Web 第一题

> 蛛蛛...嘿嘿♥我的蛛蛛[已完成]

描述

蛛蛛...嘿嘿...我的蛛蛛...我的蛛蛛正在满地找头？？？

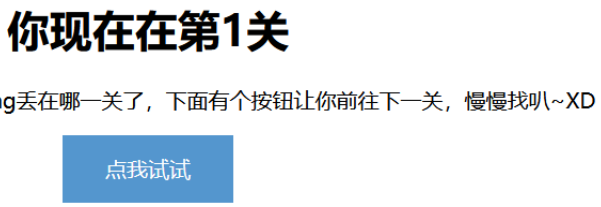
题目地址 https://hgame-spider.vidar.club/3766e8a01c

基准分数 150

当前分数 150

完成人数 393

打开链接



发现点我试试会跳到一个新网页 于是使用f12 审查元素

鼠标移到F12中的链接,在网页中会显示对应的位置,这应该就是前往下一关的按钮

一个一个点,盲猜需要点100次,果然如此,到第100关后显示

点击点我试试后发现没什么反应,但按F12查看,发现请求与返回中有flag字,于是得到flag

Web 第二题 小游戏

Tetris plus[已完成]

描述

据说没人能超过 3000 分。要是做题做累了，就来玩玩小游戏吧(x

题目地址 https://game.summ3r.top/Tetris/index.html

基准分数 100

当前分数 100

完成人数 378

3000分,一看就不是需要正经完成的游戏,按照惯例按F12查看源代码,想到这是小游戏,于是去看看JS代码

找了很久,在checking文件中找到了判断游戏结束的代码,看到atob中的字符串,于是马上复制过去Base64解码,但险恶的出题人却让结果变成了 "flag被藏起来",看起来是个障眼法.再仔细观察,发现代码下方有一行由!+组成的字符串,是JSFuck密码,于是在控制台中执行,得到flag

