




【MISC怼题入门系列】BUU-MISC-page2

原创

Em0s_Er1t  于 2021-05-31 20:45:48 发布  581  收藏 2

分类专栏: [CTF-MISC](#) 文章标签: [python](#) [加密解密](#) [bmp](#) [编码学](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/m0_46296905/article/details/116936356

版权



[CTF-MISC 专栏收录该内容](#)

4 篇文章 0 订阅

订阅专栏

buu-misc系列第二页的wp来了

BUU-MISC-page2-wp:

0x00 被劫持的神秘礼物

0x01 刷新过的图片

(1) F5隐写

0x02 snake

0x03 梅花香之苦寒来

0x04 菜刀666

0x05 [BJDCTF2020]认真你就输了

0x06 被偷走的文件

0x07 [GXYCTF2019]佛系青年

0x08 [BJDCTF2020]藏藏藏

0x09 秘密文件

0x0A [BJDCTF2020]你猜我是个啥

0x0B [SWPU2019]神奇的二维码

0x0C [BJDCTF2020]一叶障目

0x0D [BJDCTF2020]鸡你太美

0x0E 穿越时空的思念

0x0F [BJDCTF2020]just_a_rar

0x10 [BJDCTF2020]纳尼

0x11 [ACTF新生赛2020]outguess

outguess隐写

1.outguess工具安装

2.outguess工具基本使用

(1) 写入:

(2) 提取:

(3) outguess --help

0x12 [SWPU2019]我有一只马里奥

0x13 谁赢了比赛?

0x14 [GXYCTF2019]gakki

0x15 [HBNIS2018]excel破解

0x16 Mysterious

0x17 [HBNIS2018]来题中等的吧

0x18 [ACTF新生赛2020]base64隐写

0x19 [SWPU2019]伟大的侦探

——福尔摩斯小人编密码表

0x1A [WUSTCTF2020]find_me

0x1B 黑客帝国

0x1C 喵喵喵

0x1D [SWPU2019]你有没有好好看网课?

敲击码

0x1E [MRCTF2020]你能看懂音符吗

0x00 被劫持的神秘礼物

题目:

某天小明收到了一件很特别的礼物,有奇怪的后缀,奇怪的名字和格式。小明找到了知心姐姐度娘,度娘好像知道这是啥,但是度娘也不知道里面是啥。。。你帮帮小明?找到帐号密码,串在一起,用32位小写MD5哈希一下得到的就是答案。链接:

https://pan.baidu.com/s/1pwVvPA5_WWY8Og6dhCcWRw 提取码: 31vk

.pcapng 文件,用wireshark打开,

The image shows a Wireshark capture of network traffic. The main pane displays a list of packets. Packet 4 is highlighted, showing an HTTP POST request to /index.php?r=member/index/login. The packet details pane shows the request body as 'member/index/login'. The packet bytes pane shows the raw hex and ASCII data of the request body.

