HGame2024 Week1 Writeup

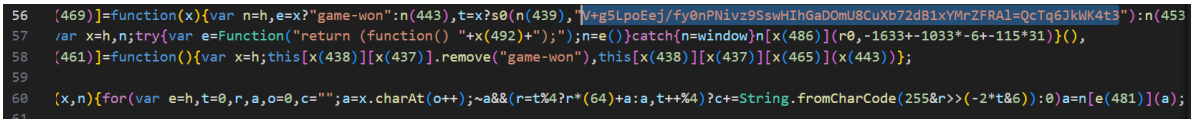
Web

2048*16

> 很好的2048，使我的大脑旋转

要求玩到32768，显然不可能是手动去玩~~（我手动最高记录才玩到2048）~~。估计flag就藏在前端，查看javascript，里面有许多混淆代码。手搓还原了一小部分获胜后的核心代码，看起来好像是个换表base64。找到表和密文后，解码得到flag。

表如下



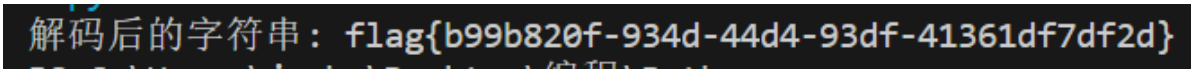
```
{469)]=function(x){var n=h,e=x?"game-won":n(443),t=x?s0(n(439),"V+g5LpoEej/fy0nPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3"):n(453
/ar x=h,n;try{var e=Function("return (function() "+x(492)+");");n=e()}catch{n=window}n[x(486)](r0,-1633+-1033*-6+-115*31)}(),
{461)]=function(){var x=h;this[x(438)][x(437)].remove("game-won"),this[x(438)][x(437)][x(465)](x(443))};

{x,n){for(var e=h,t=0,r,a,o=0,c="";a=x.charAt(o++);~a&&(r=t%4?r*(64)+a:a,t++%4)?c+=String.fromCharCode(255&r>>(-2*t&6)):0)a=n[e(481)](a);
```

密文如下

```
"I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK"
```

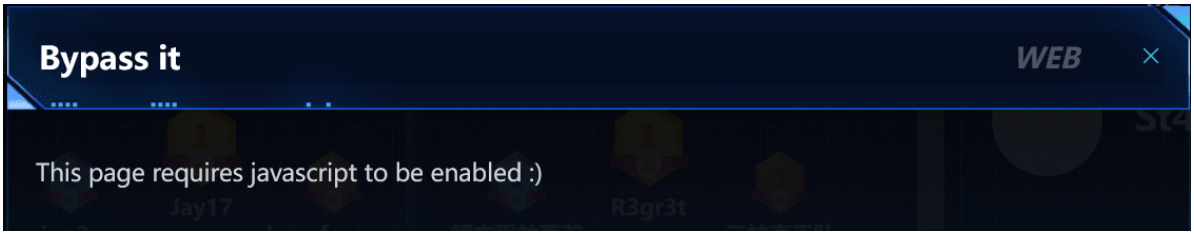
解码代码如下

```
def base64_decode(encoded_str):
    # Base64字符映射表
    base64_chars =
"V+g5LpoEej/fyOnPNivz9SswHIhGaDOmU8CuXb72dB1xYMrZFRAl=QcTq6JkWK4t3"
    # 将每个Base64字符转换为其6位二进制形式
    char_to_bin = {char: bin(index)[2:].zfill(6) for index, char in
enumerate(base64_chars)}
    # 移除Base64编码中的填充字符
    encoded_str = encoded_str.rstrip('=')
    # 解码过程
    binary_str = ''.join([char_to_bin[char] for char in encoded_str])
    # 将二进制字符串分成每8位一组，转换为字节
    bytes_list = [int(binary_str[i:i+8], 2) for i in range(0, len(binary_str),
8)]
    # 将字节序列转换为字节对象
    decoded_bytes = bytes(bytes_list)
    return decoded_bytes
# 测试解码
encoded_str = "I7R8ITMCnzbCn5eFIC=6yliXfzN=I5NMnz0XIC==yzycysi70ci7y7iK"
decoded_bytes = base64_decode(encoded_str)
# 尝试将解码后的字节序列解码为字符串（假设是UTF-8编码）
try:
    decoded_str = decoded_bytes.decode('utf-8')
    print("解码后的字符串:", decoded_str)
except UnicodeDecodeError:
    print("解码后的数据可能不是有效的UTF-8格式")
```

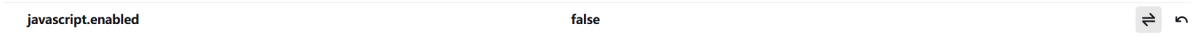


Bypass it

题目描述是个很好的提示



显然是因为前端检测而无法注册登录，用浏览器中的功能禁用javascript然后再注册即可

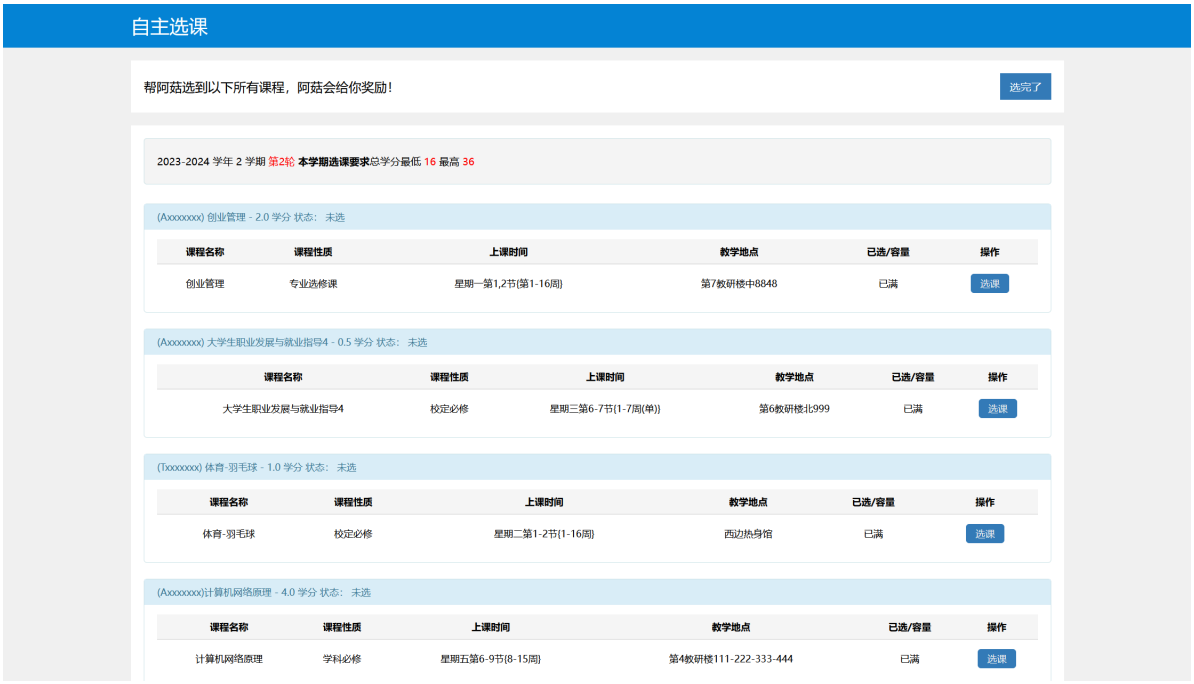


启用javascript登录，找到flag

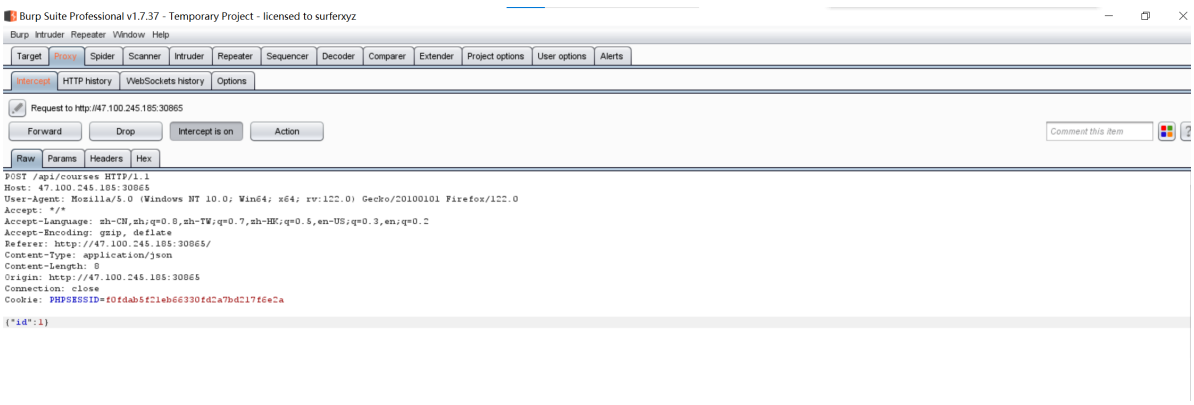
hgame{726bd746e16bf79734f1265c360696300bbdb7bc}