Web 第三题 秋名山

Fujiwara Tofu Shop

描述

昨晚我输给一辆AE86。他用惯性漂移过弯，他的车很快，我只看到他有个豆腐店的招牌。

题目地址 http://shop.summ3r.top

基准分数 100

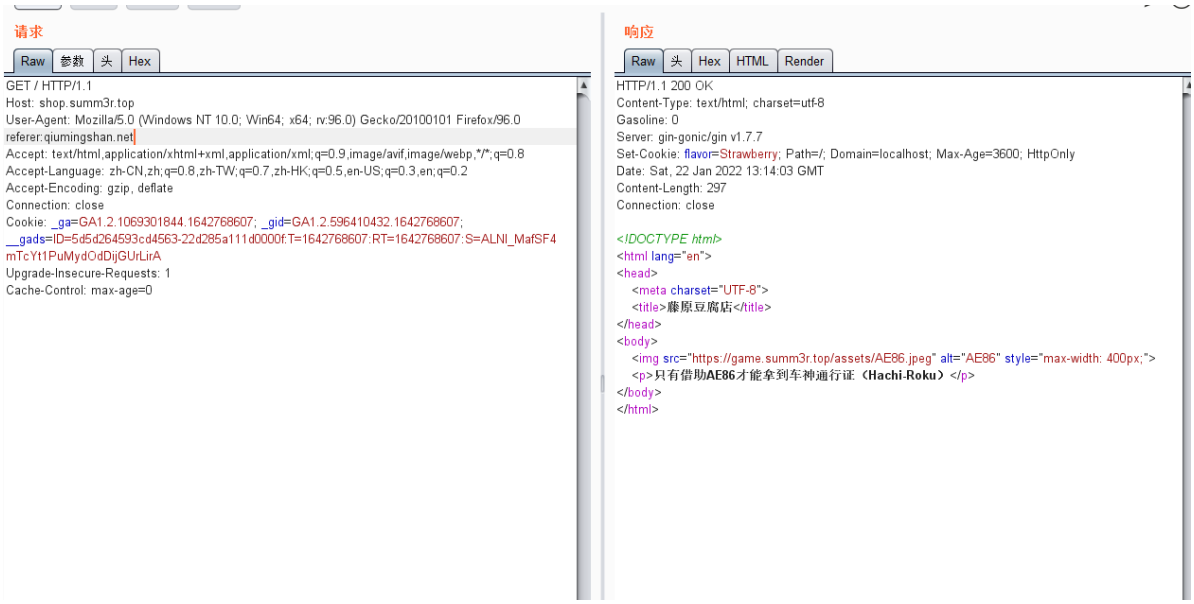
当前分数 100

完成人数 195

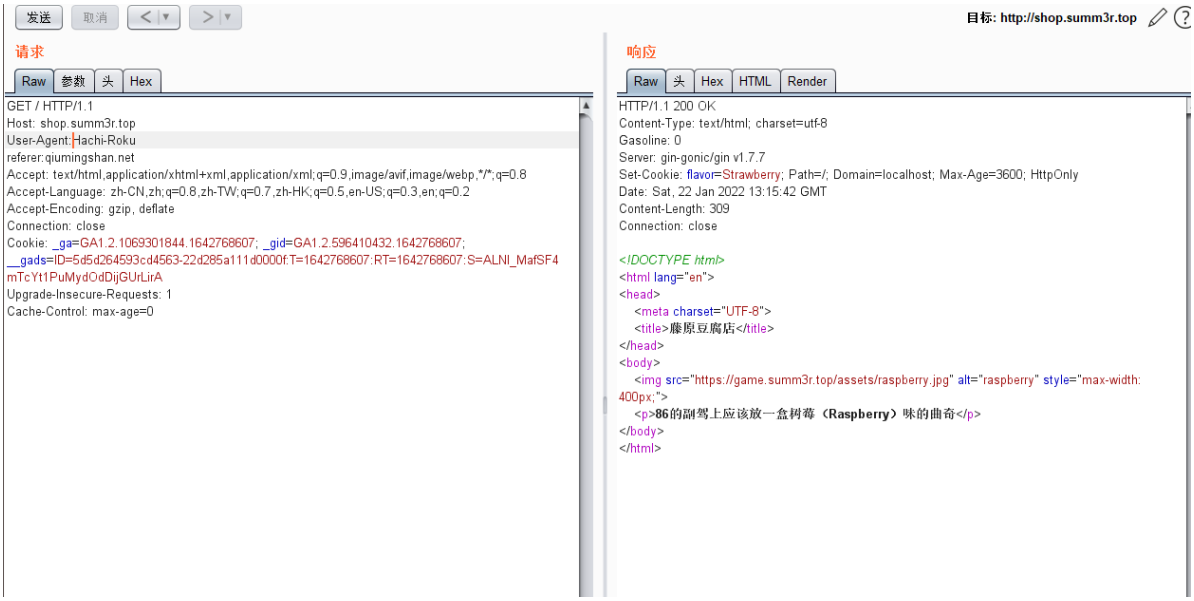
打开链接出现

想成为车神，你需要先去一趟秋名山（qiumingshan.net）

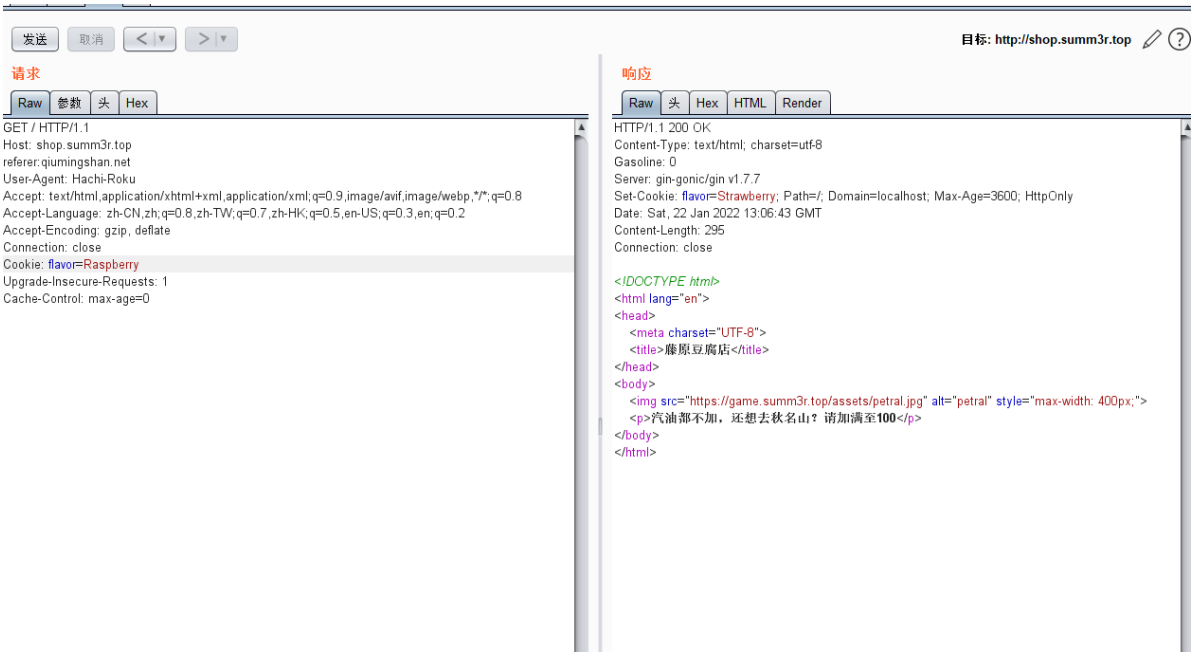
先去一趟秋名山,想到请求头中的referer,于是用burp抓包修改请求头



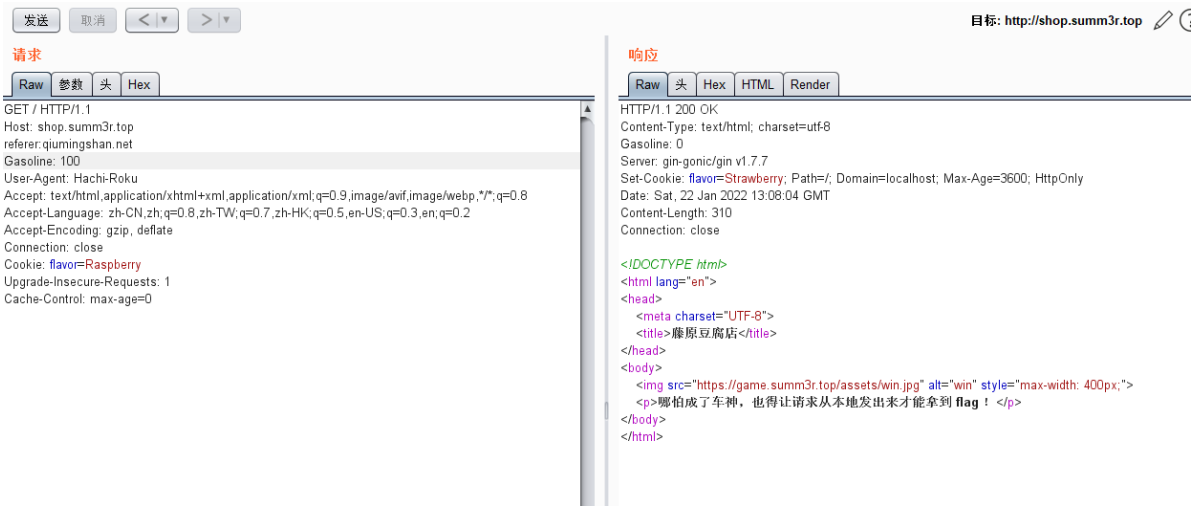
修改后得到右图结果,Hachi-Roku提示需要修改User-Agent,于是再进行修改



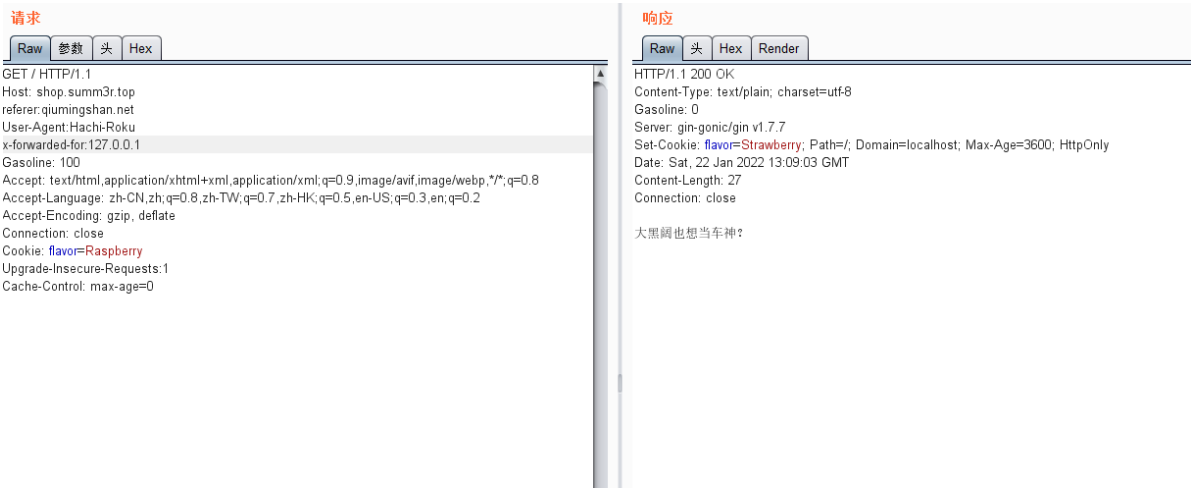
修改后得到右图,再观察右图上方的flavor=Strawberry和树莓味(Raspberry)提示我们要用这个格式修改cookie(曲奇)



