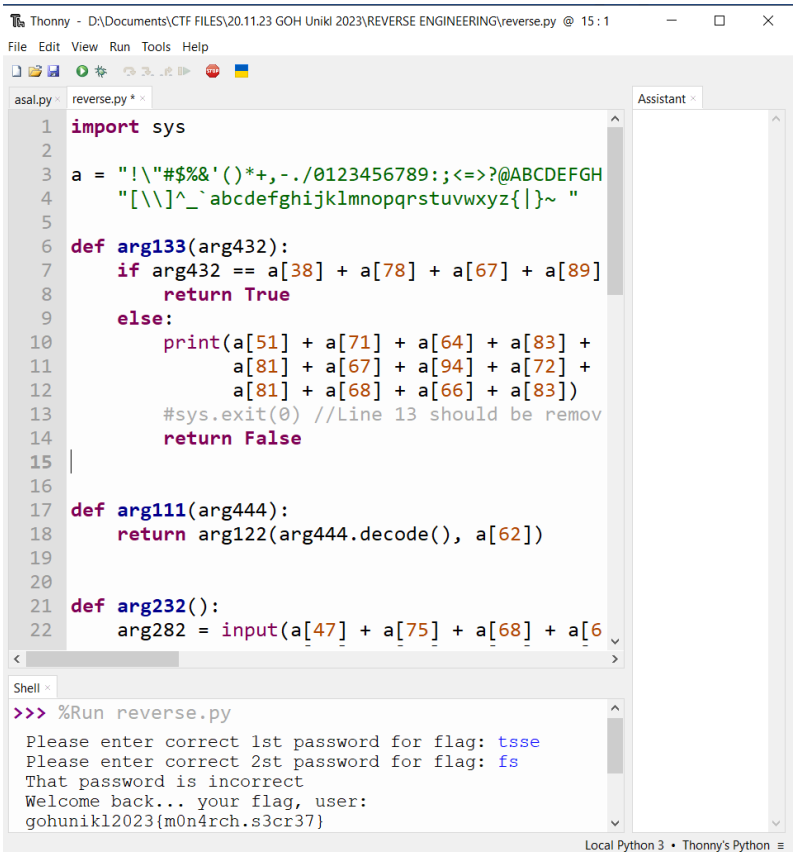
The program takes three arguments and processes them to return the correct decryption key. It utilizes XOR operations and repeating sequences of characters.

The function arg232() takes the first argument. If it's empty, the program terminates.
The function arg332() takes the second argument. If it's empty, the program terminates.
The function arg444() takes the third argument.

Solution:

```
def arg133(arg432):
    if arg432 == a[38] + a[78] + a[67] + a[89] + a[72] + a[43] + a[75] +
a[31] + a[53] + a[50] + a[42] + a[16] + a[77] + a[70] + a[38] + a[71] +
a[16] + a[67] + a[15] + a[81] + a[19] + a[39]:
        return True
    else:
        print(a[51] + a[71] + a[64] + a[83] + a[94] + a[79] + a[64] +
a[82] + a[82] + a[86] + a[78] +
              a[81] + a[67] + a[94] + a[72] + a[82] + a[94] + a[72] +
a[77] + a[66] + a[78] + a[81] +
              a[81] + a[68] + a[66] + a[83])
        #sys.exit(0) //Line 13 should be removed
        return False
```

Line 13 ‘sys.exit(0)’ should be removed to prevent process exit.