Select Courses

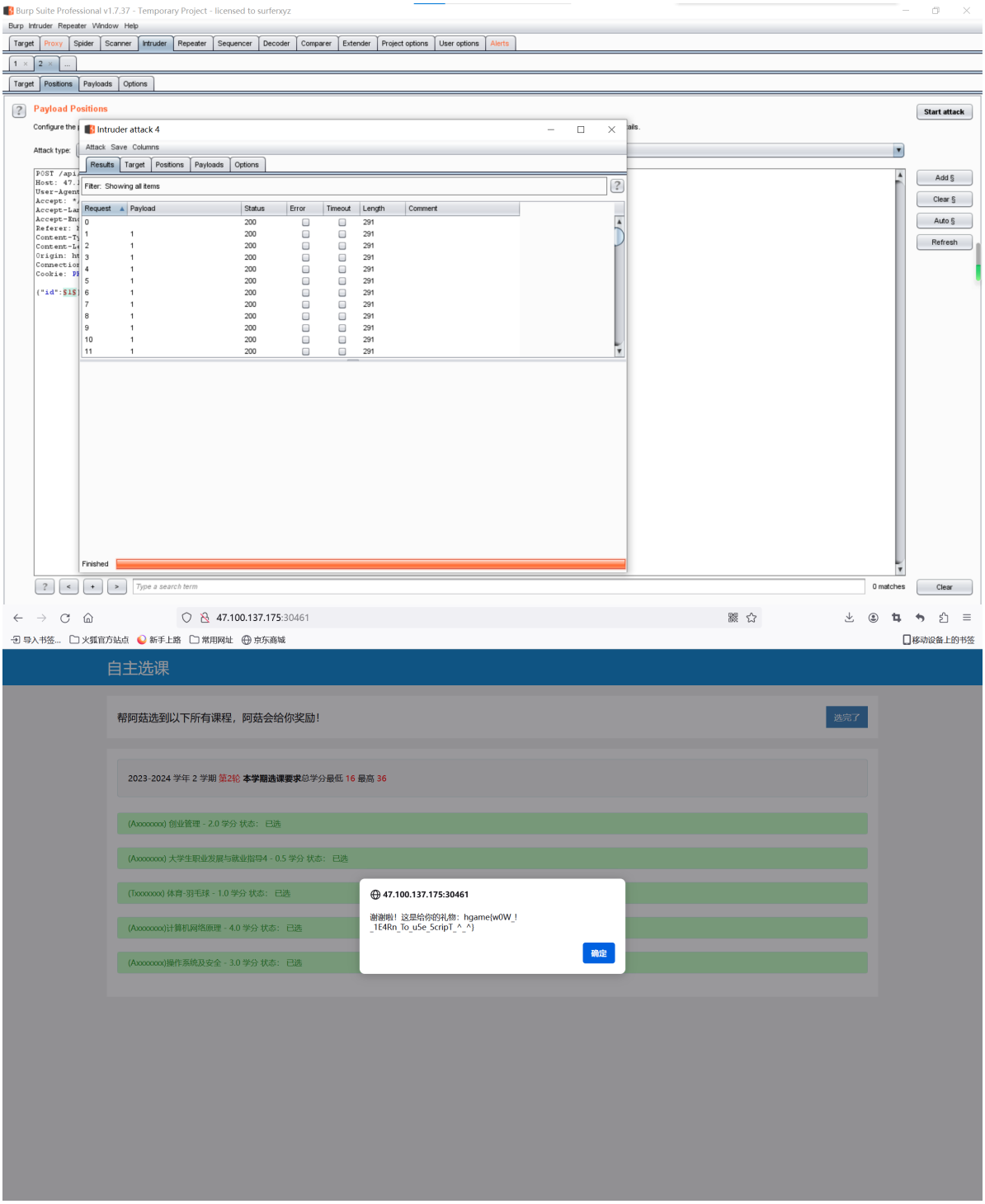
要求选到所有的课程，但是所有课程都是已满的状态



使用burpsuite抓包，看到选课实际上是对post传参id。



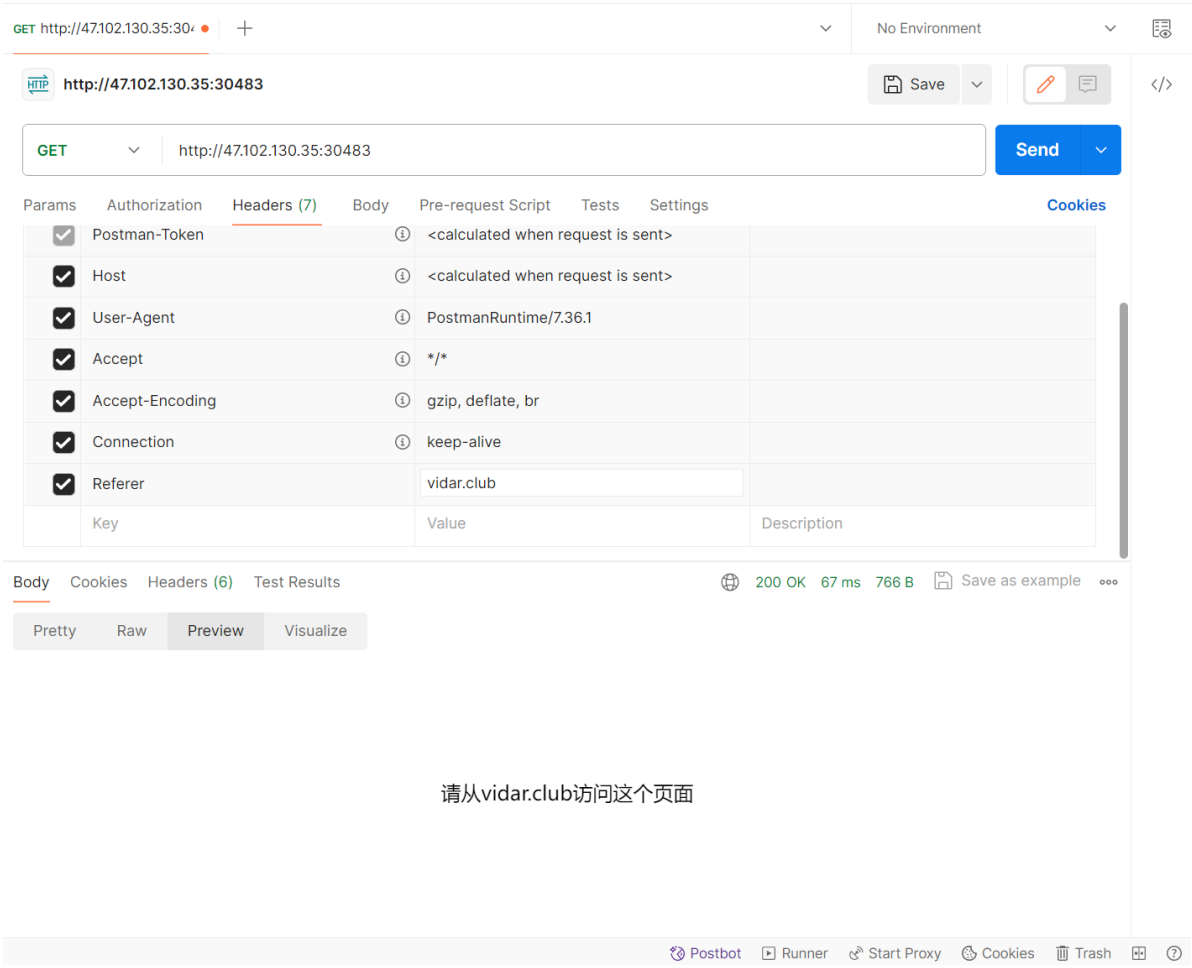
使用Intruder，卡系统自动退选的bug，不停地进行选课直到所有的课都选中为止。