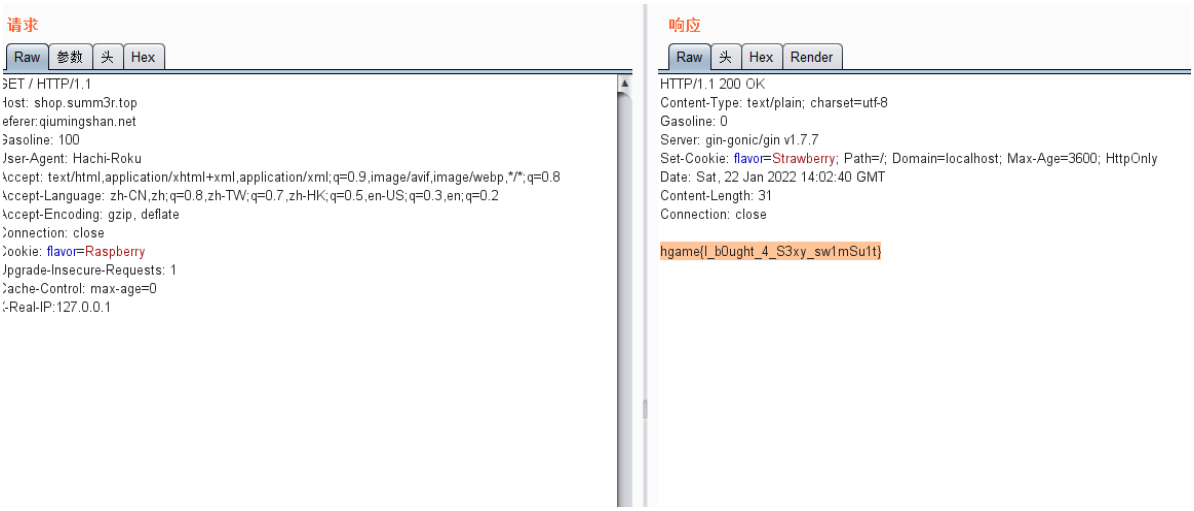
修改后得到右图,提示注满汽油,再观察,发现上方有一个Gasoline,于是猜测将Gasoline值改为100



猜测正确,右图提示要从本地发出,马上想到XFF,于是添加X-forworded-for 为127.0.0.1



但是显示并不正确,思路是正确的,但是XFF可能被出题人ban了,于是百度,发现还可以用X-real-ip,使用后果然正确



CRYPTO 第一题

Easy RSA

描述

这 RSA 不是有手就行？！

(100分的题能拿125分，这不血赚)

题目地址 https://cmfj-1308188104.cos.ap-shanghai.myqcloud.com/Week1/Easy%20RSA.zip

基准分数 125

当前分数 125

完成人数 309

题目中看到RSA,再百度了解相关算法后打开链接

```python
from math import gcd
from random import randint
from gmpy2 import next_prime
from Crypto.Util.number import getPrime
from secret import flag

def encrypt(c):
    p = getPrime(8)
    q = getPrime(8)
    e = randint(0, p * q)
    while gcd(e, (p - 1) * (q - 1)) != 1:
        e = int(next_prime(e))
    return e, p, q, pow(ord(c), e, p * q)

if __name__ == '__main__':
    print(list(map(encrypt, flag)))
    # [(12433, 149, 197, 104), (8147, 131, 167, 6633), (10687, 211, 197, 35594), (19681, 131, 211, 15710), (33577, 251, 211, 38798), (30241, 157, 251, 35973),
```

观察代码发现最下面的是return的结果,c是原flag中的字母,所以目的就是要逆运算算出各个c的字母就行于是百度搜了个脚本运算

```python
import libnum
from Crypto.Util.number import long_to_bytes
e = 13537
q = 179
p = 137
c = 11702
n = p*q
# n = int("",16)

# e = int("",16)


d = libnum.invmod(e, (p - 1) * (q - 1))
m = pow(c, d, n)  # m 的十进制形式
string = long_to_bytes(m)  # m明文
print(string)  # 结果为 b' m ' 的形式
```

把上上图中括号中数字依次代到e、q、p、c中进行运算拼接后的字符串既是flag

hgame{L00ks_l1ke_y0u've_mastered_RS4!}

IOT 第一题

Level - Week1

饭卡的uno[已完成]

描述

饭卡今天第一天学iot 然后他的好朋友Actue让他先去学uno 然后悄悄给饭卡塞了一个固件

题目地址 https://potat0-1308188104.cos.ap-shanghai.myqcloud.com/Week1/%E5%A5%87%E5%A6%99%E7%9A%84%E5%9B%BA%E4%BB%B6.hex

基准分数 100

当前分数 100

完成人数 230

下载来得到一个后缀为hex的文件，百度后知道这是一个内容全部为16进制的文件,刚开始没有思路,后来查看资料后看到可以进行反编译,于是尝试使用IDA进行反编译

```
seg000:00000000
seg000:00000000 ; Processor       : ARM
seg000:00000000 ; ARM architecture: ARMv7-M
seg000:00000000 ; Target assembler: Generic assembler for ARM
seg000:00000000 ; Byte sex        : Little endian
seg000:00000000
seg000:00000000 ; ===========================================================================
seg000:00000000
seg000:00000000 ; Segment type: Pure code
seg000:00000000                 AREA seg000, CODE, READWRITE, ALIGN=0
seg000:00000000                 CODE32
seg000:00000000                 STR     R4, [SP,#0x30]
seg000:00000002                 MOVS    R5, R6
seg000:00000004                 STR     R4, [SP,#0x30]
seg000:00000006                 LSLS    R5, R3, #1
seg000:00000008                 STR     R4, [SP,#0x30]
seg000:0000000A                 LSLS    R5, R3, #1
seg000:0000000C                 STR     R4, [SP,#0x30]
seg000:0000000E                 LSLS    R5, R3, #1
seg000:00000010                 STR     R4, [SP,#0x30]
seg000:00000012                 LSLS    R5, R3, #1
seg000:00000014                 STR     R4, [SP,#0x30]
seg000:00000016                 LSLS    R5, R3, #1
seg000:00000018                 STR     R4, [SP,#0x30]
seg000:0000001A                 LSLS    R5, R3, #1
seg000:0000001C                 STR     R4, [SP,#0x30]
seg000:0000001E                 LSLS    R5, R3, #1
seg000:00000020                 STR     R4, [SP,#0x30]
seg000:00000022                 LSLS    R5, R3, #1
seg000:00000024                 STR     R4, [SP,#0x30]
seg000:00000026                 LSLS    R5, R3, #1
seg000:00000028                 STR     R4, [SP,#0x30]
UNKNOWN 00000000: seg000:00000000
```

上图便是反编译的结果，想用F5发现没有反应，于是尝试性往下翻翻页面

```
seg000:000005B8                 DCB 0xBD
seg000:000005B9                 DCB 0, 0x9B, 0
seg000:000005BC                 DCB 0xAF
seg000:000005BD                 DCB 0, 0x68, 0x67
seg000:000005C0 aAmeF1rst5tep0f DCB "ame{F1rst_5tep_0F_IOT}",0
seg000:000005D7                 DCB 0xD,0xA,0
seg000:000005DA                 % 2
seg000:000005DC                 % 0x11C
seg000:000006F8                 % 4
seg000:000006FC                 % 1
seg000:000006FD                 % 1
seg000:000006FE                 % 1
seg000:000006FF                 % 1
seg000:00000700                 % 1
seg000:00000701                 % 1
```

突然翻到这一行，这应该就是flag了，再在前面加上hg，答案正确

Misc第一题 压缩包

> 这个压缩包有点麻烦[已完成]

描述

这个压缩包，它真的可以打开吗？

（附件已更新，请重新下载）

题目地址

https://potat0-1308188104.cos.ap-shanghai.myqcloud.com/Week1/%E8%BF%99%E4%B8%AA%E5%8E%8B%E7%BC%A9%E5%8C%85%E6%9C%89%E7%82%B9%E9%BA%BB%E7%