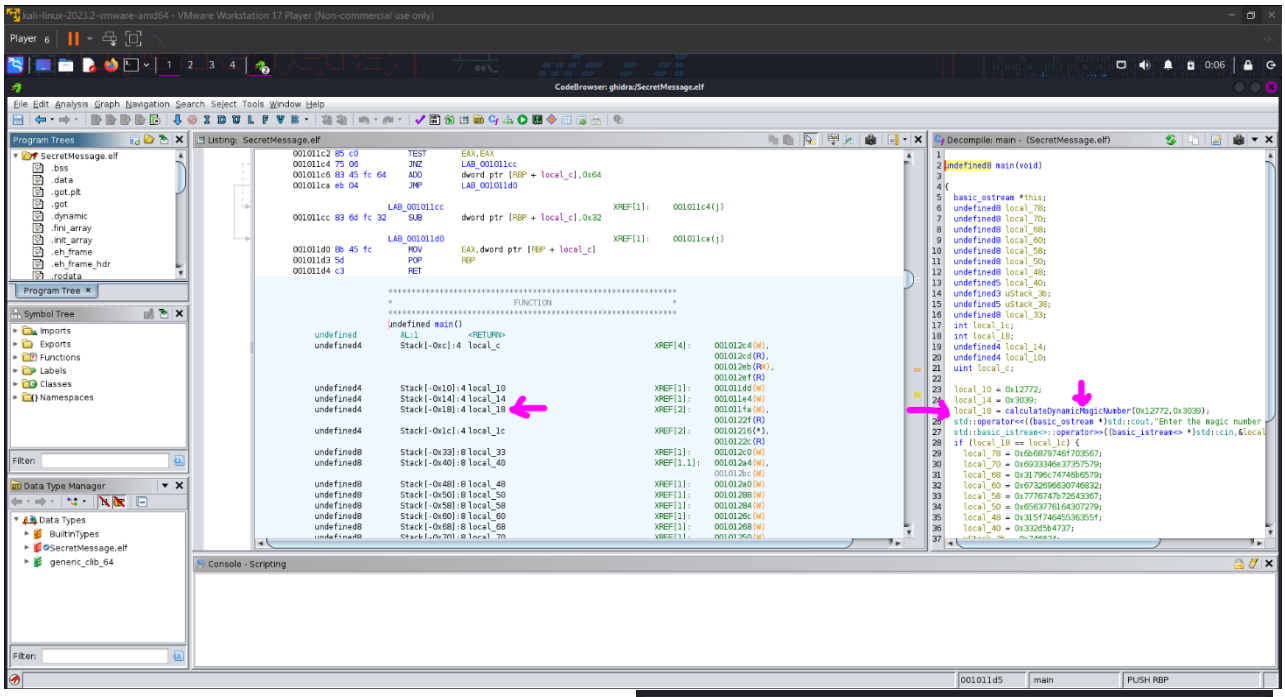


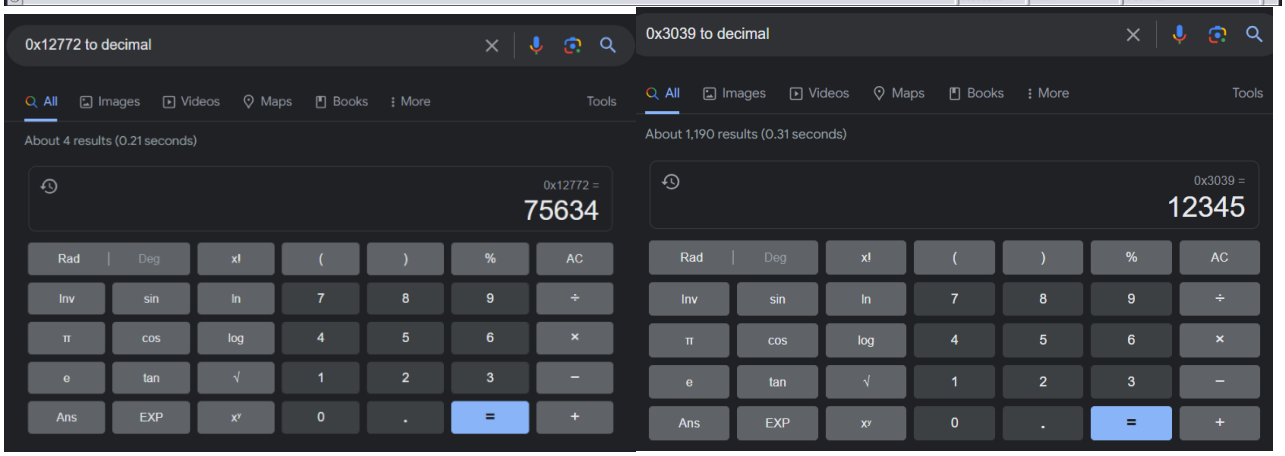
The flag is gohunikl2023{m0n4rch.s3cr37}