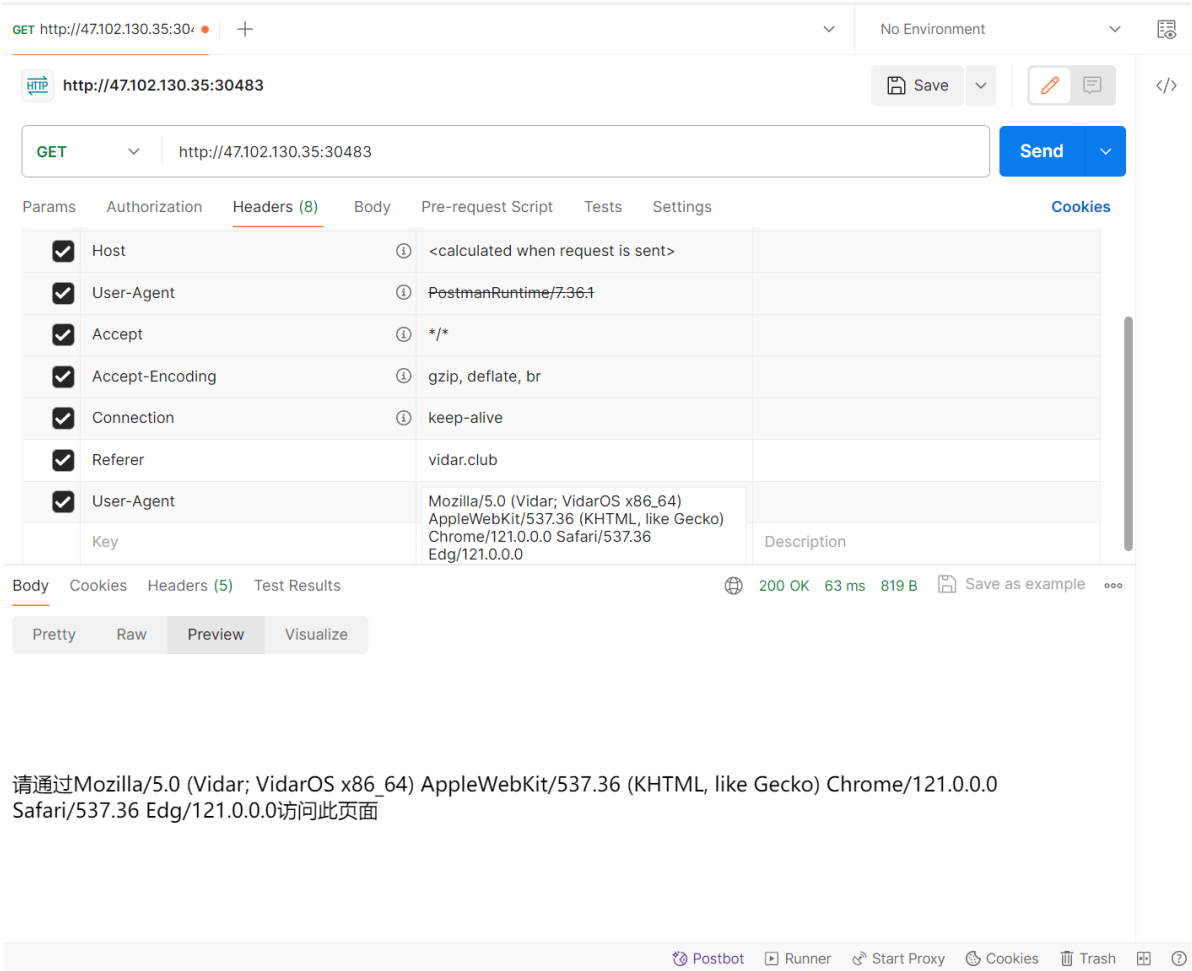
ezHTTP

一眼就是经典的各种http文件头的套娃题

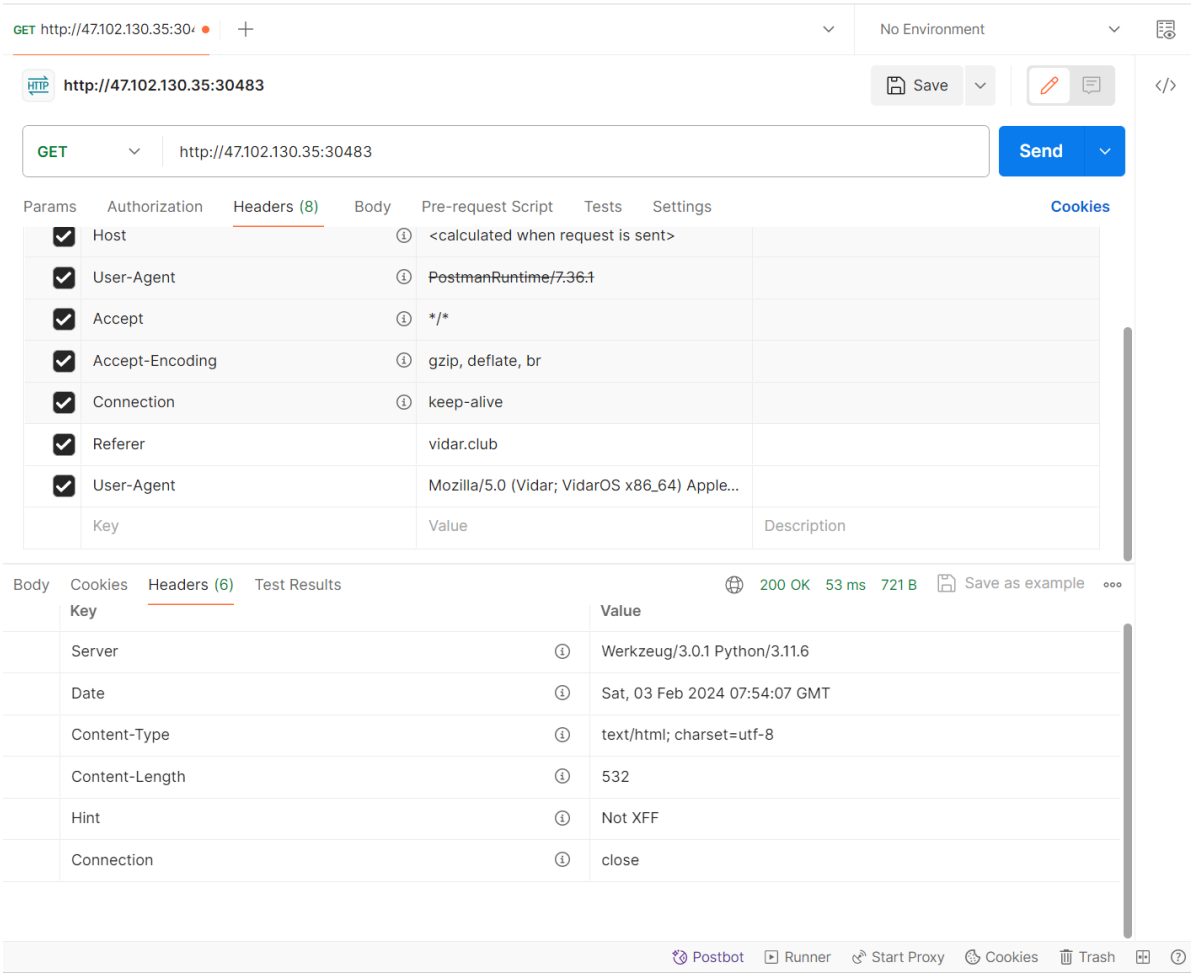
要求从vidar.club访问，referer头



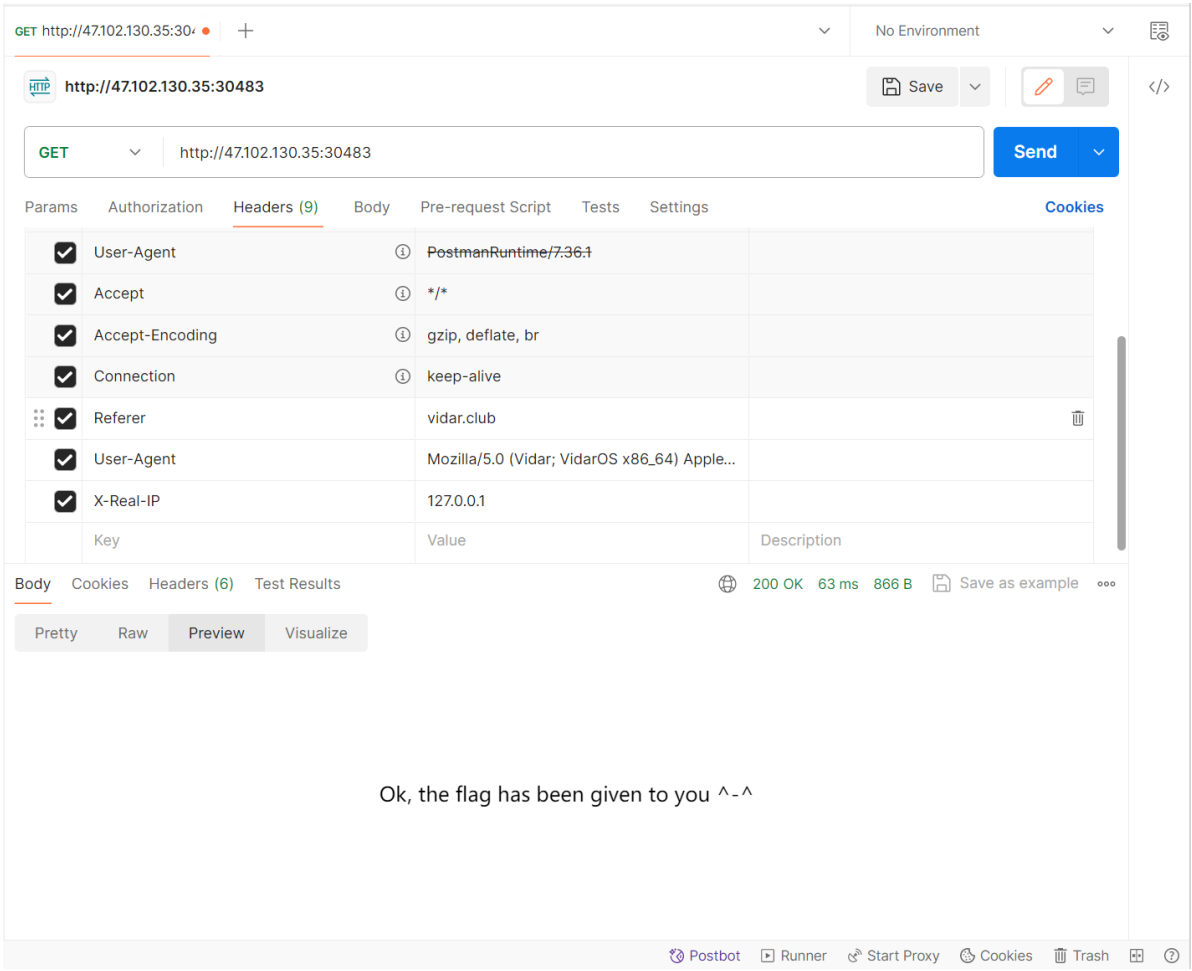
制定了浏览器，UA头



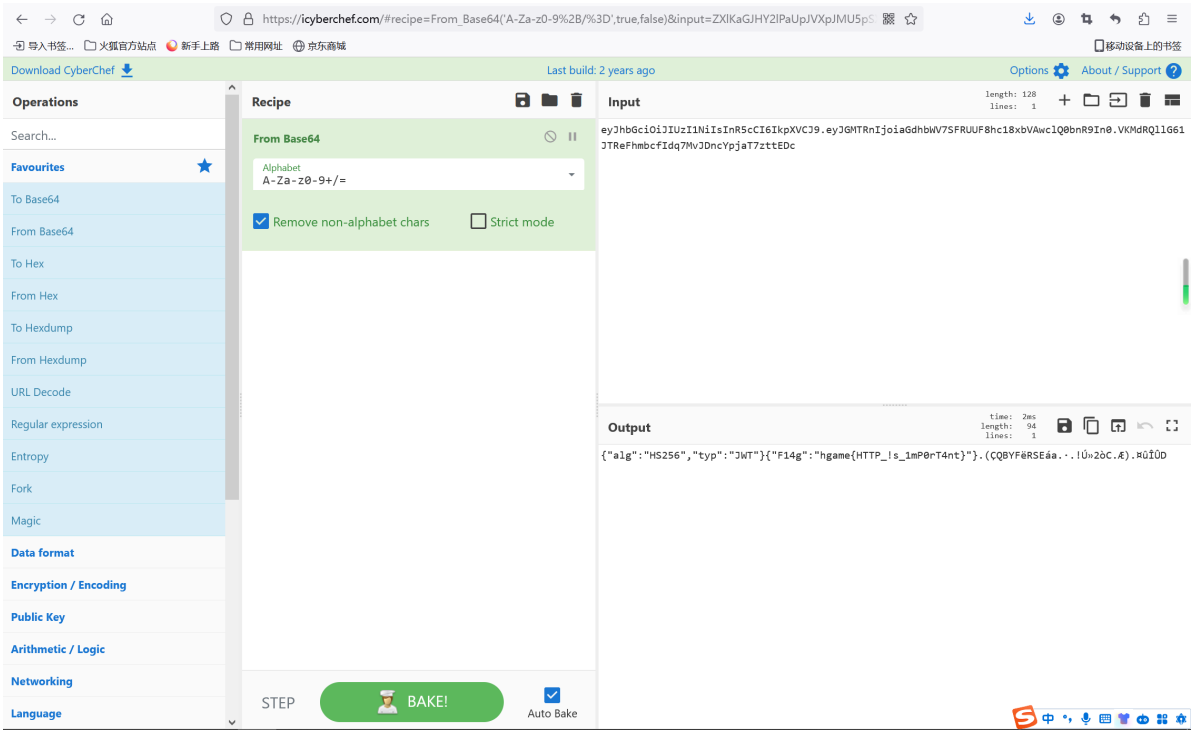
要求从本地访问，但是响应头里面要求不能使用XFF头



那么我们就使用X-Real-IP头，值为本地IP 127.0.0.1



在响应头的Authorization中发现了一个JWT，base64解码，得到flag



Reverse

ezASM

t题目是一串汇编语言

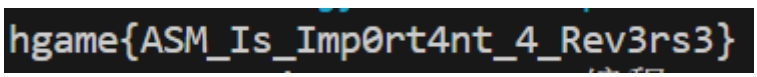
```
section .data
    c db 74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125, 107, 79, 82,
18, 80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17, 80, 81, 17, 95, 34
    flag db 33 dup(0)
    format db "plz input your flag: ", 0
    success db "Congratulations!", 0
    failure db "Sry, plz try again", 0

section .text
    global _start

_start:
    ; Print prompt
    mov eax, 4
    mov ebx, 1
    mov ecx, format
    mov edx, 20
    int 0x80

    ; Read user input
    mov eax, 3
    mov ebx, 0
    mov ecx, flag
    mov edx, 33
    int 0x80

    ; Check flag
    xor esi, esi
check_flag:
    mov al, byte [flag + esi]
```

```
    xor al, 0x22
    cmp al, byte [c + esi]
    jne failure_check

    inc esi
    cmp esi, 33
    jne check_flag

    ; Print success message
    mov eax, 4
    mov ebx, 1
    mov ecx, success
    mov edx, 14
    int 0x80

    ; Exit
    mov eax, 1
    xor ebx, ebx
    int 0x80

failure_check:
    ; Print failure message
    mov eax, 4
    mov ebx, 1
    mov ecx, failure
    mov edx, 18
    int 0x80

    ; Exit
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

根据其中的逻辑只要将c中的元素都与0x22异或再转ascii即可得到flag

exp:

```
c = [74, 69, 67, 79, 71, 89, 99, 113, 111, 125, 107, 81, 125, 107, 79, 82, 18,
80, 86, 22, 76, 86, 125, 22, 125, 112, 71, 84, 17, 80, 81, 17, 95, 34]
flag=''
for i in c:
    flag+=chr(i^0x22)
print(flag)
```



ezPYC

（修改前）

先用下面的指令pyinstxtractor将exe反编译为pyc文件

```
python3 pyinstxtractor.py ezPYC.exe
```

然后再用pycdc将pyc转py时出现了错误，于是用pycdas将其转为字节码

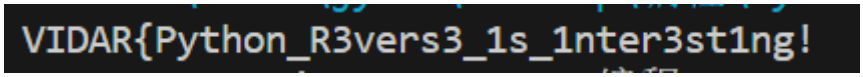
```
#先切到pycdc文件夹内
./pycdas ezPYC.pyc > ezPYC.txt
```

通过阅读字节码人为排除错误，再根据其中的逻辑（一个简单的异或）编写exp：

```
flag = [
    87,
    75,
    71,
    69,
    83,
    121,
    83,
    125,
    117,
    106,
    108,
    106,
    94,
    80,
    48,
    114,
    100,
    112,
    112,
    55,
    94,
    51,
    112,
    91,
    48,
    108,
    119,
    97,
    115,
    49,
    112,
    112,
    48,
    108,
    100,
    37,
    124,
    2]
c=[1,2,3,4]
s=''
for i in range(0, 36, 1):
    s+=chr(flag[i] ^ c[i % 4])
```

```
print(s)
```

得到flag：



（修改后）

~~（早知道晚点做了。。。）~~

先用下面的指令pyinstxtractor将exe反编译为pyc文件

```
python3 pyinstxtractor.py ezPYC.exe
```

然后用pycdc如下将其反编译为py文件

```
#先切到pycdc文件夹内
./pycdc ezPYC.pyc > ezPYC.py
```

下面是反编译出的源码

```
flag = [
    87,
    75,
    71,
    69,
    83,
    121,
    83,
    125,
    117,
    106,
    108,
    106,
    94,
    80,
    48,
    114,
    100,
    112,
    112,
    55,
    94,
    51,
    112,
    91,
    48,
    108,
    119,
    97,
    115,
    49,
    112,
    112,
    48,
    108,
```

```
    100,
    37,
    124,
    2]
c = [
    1,
    2,
    3,
    4]
input = input('plz input flag:')
for i in range(0, 36, 1):
    if ord(input[i]) ^ c[i % 4] != flag[i]:
        print('Sry, try again...')
        exit()
        continue
        print('Wow!You know a little of python reverse')
        return None
```

根据源码编写exp：