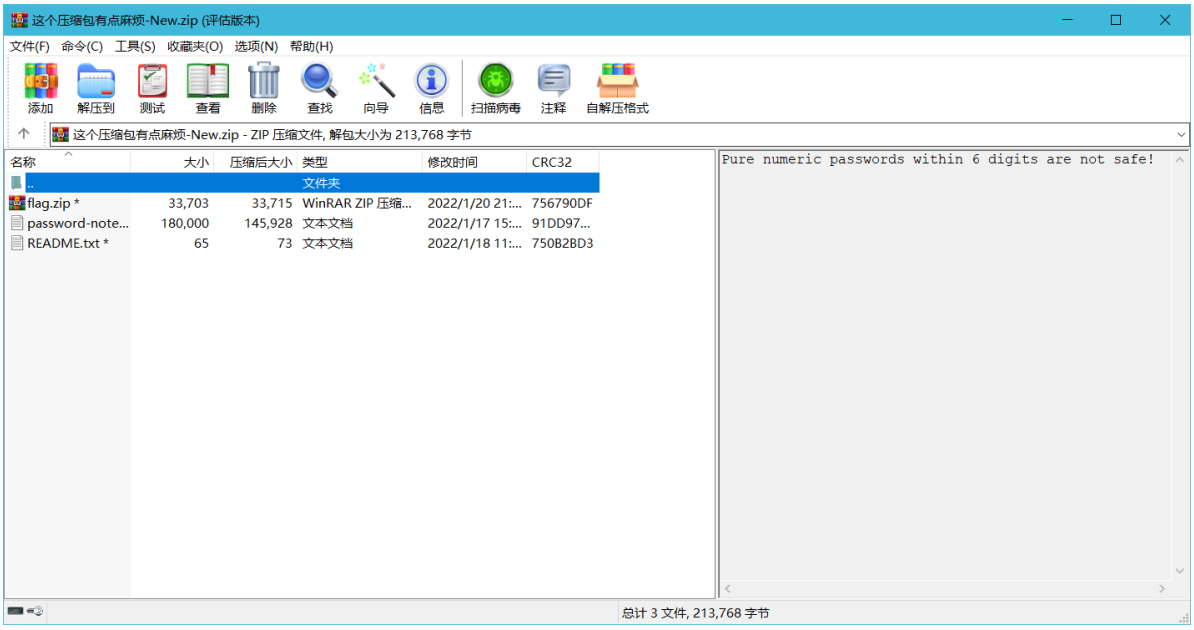
New.zip

基准分数 100

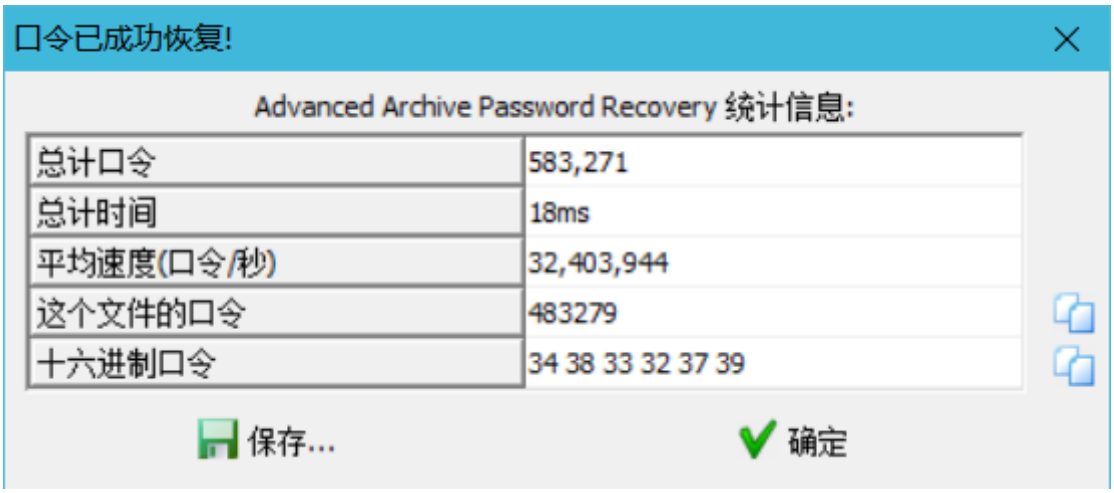
当前分数 100

完成人数 267

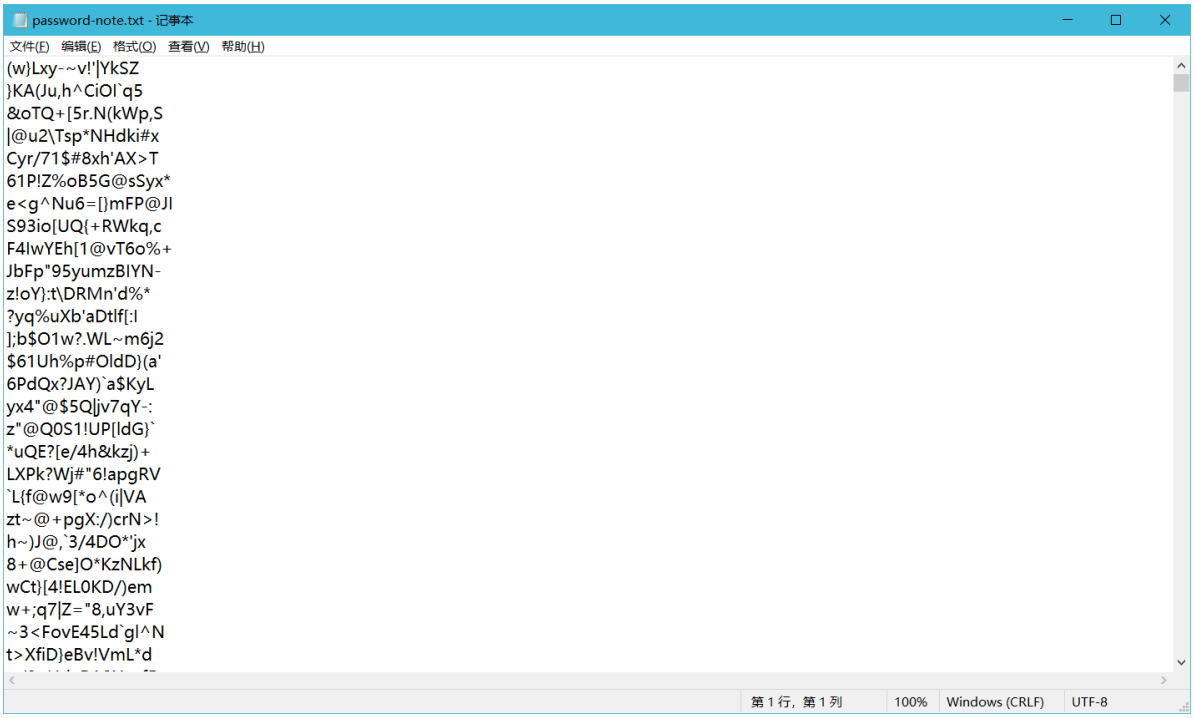
下载后是一个压缩包

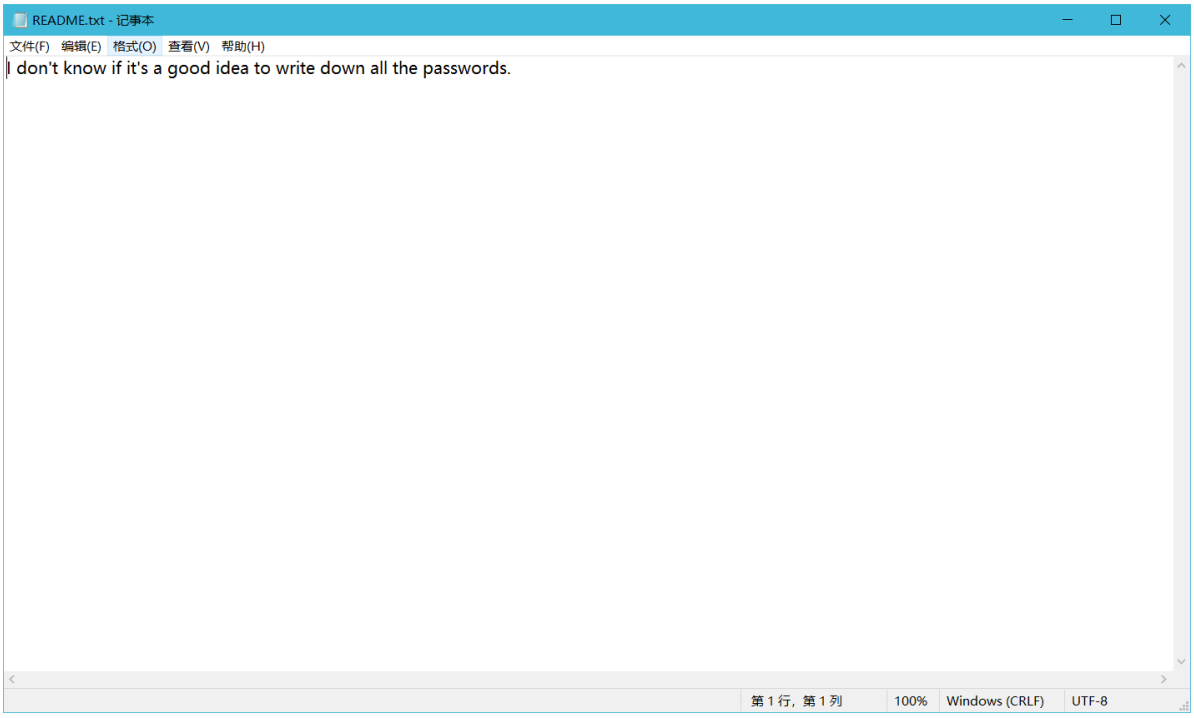


根据提示发现是6位密码,于是使用ARCHPR爆破,几秒钟就爆破出来

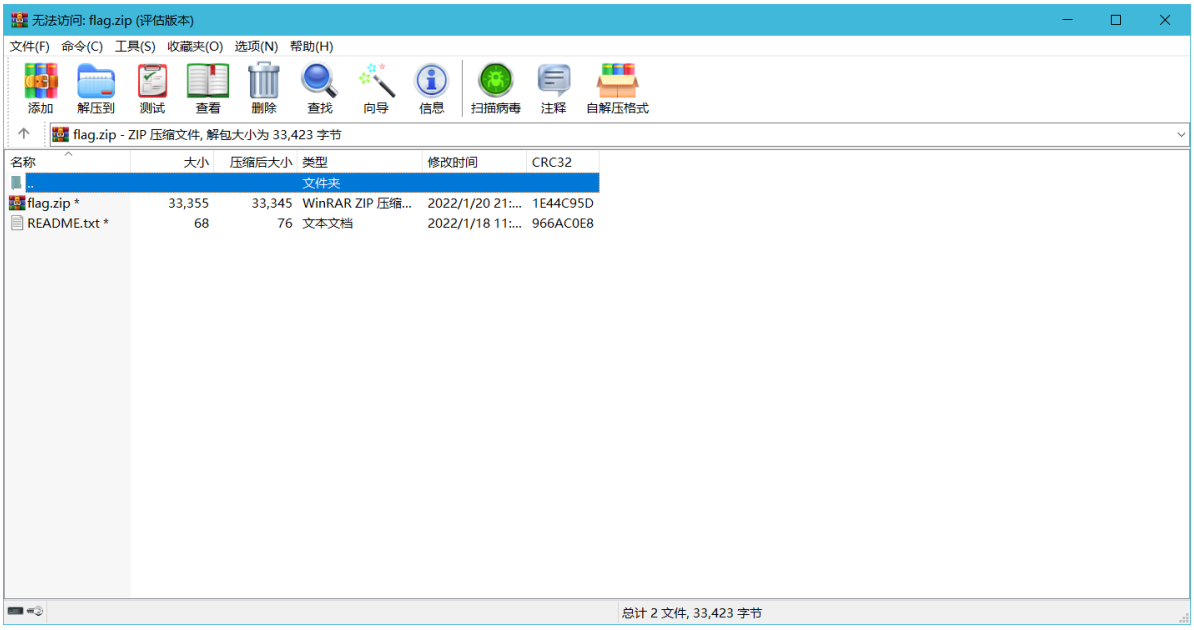


输入密码打开password-note 发现是一个密码字典

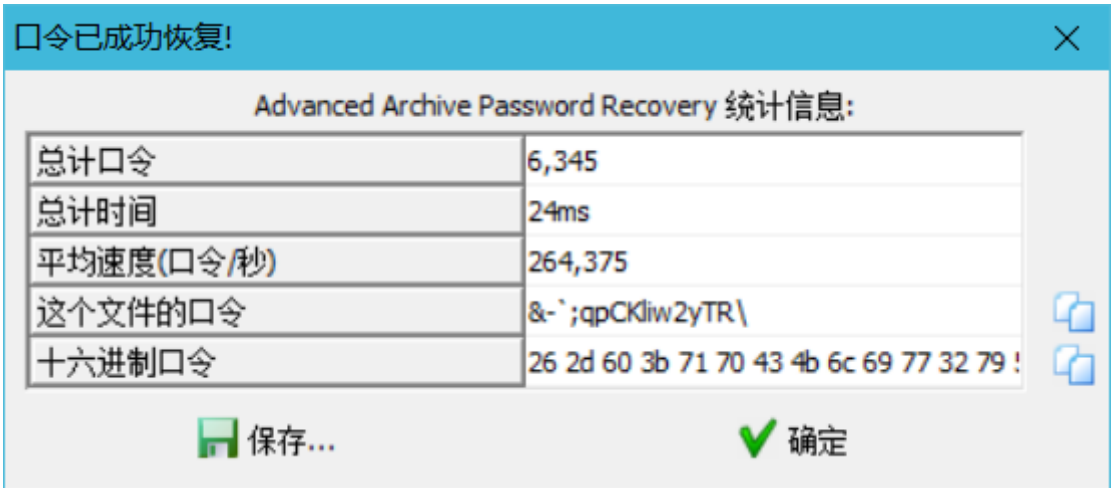




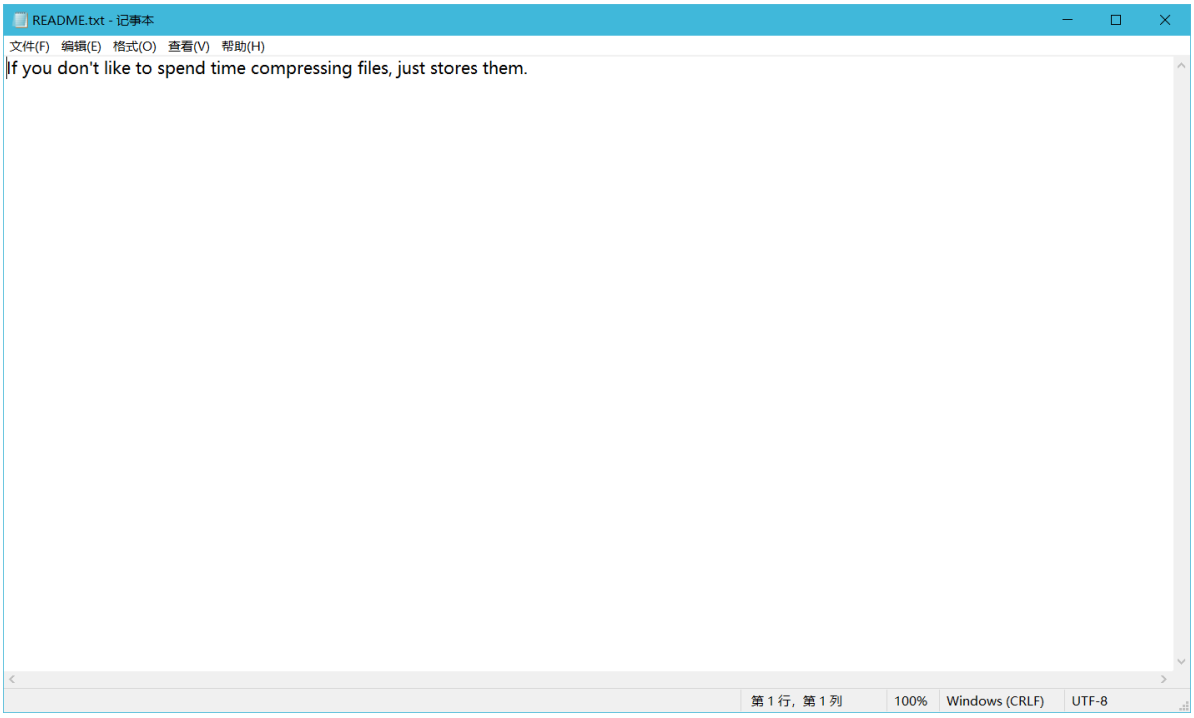
根据提示猜测后面有用,然后打开这个压缩包发现又要解密



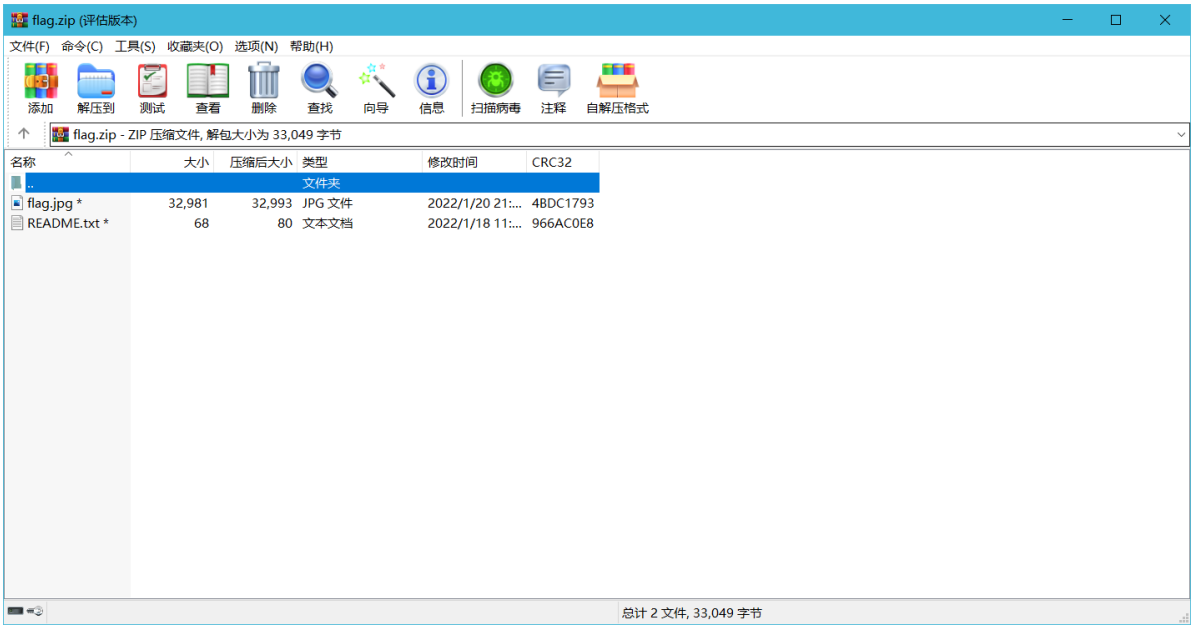