Magic Number (100 Points) Credit to: Haziq

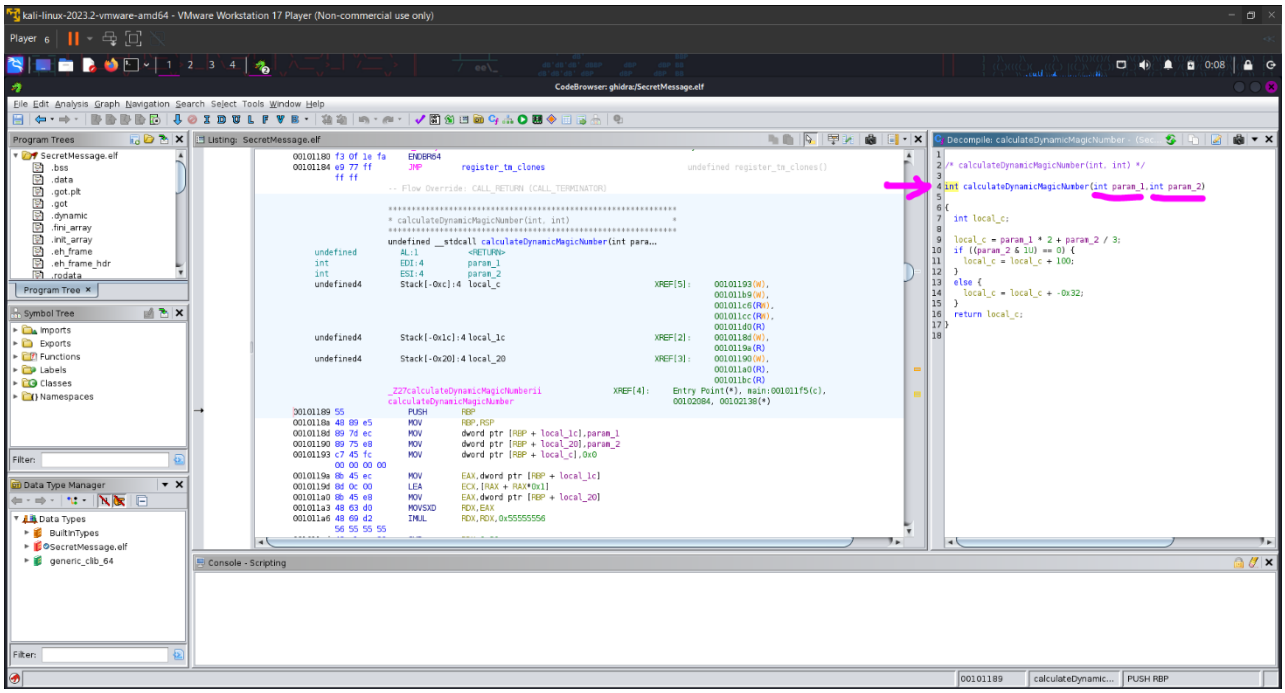
Given an ELF file SecretMessage.elf

For ELF file, we can practice ghidra in Kali Linux with reference link:
https://youtu.be/oTD_ki86c9I?si=9OeVSQrkTOkiFDsC&t=450



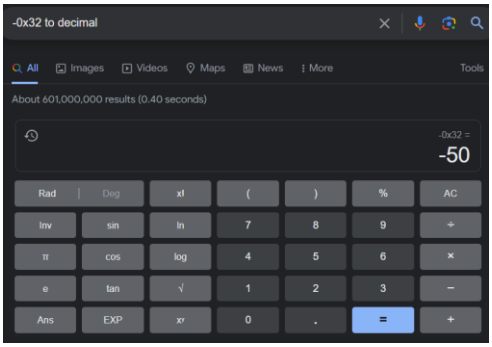


As you can see, there is a function local_18 = calculateDynamicMagicNumber(0x12772,0x3039); with converted value parameter 75634 and 12345 in decimal respectively. Double click on the calculateDynamicMagicNumber function.