```
flag = [
    87,
    75,
    71,
    69,
    83,
    121,
    83,
    125,
    117,
    106,
    108,
    106,
    94,
    80,
    48,
    114,
    100,
    112,
    112,
    55,
    94,
    51,
    112,
    91,
    48,
    108,
    119,
    97,
    115,
    49,
    112,
    112,
    48,
    108,
    100,
    37,
```

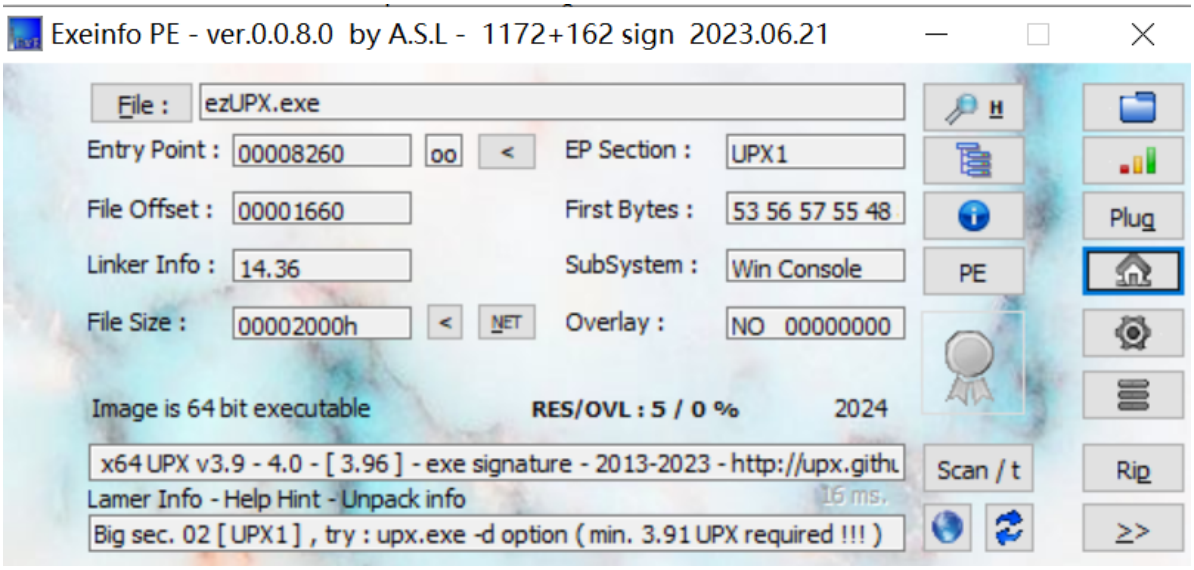
```
    124,
    2]
c=[1,2,3,4]
s=''
for i in range(0, 36, 1):
    s+=chr(flag[i] ^ c[i % 4])
print(s)
```

得到flag：

```
VIDAR{Python_R3vers3_1s_1nter3st1ng!
```

ezUPX

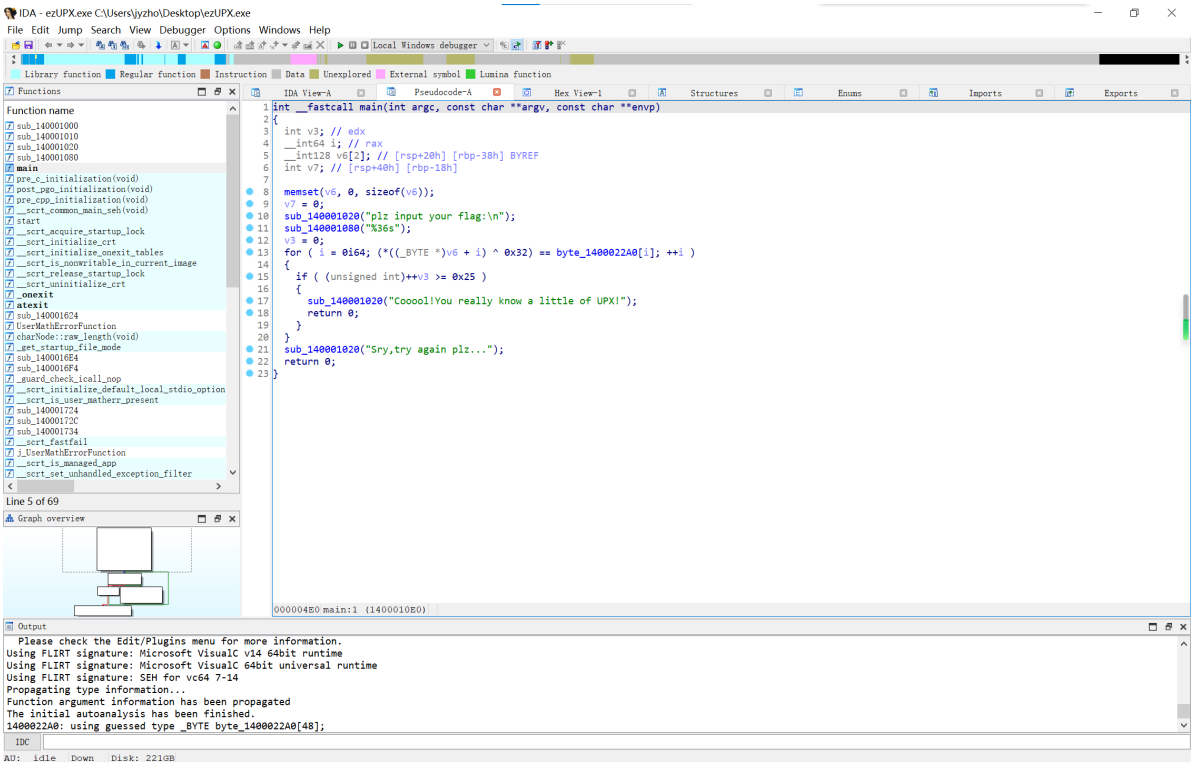
先用exeinfope查壳



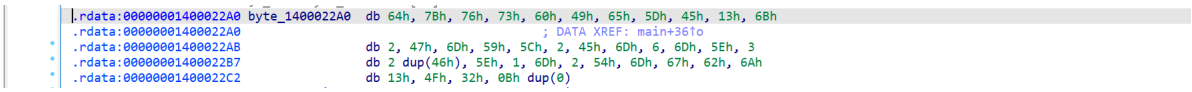
被加壳了，用upx脱壳

```
C:\Users\jyzho\Desktop\h4ck3r t0015\upx>upx -d ezUPX.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2020
UPX 3.96w       Markus Oberhumer, Laszlo Molnar & John Reiser   Jan 23rd 2020

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     10752 <-      8192   76.19%    win64/pe     ezUPX.exe

Unpacked 1 file.
```

扔进upx生成伪代码，又是一个异或

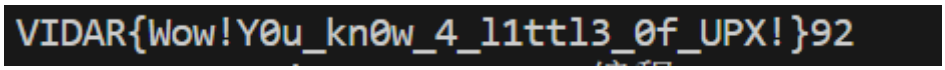


查看byte_1400022A0



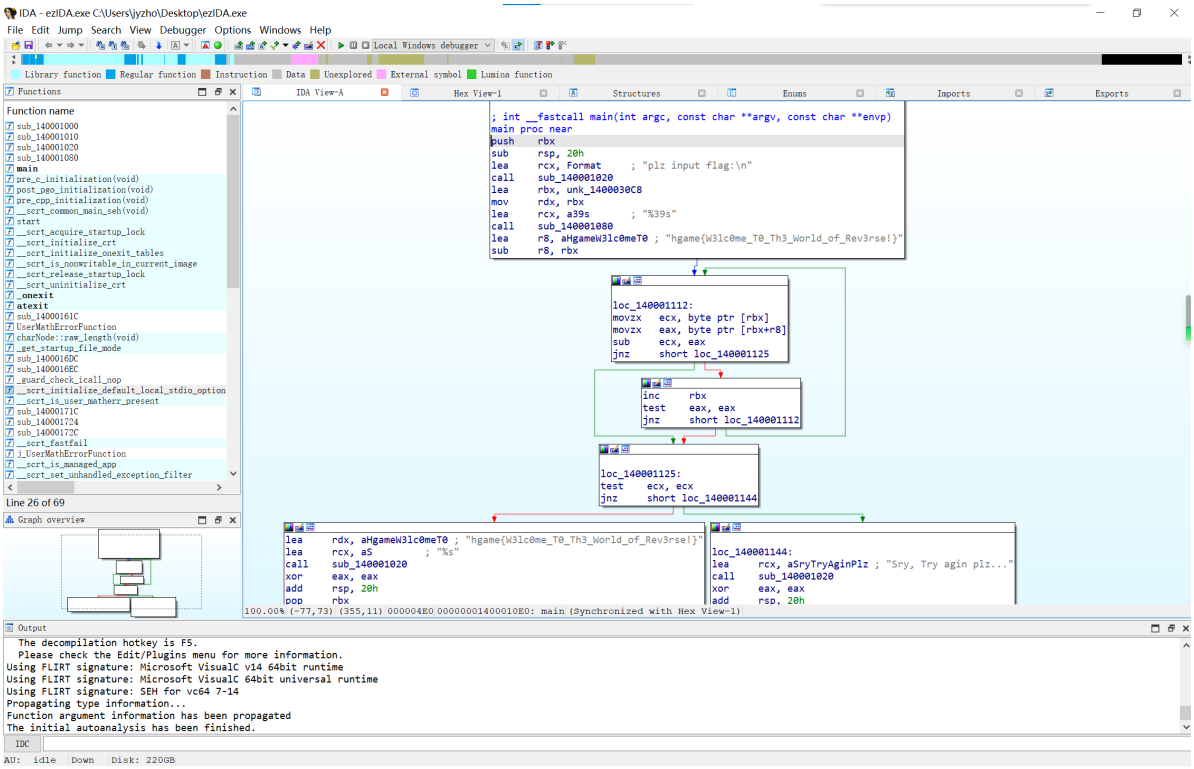
exp:

```
c =
[0x64,0x7B,0X76,0X73,0X60,0X49,0X65,0X5D,0X45,0X13,0X6B,0x2,0X47,0X6D,0X59,0X5C,
0X2,0X45,0X6D,0X6,0X6D,0X5E,0X3,0X46,0x46,0X5E,0X1,0X6D,0X2,0X54,0X6D,0X67,0X62,
0X6A,0X13,0X4F,0X32,0XB,0X0]
s=''
for i in c:
    s+=chr(i^0x32)
print(s)
```



ezIDA

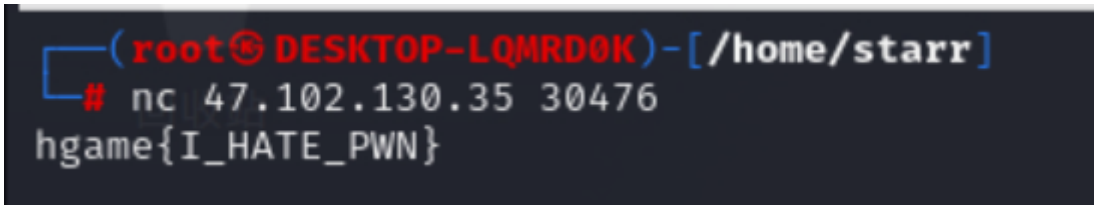
直接丢进IDA，即可看到flag



Pwn

EzSignIn

签个到



Crypto

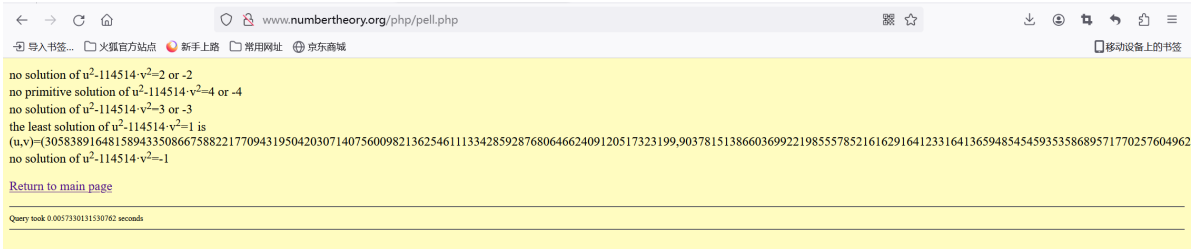
ezMath

题目

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
import random,string
from secret import flag,y,x
def pad(x):
    return x+b'\x00'*(16-len(x)%16)
def encrypt(KEY):
    cipher= AES.new(KEY,AES.MODE_ECB)
    encrypted =cipher.encrypt(flag)
    return encrypted
D = 114514
assert x**2 - D * y**2 == 1
flag=pad(flag)
key=pad(long_to_bytes(y))[:16]
enc=encrypt(key)
print(f'enc={enc}')
```

```
#enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17
g\x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\
xdec\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"
```