于是使用字典密码爆破,很快得出结果



打开 README.txt

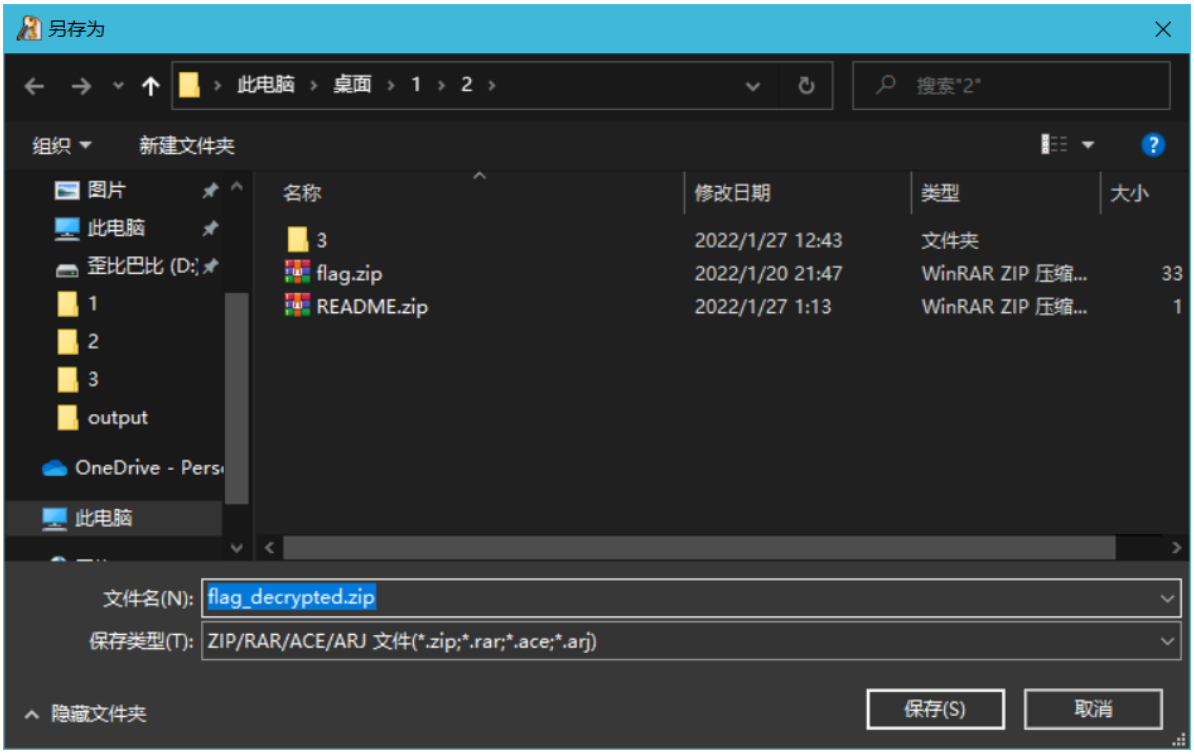


没有什么思路但打开这个压缩包后发现有一个同名文件README.txt

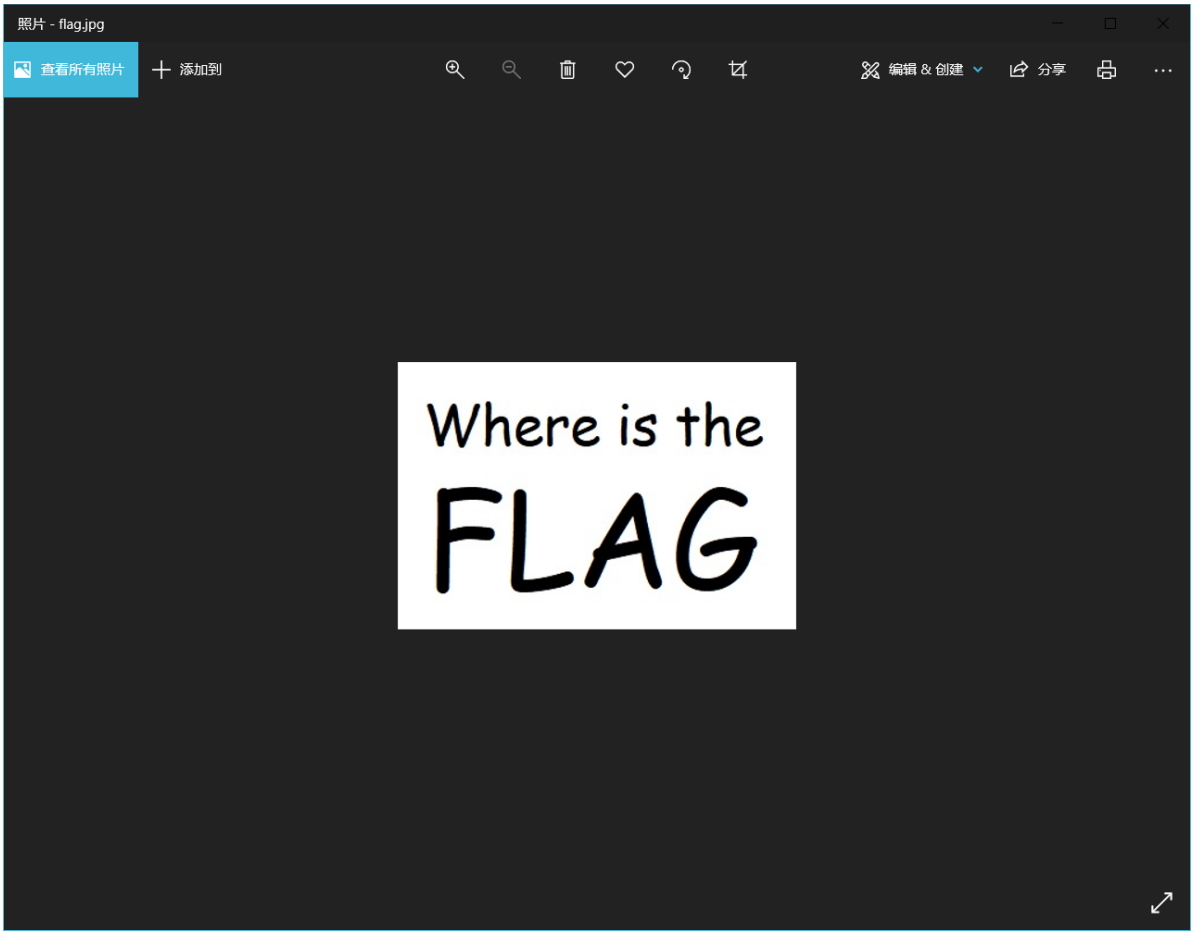


并且两者CRC32相同,于是想到用明文解密

找到了密匙,但一开始我以为找不到口令以为这个思路是错误的,但后来发现没有口令不要紧,可以直接保存一个未加密的zip,如下图



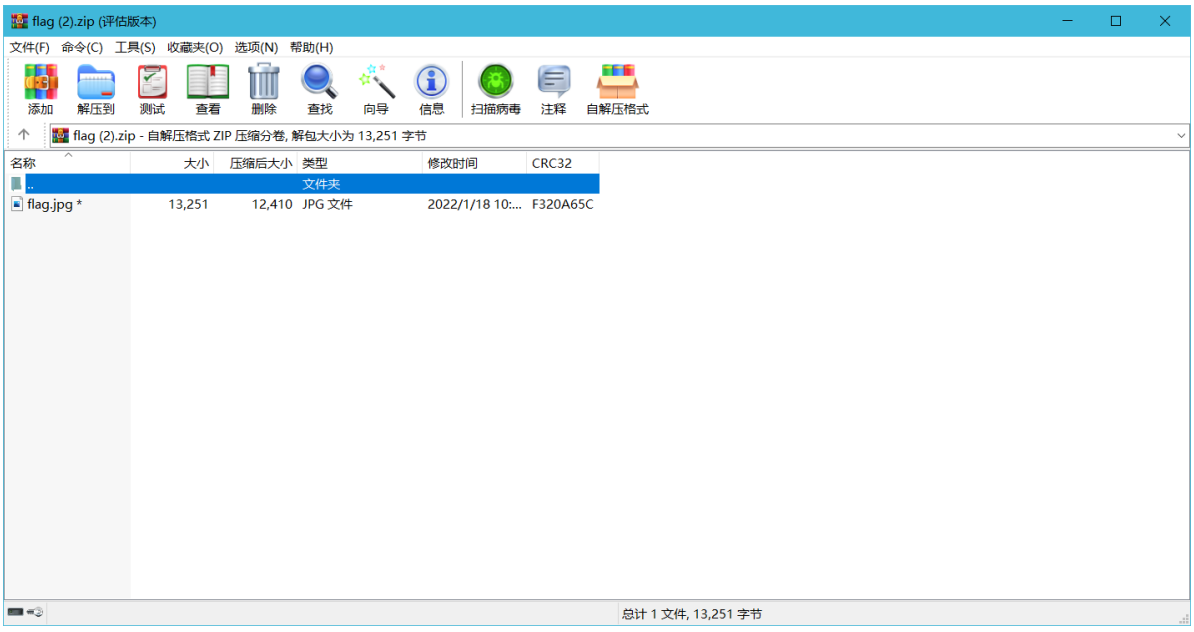
打开这个刚保存的zip里面有个flag.jpg 打开



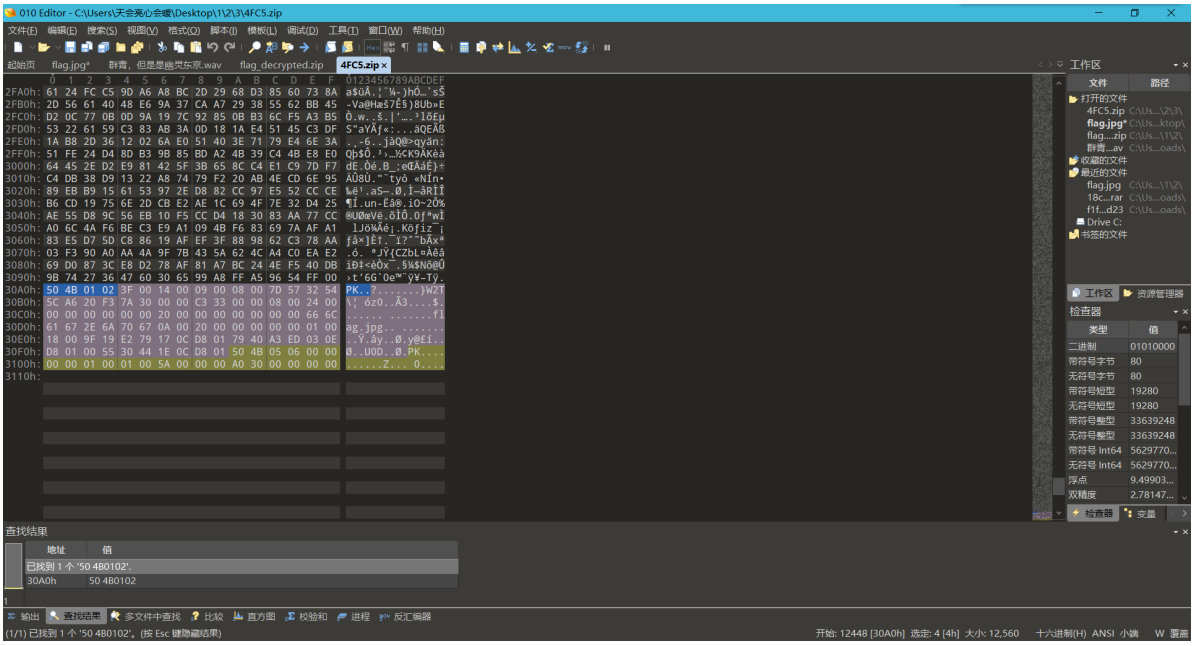
尝试着用010Editor查看这张图片,翻到最后面发现

```
FB0h: 71 79 E4 6E 3A 51 FE 24 D4 8D B3 9B 85 BD A2 4B  qyän:Qþ$Ô.³›…½¢K