We can observe the calculation source code for param_1 and param_2.
So we can calculate the 'magic number' with the parameter given.

```
Decompile: calculateDynamicMagicNumber - (Sec...

1
2  /* calculateDynamicMagicNumber(int, int) */
3
4  int calculateDynamicMagicNumber(int param_1,int param_2)
5
6  {
7    int local_c;
8
9    local_c = param_1 * 2 + param_2 / 3;
10   if ((param_2 & 1U) == 0) {
11     local_c = local_c + 100;
12   }
13   else {
14     local_c = local_c + -0x32;
15   }
16   return local_c;
17 }
18
```



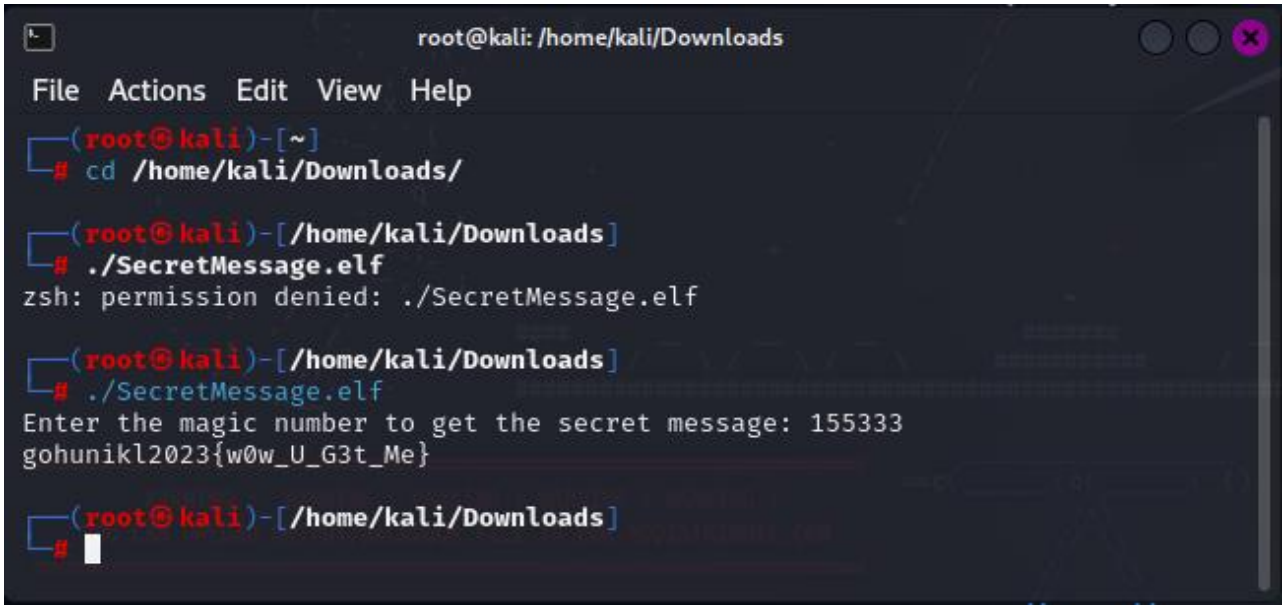
param_1 = 75634, param_2 = 12345

75634 * 2 = 151268 , 12345 / 5 = 4115
151268 + 4115 = 155383

So value local_c is 155,383.

Next, the if statement checks if 12345 is even or odd. Since 12345 is odd, the process continue to the else statement. The local_c value will minus 50 (155383 – 50). And BOOMM! That's the magic number.. 155333

Now, execute the .elf file with the input of the magic number 155333 and the flag will be printed out.