断言函数后面是个pell方程，用在线网站http://www.numbertheory.org/php/pell.html解一下



AES解密即可

```
from Crypto.Util.number import *
from Crypto.Cipher import AES
import random,string
def decrypt(KEY, encrypted):
    cipher = AES.new(KEY, AES.MODE_ECB)
    decrypted = cipher.decrypt(encrypted)
    return decrypted

def pad(x):
    return x+b'\x00'*(16-len(x)%16)

# 使用相同的KEY和encrypted进行解密
enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g
\x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\x
dec\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"
y=90378151386603699221985557852161629164123316413659485454593535868957177025760496
2625533527779108680
KEY=pad(long_to_bytes(y))[:16]
flag = decrypt(KEY, enc)
print(flag)
```

```
b'hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

ezRSA

题目

```
from Crypto.Util.number import *
from secret import flag
m=bytes_to_long(flag)
p=getPrime(1024)
q=getPrime(1024)
n=p*q
phi=(p-1)*(q-1)
e=0x10001
c=pow(m,e,n)
leak1=pow(p,q,n)
leak2=pow(q,p,n)

print(f'leak1={leak1}')
print(f'leak2={leak2}')
print(f'c={c}')
```

```
"""
leak1=14912717007361127196818257675129033155901844180572531042609541283758922767
07575407439298658536503998391028384315072007447249396594632001580124696769799876
96419050900842798225665861812331113632892438742724202916416060266581590169063867
68829928898573410412763223217565735269789838344132347745065817972772890866 9
leak2=11612299271467091538130991696749043648902000117288064416717991546702179489
29279772720805966417855691191342590375223883351980431522061502591034855745588164
24740204736215551933482583941959994625356581201054534529395781744338631021423703
17114645666343295584359854812259330878224522079201871650853849740257670946 1
c=10529481867532520034258056773864074017027019578041866245400647840230251661652 9
99709715919620810933437191661180003295923273655675729588558899592524235622728816
06550191807612081223658034499114098099153234799125270528863301491347997061005684
55435235913241775670619489225522752354866155149139321254365439916426070286897626
93617305246716492783116813070355512606971626645594961850567586340389705821314842
09646563188681228128984313225813180977379777049358789182212570606252509790830 99
42631320200941536462967935229756321919124639198989883492822849729199327619526033
79733234575351624039162440021940592552768579639977713099971
"""
```

利用费马小定理a^(p-1)%p=1（可推出a^p%p=a%p）

题中给出了

```
p^q%n=leak1  (1)

q^p%n=leak2  (2)
```

对（1）式分析，

p^q=leak1+k*n

两边模q，

p^q%q=leak1%q

根据费马小定理得，

p%q=leak1%q

同理，

q%p=leak2%p

exp:

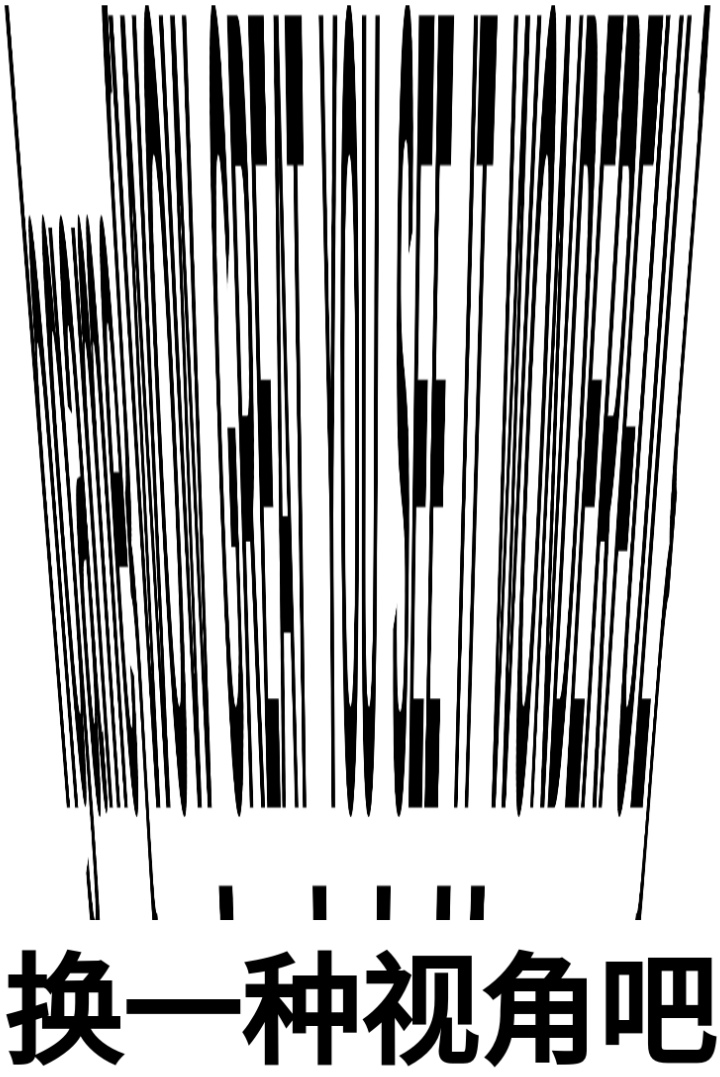
```
import gmpy2
from Crypto.Util.number import *
c =
10529481867532520034258056773864074017027019578041866245400647840230251661652999
70971591962081093343719166118000329592327365567572958855889959252423562272881606
55019180761208122365803449911409809915323479912527052886330149134799706100568455
43523591324177567061948922552275235486615514913932125436543991642607028689762693
61730524671649278311681307035551260697162664559496185056758634038970582131484209
64656318868122812898431322581318097737977770493587891822125706062525097908309942
63132020094153646296793522975632191912463919898988349282284972919932761952603379
73323457535162403916244002194059255276857963997771309997 1
E =0x10001
leak1=14912717007361127196818257675129033155901844180572531042609541283758922767
07575407439298658536503998391028384315072007447249396594632001580124696769799876
96419050900842798225665861812331113632892438742724202916416060266581590169063867
68829928898573410412763223217565735269789838344132347745065817972772890866 9
leak2=11612299271467091538130991696749043648902000117288064416717991546702179489
29279772720805966417855691191342590375223883351980431522061502591034855745588164
24740204736215551933482583941959994625356581201054534529395781744338631021423703
17114645666343295584359854812259330878224522079201871650853849740257670946 1
phi=(leak1-1)*(leak2-1)
n=leak1*leak2
d = gmpy2.invert(E,phi)
m = pow(c,d,n)
m=long_to_bytes(m)
print(m)
```