FC0h: 39 C4 4B E8 E0 64 45 2E D2 E9 81 42 5F 3B 65 8C  9ÄKèàdE.Òé.B_;eŒ
FD0h: C4 E1 C9 7D F7 C4 DB 38 D9 13 22 A8 74 79 F2 20  ÄáÉ}÷ÄÛ8Ù."¨tyò
FE0h: AB 4E CD 6E 95 89 EB B9 15 61 53 97 2E D8 82 CC  «NÍn•‰ë¹.aS—.Ø‚Ì
FF0h: 97 E5 52 CC CE B6 CD 19 75 6E 2D CB E2 AE 1C 69  —åRÌÎ¶Í.un-Ëâ®.i
000h: 4F 7E 32 D4 25 AE 55 D8 9C 56 EB 10 F5 CC D4 18  O~2Ô%®UØœVë.õÌÔ.
010h: 30 83 AA 77 CC A0 6C 4A F6 BE C3 E9 A1 09 4B F6  0ƒªwÌ lJö¾Ãé¡.Kö
020h: 83 69 7A AF A1 83 E5 D7 5D C8 86 19 AF EF 3F 88  fiz¯¡få×]È†.¯ï?ˆ
030h: 98 62 C3 78 AA 03 F3 90 A0 AA 4A 9F 7B 43 5A 62  ˜bÃxª.ó. ªJŸ{CZb
040h: 4C A4 C0 EA E2 69 D0 87 3C E8 D2 78 AF 81 A7 BC  L¤ÀêâiÐ‡<èÒx¯.§¼
050h: 24 4E F5 40 DB 9B 74 27 36 47 60 30 65 99 A8 FF  $Nõ@Û›t'6G`0e™¨ÿ
060h: A5 96 54 FF 00 50 4B 01 02 3F 00 14 00 09 00 08  ¥–Tÿ.PK..?......
070h: 00 7D 57 32 54 5C A6 20 F3 7A 30 00 00 C3 33 00  .}W2T\¦ óz0..Ã3.
080h: 00 08 00 24 00 00 00 00 00 00 00 20 00 00 00 00  ...$....... ....
090h: 00 00 00 66 6C 61 67 2E 6A 70 67 0A 00 20 00 00  ...flag.jpg.. ..