The flag is gohunikl2023{w0w_U_G3t_Me}

Reverse It (150 Points) Credit to: Haziq

Given the description of the challenge: 杯桉潼胥㈠㈱筎 | 敝剥癥牳叫杼

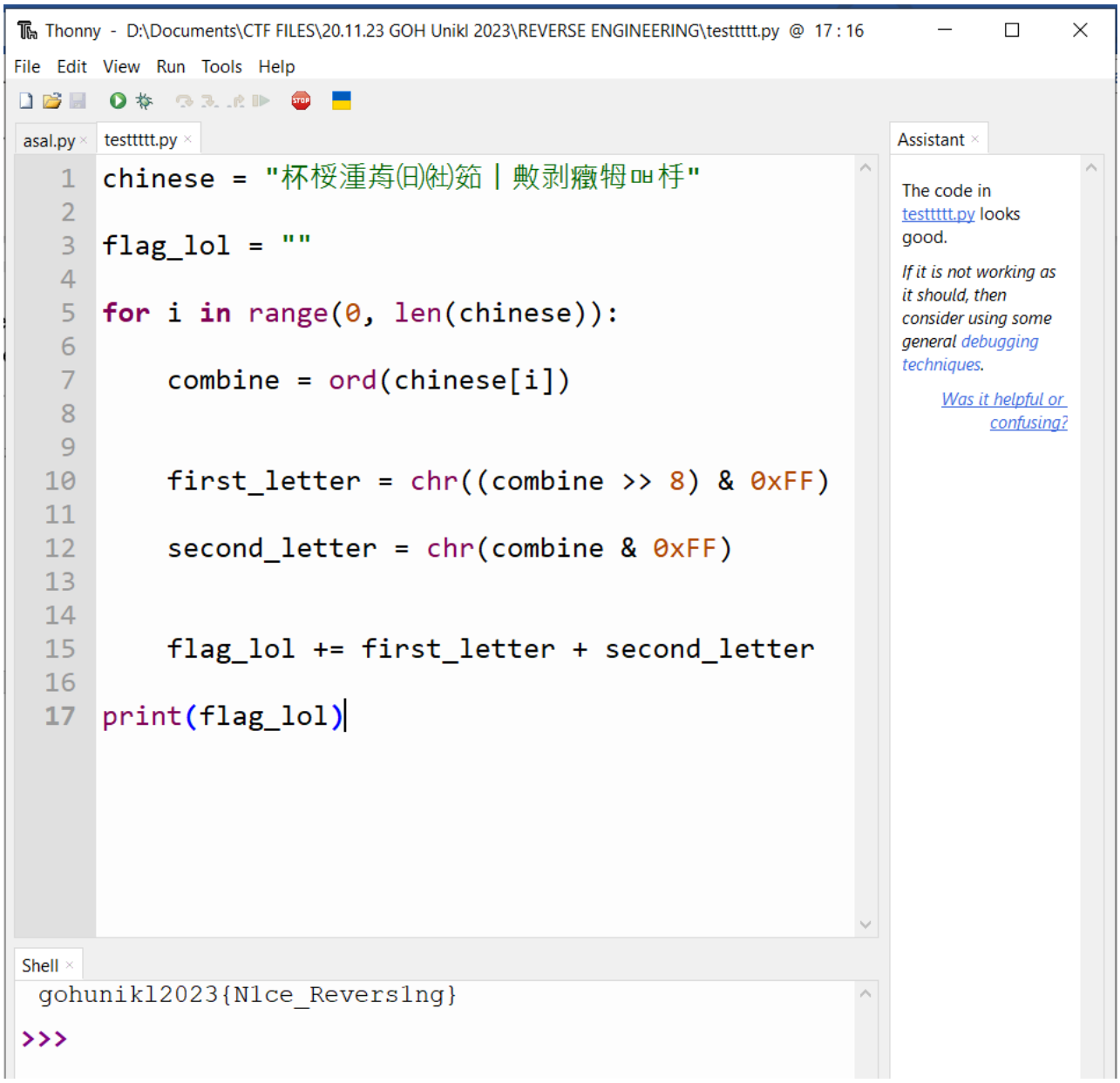
And a python code:

```
my_string = "My_Strings"
my_result = ""

for i in range(0, len(my_string), 2):
    my_first_letter = ord(my_string[i]) << 8
    my_second_letter = ord(my_string[i + 1])
    my_final = chr(my_first_letter + my_second_letter)
    my_result += my_final

print(my_result)
```

It seems that the Python works as ascii encoder to (like Chinese) code..
Reverse it by creating a new Python code to decode this given clue 杯桉潼胥㈠㈱筎 | 敝剥癥牳叫杼 ..... a little bit hardwork of coding this is the python code to decode.

```
chinese = "杯桉潼胥㈠㈱筎 | 敝剥癥牳叫杼"

flag_lol = ""

for i in range(0, len(chinese)):

    combine = ord(chinese[i])
```

```
    first_letter = chr((combine >> 8) & 0xFF)
    second_letter = chr(combine & 0xFF)

    flag_lol += first_letter + second_letter
print(flag_lol)
```



The flag is gohunikl2023{N1ce_Revers1ng}