```
b'hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}'
```

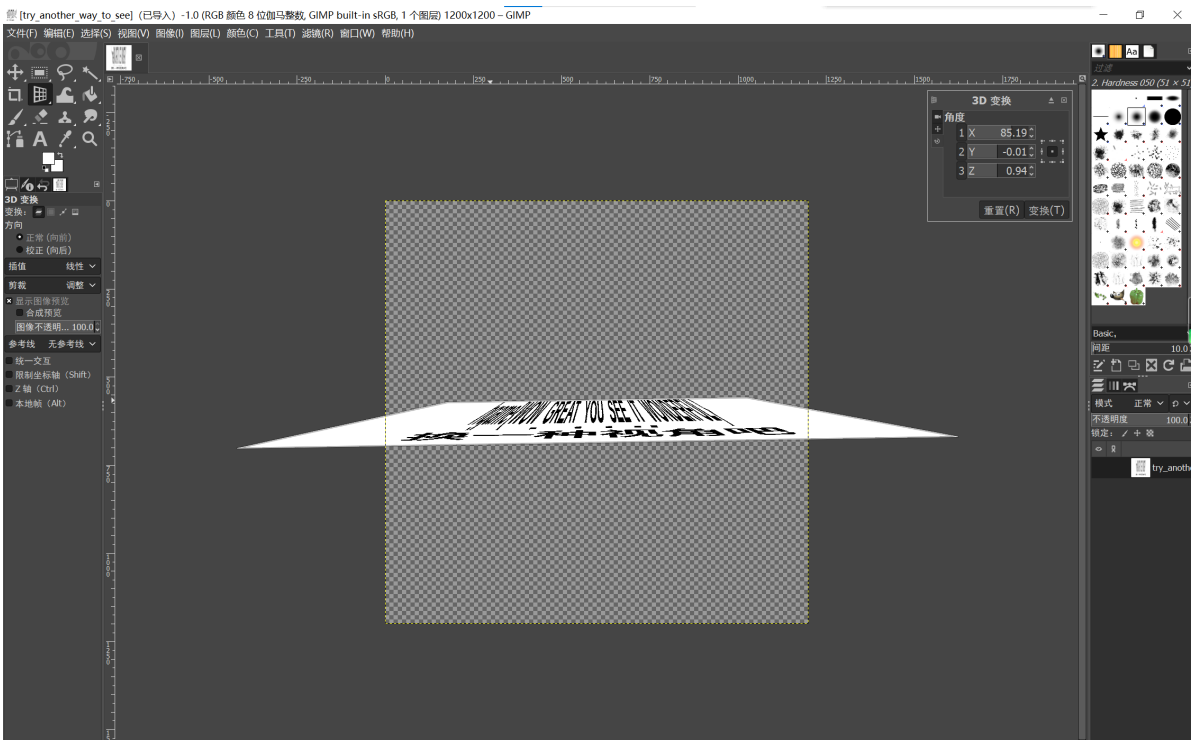
Misc

SignIn

题目

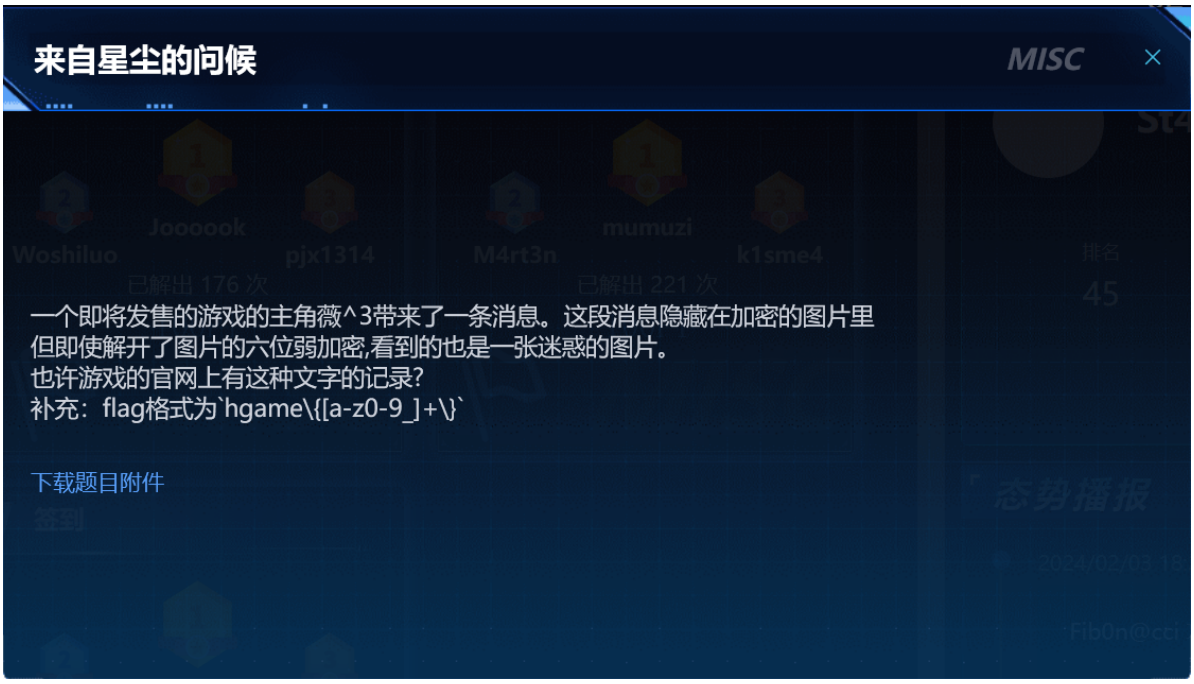


提示换一种视角，用GIMP~~（PS太贵了用不起）~~旋转一下，看到flag

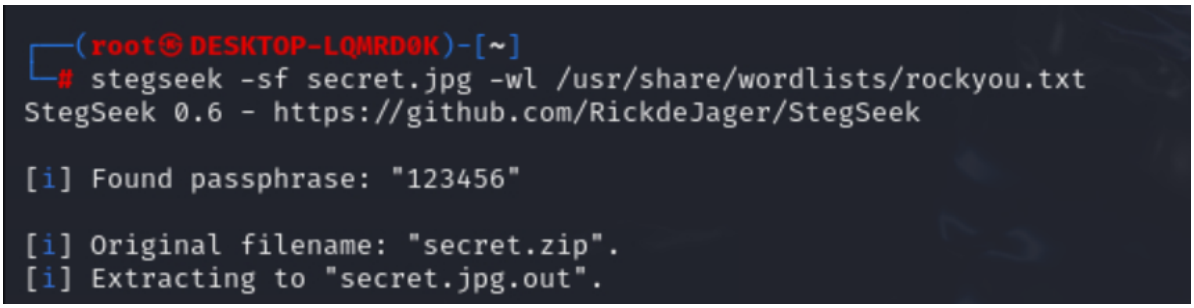


hgame{WOW_GREAT_YOU_SEE_IT_WONDERFUL}

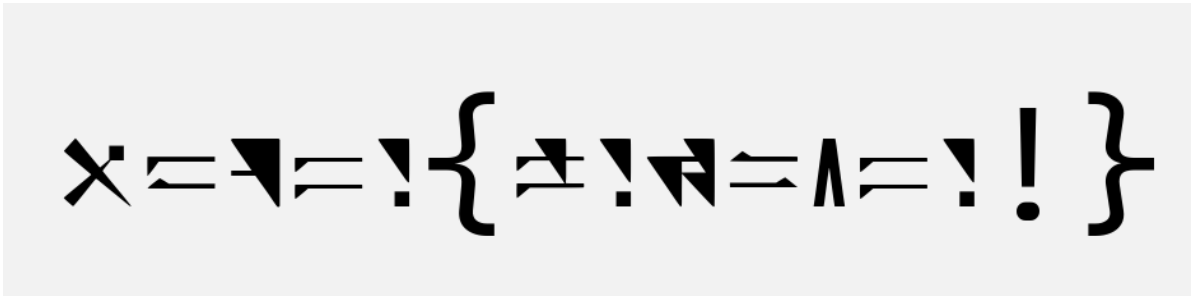
来自星尘的问候



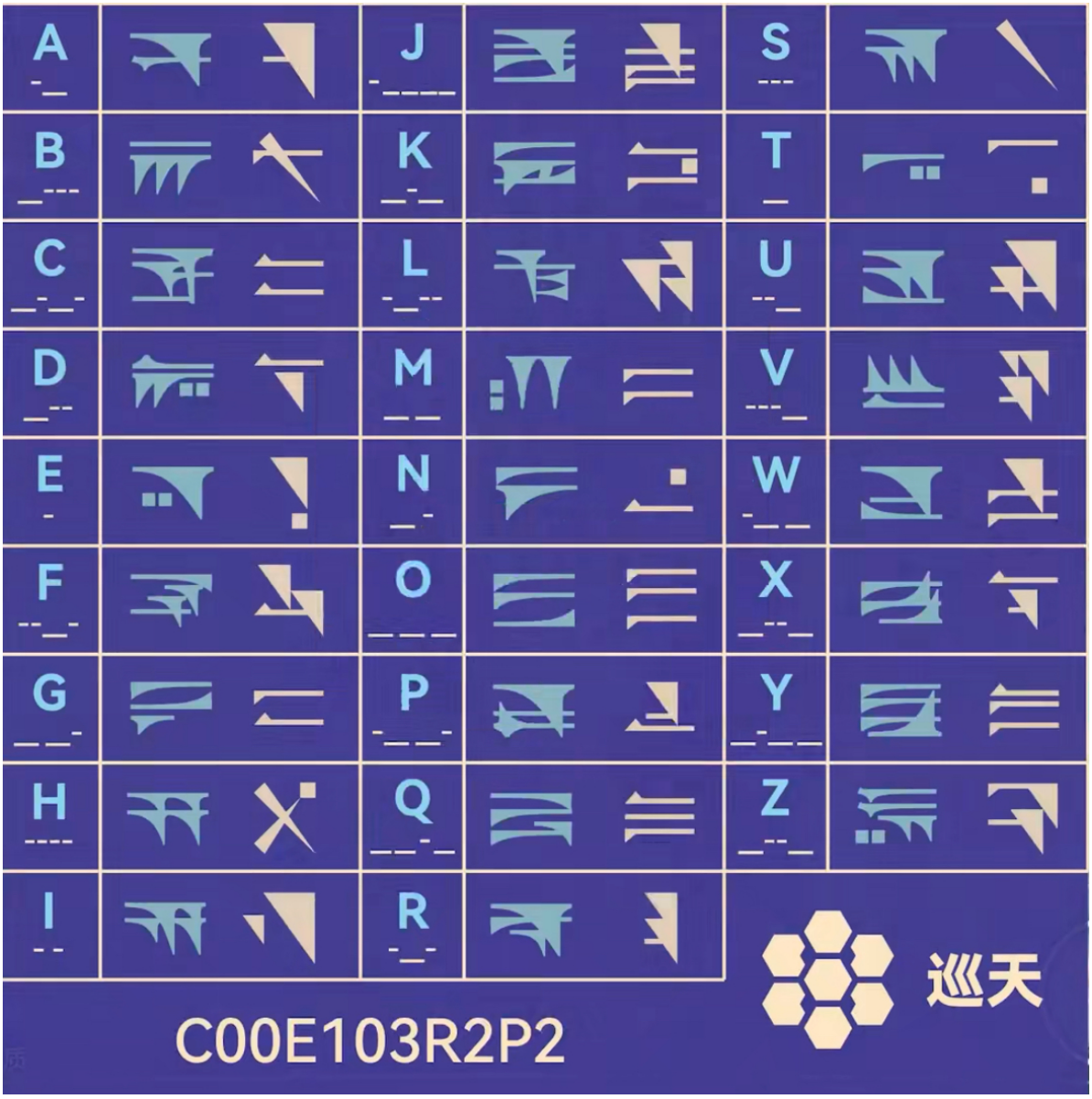
根据题目描述，推测下载得到的图片存在隐写，密码为六位弱口令，直接stegseek爆破



提取出一张图片，上面有一些奇奇怪怪的字符。



。根据题目描述看了半天不知道是什么游戏的官网。问了一个二次元朋友才知道，这整道题指的就是一个叫《来自星尘》的游戏。。。然后就在游戏相关的地方找到了玩家破解出的对照表。



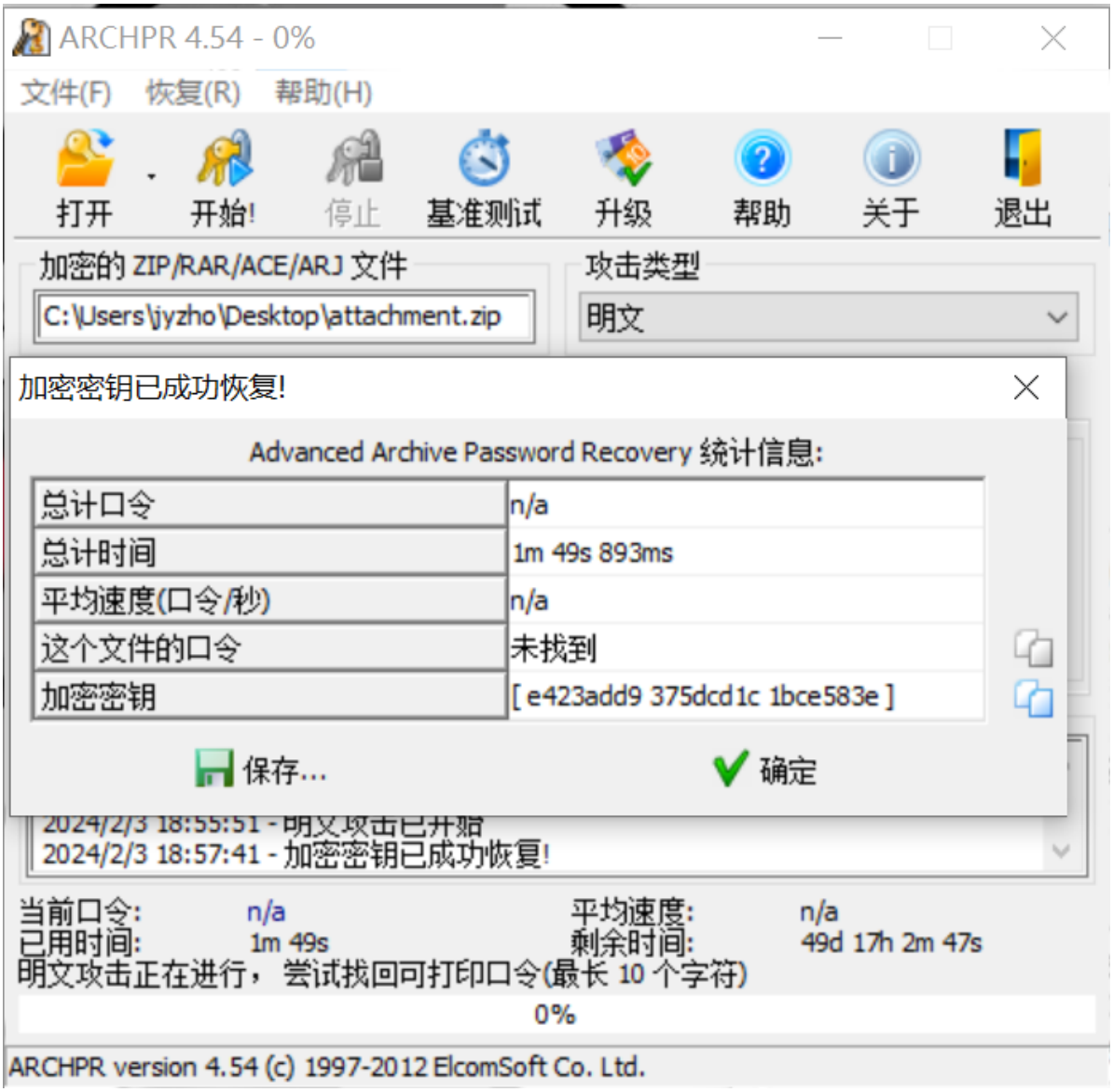
对照+猜测得到flag

hgame{welc0me!}

simple_attack

给了一个压缩包和其中的一个文件，一眼明文攻击。然而一开始我用的是Winrar进行压缩，由于压缩加密方法的不同导致明文攻击失败。后来换成使用Bandizip就成功了。

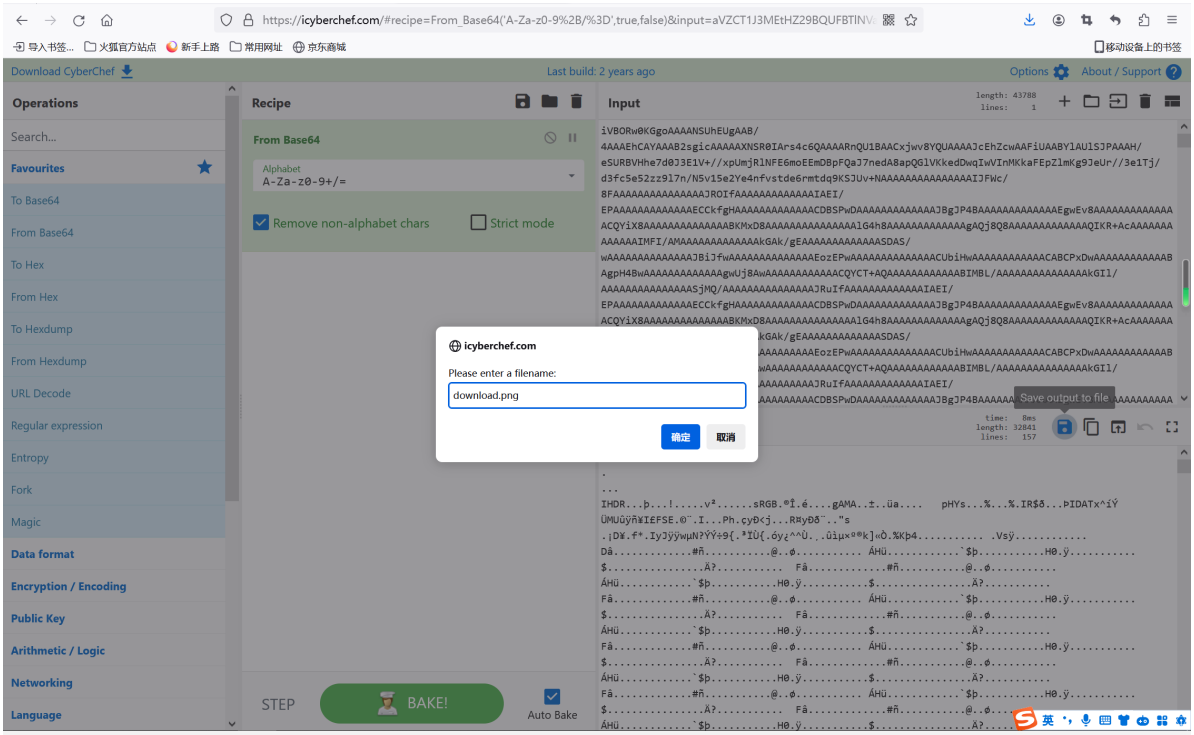
在ARCHPR运行了一会儿以后点停止，保存后即可看到之前被加密的文档



查看photo.txt，一眼base64


photo - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

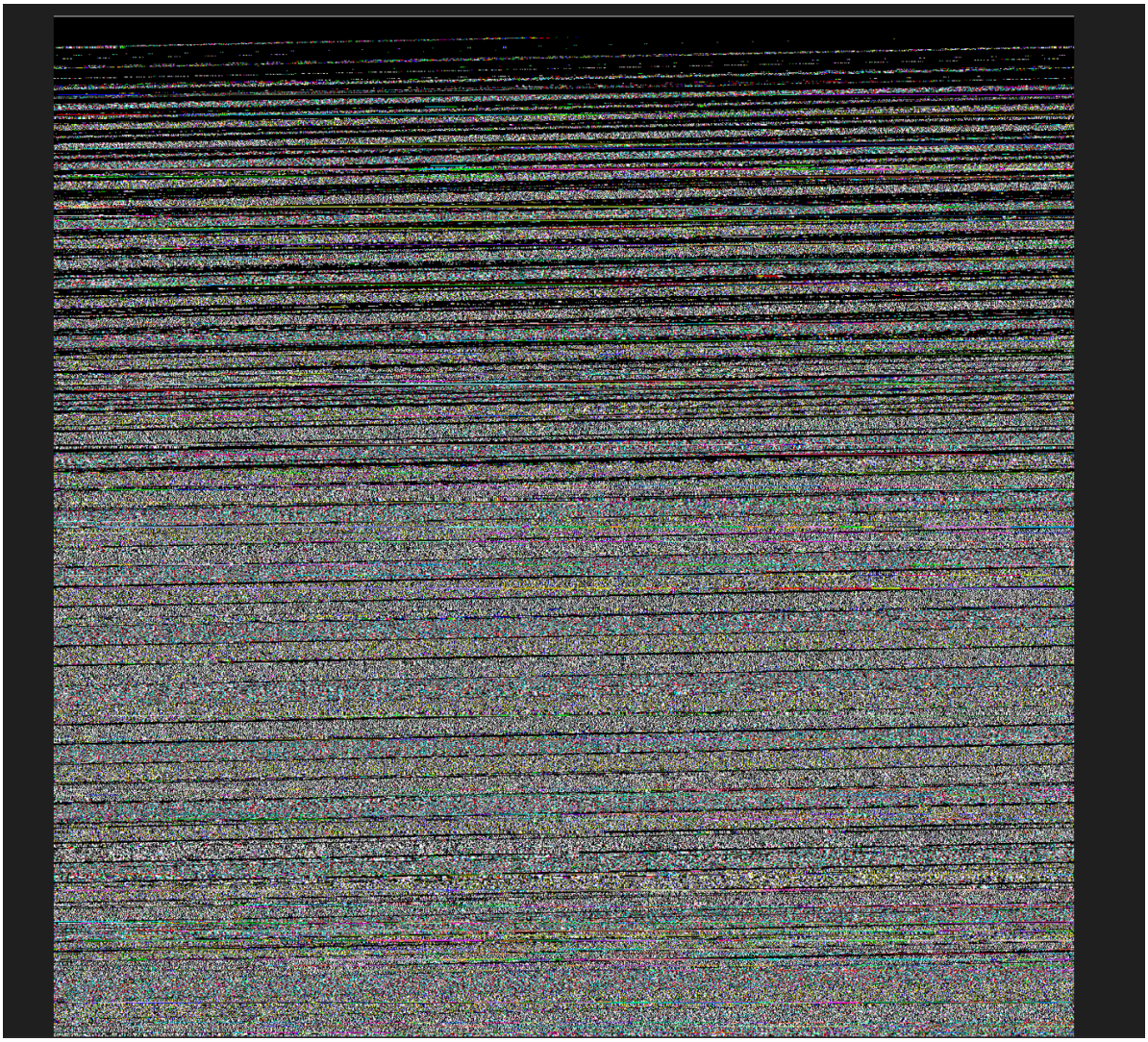
AAkGAk/gEAAAAAAAAAAAAASDAS/wAAAAAAAAAAAAAAJBiJfwAAAAAAAAAAAAAAEozEPwAAAAAAAAAAAAAACUbiHwAAAAAAAAAAAC/
AAAAAAAAAACUbiHwAAAAAAAAAAACABCPxDwAAAAAAAAAAAABAgpH4BwAAAAAAAAAAAAAgwUj8AwAAAAAAAAAAACQYCT+AQAA
nVq5c9YhtAvJD4BwAAKHJffvmlDSJxrigAAAAAxleS/rfeequ57bbb3CsAEB3t4n/ppZfMOeecY+bMmeNe/TcVAqgA4l477rDjGID4IPEPAABQpLRgm
r6q5XgpZdeypnvRUI7Rd59911blXvyySezyx8AAAAAYk67K0eMGOGuMttrr73MDTfcYLbcckv3CgDkx2abbWaaNWvmrvxsv/327mdIKh0BcO6555qr
JKBRwbbLCBjesqvqu1g+K96Yo5yvPyyy/TdTSBiOIDAFCk1B51xIgR7iozVfT279/fdOrUyZ7rhcLSrja1fbzuuutMkyZNTJcuXTiiAQAAWL/99ptt33/yySc
KBxD8AAEVm8eLFZuHChe4qvSBnOSEe6tevb/r27WtefvIle0QDZz4CAAAAAAAAyEatWrXcz1BMiBgDAFCi1NJprbXWcleIq6pVq5o777zTDB8+3HTc
YHfZ40Nmgs/eqrr/41D9GYWadOHft50pi/66672s9SFOPn0qVL/54z6fOvr1fXuu9XVXbeVErzy0zKey5U9D6m5kwaJ1PPBP03zs9P/dvOO+88M2zYM
nkETvuIC12TEdziRtvvLFkYr5AEKUULyiLxH9xlvFfAZ/EvwYGVdY9/PDDpn///u7V8OhmUkvgI488MrETPr3/WrQ/+uijNkiTDwpgn3766XYRnGnSpwm
J6dmpXt/6N+ZyD6s/W81nPaf2dYXwGShnPdGg80vfi9ttvt2sg3ccauwuRCBTN5TWma4zUer/YxkeER+tKJasVLFXxtDZsFCLpL6uOX5ozhLHujIKOMH
sf6rmoTSJ6/7Q2J4FVGPpMKA6iOaovJf3VYlmdswDRfax4lRLsihvFKXYpqcK9JHazLI9ixZobab1G3ARBqUvLJpts4q4gxAtQSkj8e/rwww/tWThRVnSLk
kFfno2xHkXjBbzSsAMGDCAxfwqeKYjqTSuJ604CqVHzx6tO+PyLFe8SMcGxm2TSCnQZ0CfBd9ER9WqVe3Ghf3335/EESx1wFUCXc/ApFGRoXb+J61
NjcrYJAt2/HTt2tOPzDjvskHZ81t+nAJkCukoi5nIEh6psu3TpYtZdd133SmEouKOK9kwBtkKMbykvvfSSueCCC7L6HuqeUeBJ54E2btzYbLzxxhmDUPc
WoKKyQgZO4TdpUaHT++ed7Bws0rqjqVc/XMMYYPbdmzZplA6jlJUTzuVgplFJN/B933HFpCxE11mrBp8B3NtXvCuy98MILdtzOJpGuv1/3WqpQLm
3d2bTnzHdhaTZzdBWN6r3JVHyqRM/ll18eaqcaPfv1/dduimwKJiXfuy/jmvjnmY7yBJnn6/3Wmls7cPR5CJPWkPocaIz0TVBqrql5jgLthRDXe9tH0Dlqy
l/ff6VdC90e/g4I14AkPivkIKtukHLoyoqtXnUQiTsCYwCUeo0oArQbCYuqlpWZVE+KWGZKVFZnlTyUpXfYQXuyqOJj1qm6+/K5j2ME1W6Dx061D7w
Fk6o8//uiuUEqUfFdxinZ1FSLBlaJxS2OW5jo+9OzUkY35pPFeXQp96b5X8l2bYPKx7s1E3z+dkb3NNtu4V5JFzyMdAZHNe6/uDPo8kfRHio6L0G79dI
mnYGnmzJl5e5YrVqROHUHPI07R168gfFJ32kdNyYDHH3/cJvB96WiDfB/DieTRRjLfzV66j7VDWc/PqClmrzWBb3czFTsE3WGdT+XtVFayMMqNYSiN
PfONFnW0dY+MYzdbxB3BJaWlf4jvNIno8++sj07Nkz8HiuPJUKbhjHgyFekJkKn4N0X9WGkHxsAkR+EGEPQDdkVK1AxDchl4+d1xpgfM940o7lBg0
A6uzcs+ty89dZbgXctKMmmIok4FBf57pzVYjLsoolVLV26NHBLbnXEiEvi/+OPP/betQKkKOikBVRUhZpqC51N4EtnUNarV89dFZZvskC7WX/++Wd3
jmloQCHiXPfY5w0nrHt6NGPmidw3EoxUtx1F69epmXXnrJvZKe1r5nnXVW3uN3xYR4gT/lUYlUI+hYlAkTJrD+TQAS/2koiKwd4nGgZKBPAkWLsLBuC
h2jfNMB4DS4NsZMmnJwqgp8Xj77bcHXt+laJ2npBItoRGE5p4+8zatG5RITxLNyX029mlX7U8//eSuCk/3bpy76CJ76uBy991321bpQWjHulr8c1yLP+
N20EEHmSOPPNK9kp6S/1o/af6hrw/xQuK/AqoI9N25lW9RnOfis4hM0gSlWNsmaeevEvVK7o8ePdo8/PDDNrnfrl07+/1RckTV9sWQ3C82+v7Eodic
NPWzWtYAXHAJQ2FWoSTEYcldp8hGc68kU7pNXtSAkxFTjXq1fPJlFvvvlmkvuIDc1HNQY+/fTT5vjjj7dBT31eVfyaxOR+Or7JQt/dtqVGnx19Prp27eq
UdroB0o04XaCxljEoOEv+okHZb+FRyIzeadCtJpfMAlfgn4Q8gW+oQok4hav3YvXt3W1AUJzpfT1+X2lPmq13UM888Yw444ADTqVMnM2bMGPtM/
hAlgLx7dq1c78Svv79+9s2ak899RTt/wEAiIiK8C655BKvs60zSe3cV1JMyX21Wn3//fdtcl/zIM0ztD5SS9W9997bJvcpckY6CnyqKERr5zCL9BXsP+yww/
Dn/qxKi0IFurTIa926tdlggw3crwDF79tvvzXLly93V5DKlSvbisq77rorcPJfSQqdFRp3mjTWqlXL7oaaNGmSGTJkiFeHgyB0dIL+fM79BxCmFStWeBV3qh
```