0A0h: 00 00 00 01 00 18 00 9F 19 E2 79 17 0C D8 01 79  .......Ÿ.ây..Ø.y
0B0h: 40 A3 ED 03 0E D8 01 00 55 30 44 1E 0C D8 01 50  @£í..Ø..U0D..Ø.P
0C0h: 4B 05 06 00 00 00 00 01 00 01 00 5A 00 00 00 A0  K..........Z...
0D0h: 30 00 00 00 00                                   0....
```

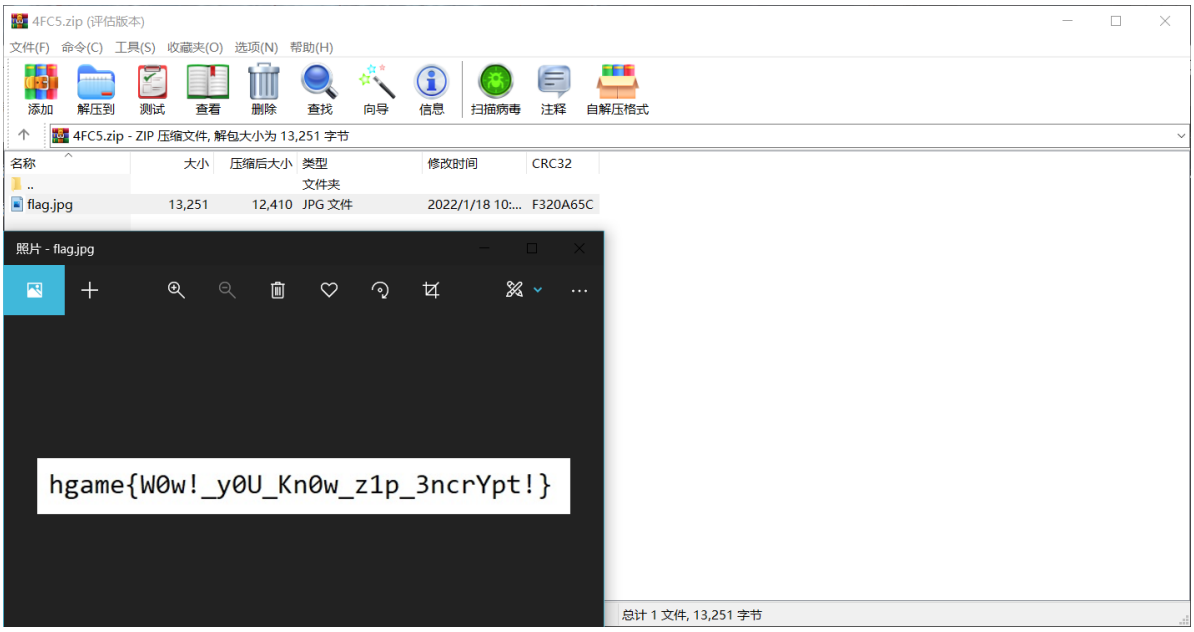
出现了PK两字,于是猜测这是zip,将其后缀改为zip,能够正常打开,说明猜测正确打开后又出现一个需要解密的图片



再想想常用的ZIP解密方法,于是猜测这是一个伪加密用010Editor打开这个zip,搜索504B0102



然后将从50开始的第9、10个字节改为0000，即把本题中0900改为0000，于是解密成功



打开图片即是flag