用cyberchef转换并保存为png
```



hgame{s1mple_attack_for_zip}

希儿希儿希尔

打开发现图片长这样，肯定是要修复宽和高

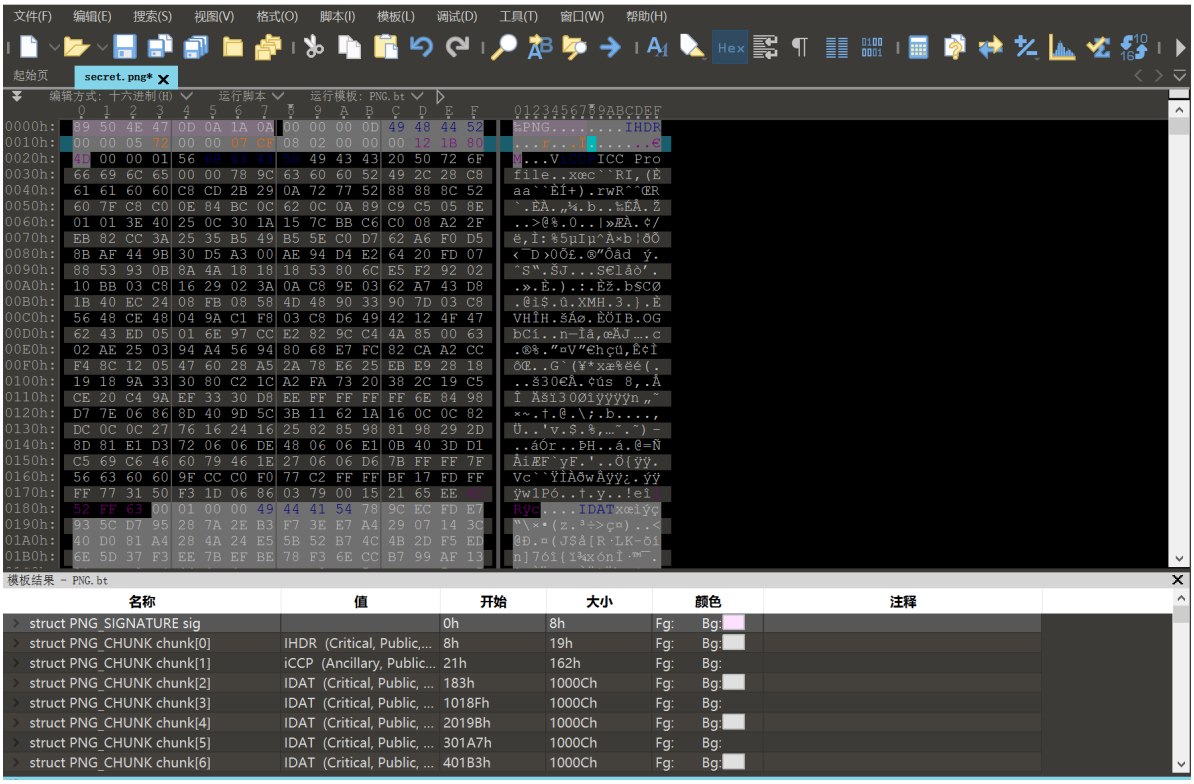


脚本跑一下，算出宽和高

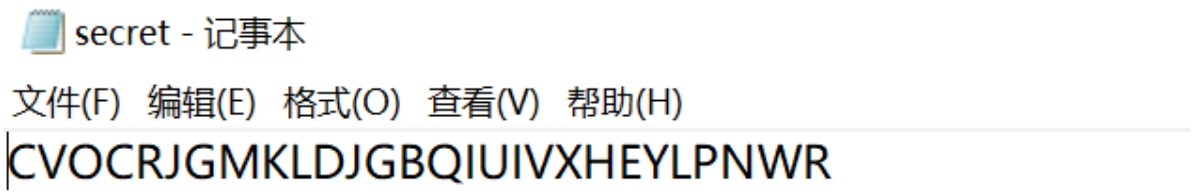
```
import binascii
import struct
crcbp = open("secret.png", "rb").read()     #填入图片名
crc32frombp = int(crcbp[29:33].hex(),16)      #读取图片中的CRC校验值
print(crc32frombp)

for i in range(4000):                           #宽度1-4000进行枚举
    for j in range(4000):                       #高度1-4000进行枚举
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        # print(crc32)
        if(crc32 == crc32frombp):
            print(i, j)
            print('hex:', hex(i), hex(j))
            exit(0)
```

```
303792205
1394 1999
hex: 0x572 0x7cf
```

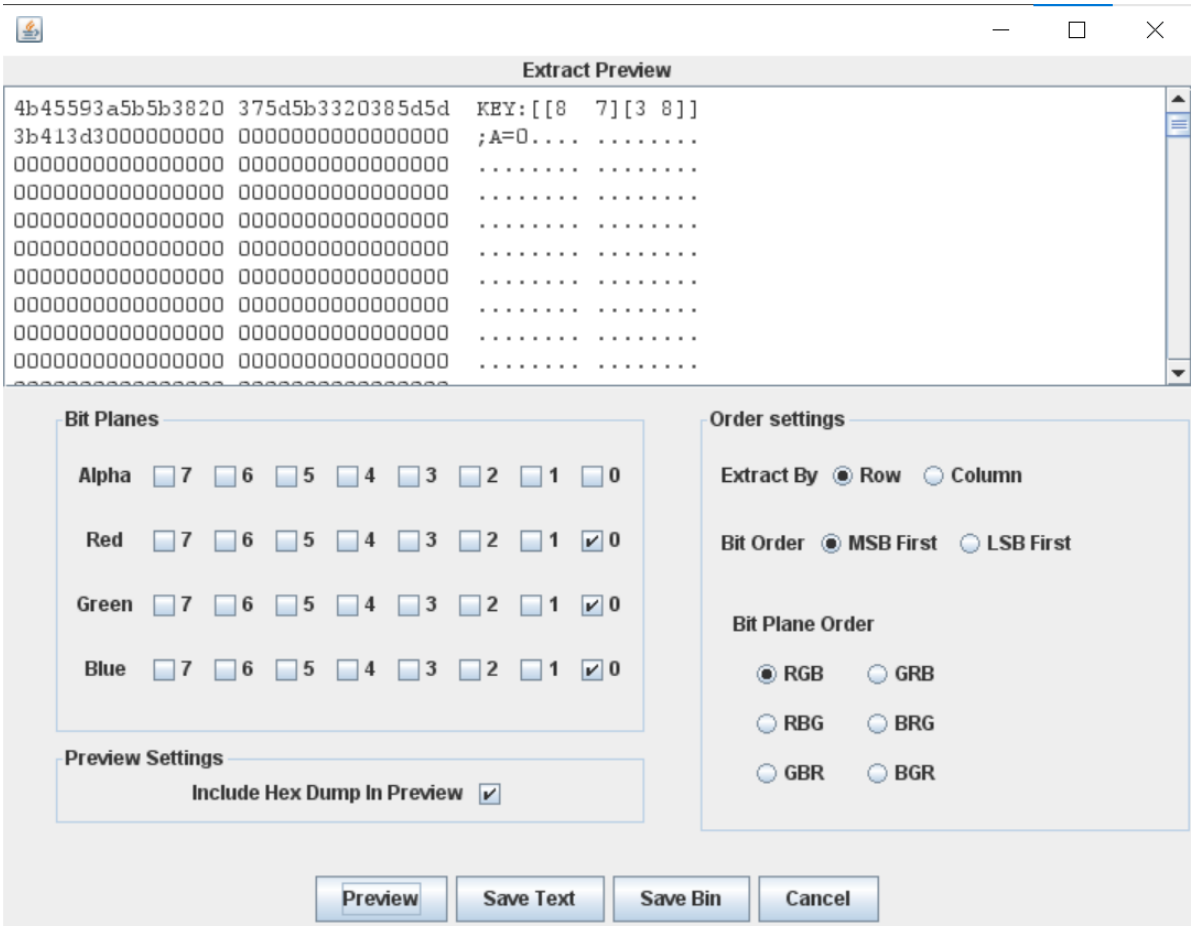
用010Editor修改一下



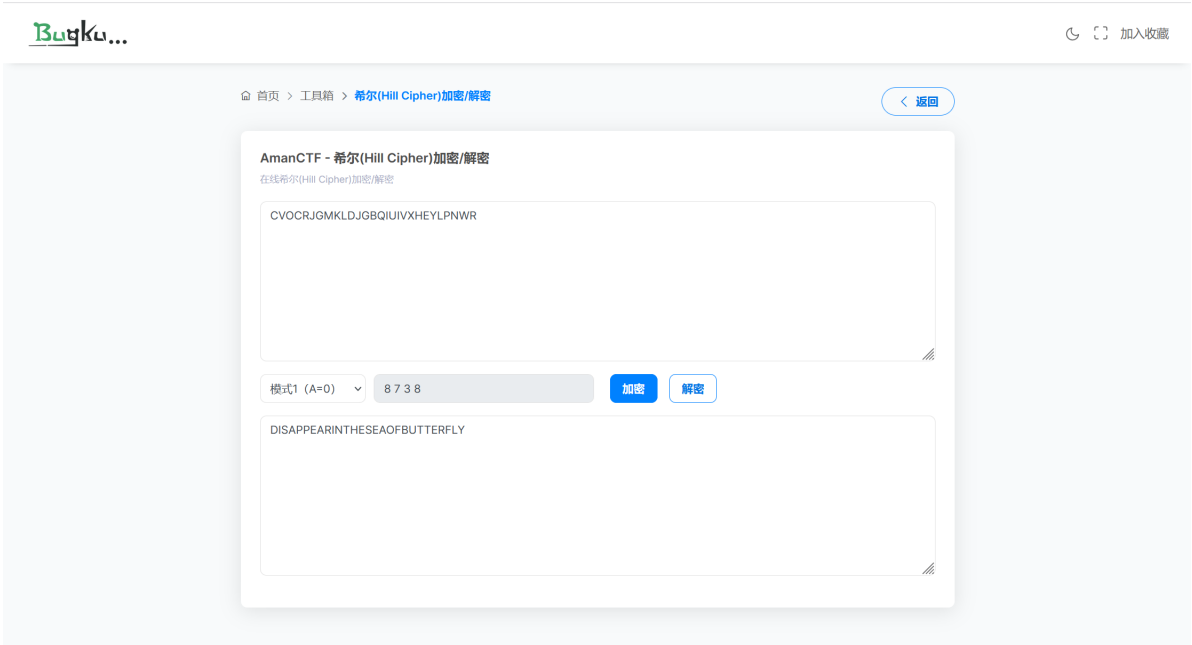
binwalk一下，分离出一个secret.txt



根据题目得知，这是一串希尔加密过的文字，接下来还得寻找密钥。对之前修复的图片分析，发现存在LSB隐写。



解密得到flag



签到

签个到

凌武科技

19:48

你好，欢迎关注凌武科技！

HGAME2024

hgame{
welc0me_t0_HGAME_2024}

乘风破浪 奋勇前行 | 2023凌武科技年度总结