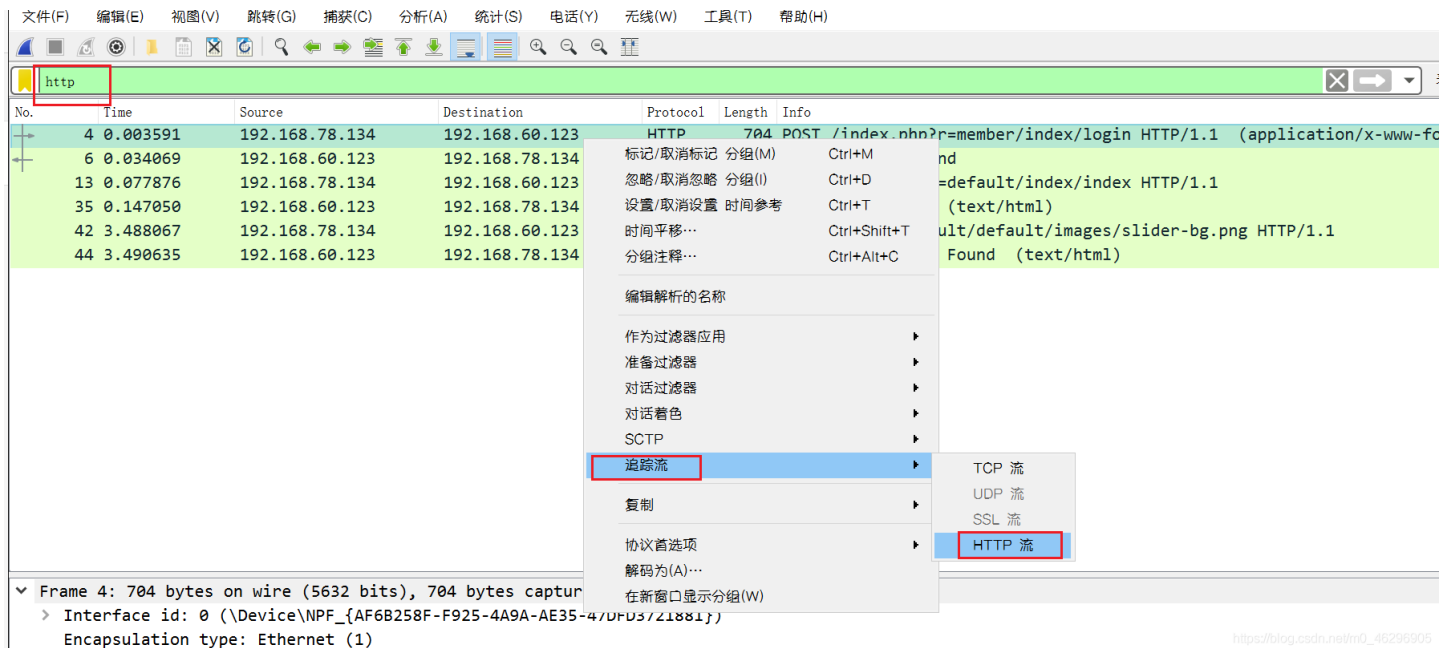
| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|----------|----------------|----------------|----------|--------|--|
| 1 | 0.000000 | 192.168.78.134 | 192.168.60.123 | TCP | 62 | 1328 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 |
| 2 | 0.002878 | 192.168.60.123 | 192.168.78.134 | TCP | 60 | 80 → 1328 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 |
| 3 | 0.002937 | 192.168.78.134 | 192.168.60.123 | TCP | 54 | 1328 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 4 | 0.003591 | 192.168.78.134 | 192.168.60.123 | HTTP | 704 | POST /index.php?r=member/index/login HTTP/1.1 (application/x-www-form-urlencoded) |
| 5 | 0.004248 | 192.168.60.123 | 192.168.78.134 | TCP | 60 | 80 → 1328 [ACK] Seq=1 Ack=651 Win=64240 Len=0 |
| 6 | 0.034069 | 192.168.60.123 | 192.168.78.134 | HTTP | 619 | HTTP/1.1 302 Found |
| 7 | 0.034146 | 192.168.78.134 | 192.168.60.123 | TCP | 54 | 1328 → 80 [ACK] Seq=651 Ack=567 Win=63675 Len=0 |
| 8 | 0.035203 | 192.168.78.134 | 192.168.60.123 | TCP | 54 | 1328 → 80 [FIN, ACK] Seq=651 Ack=567 Win=63675 Len=0 |
| 9 | 0.035404 | 192.168.60.123 | 192.168.78.134 | TCP | 60 | 80 → 1328 [ACK] Seq=567 Ack=652 Win=64239 Len=0 |
| 10 | 0.053720 | 192.168.78.134 | 192.168.60.123 | TCP | 62 | 1329 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 |
| 11 | 0.077558 | 192.168.60.123 | 192.168.78.134 | TCP | 60 | 80 → 1329 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 |
| 12 | 0.077603 | 192.168.78.134 | 192.168.60.123 | TCP | 54 | 1329 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0 |
| 13 | 0.077876 | 192.168.78.134 | 192.168.60.123 | HTTP | 615 | GET /index.php?r=default/index/index HTTP/1.1 |
| 14 | 0.078086 | 192.168.60.123 | 192.168.78.134 | TCP | 60 | 80 → 1329 [ACK] Seq=1 Ack=562 Win=64240 Len=0 |
| 15 | 0.124769 | 192.168.60.123 | 192.168.78.134 | TCP | 1514 | 80 → 1329 [ACK] Seq=1 Ack=562 Win=64240 Len=1460 [TCP segment of a reassembled data stream] |
| 16 | 0.124800 | 192.168.60.123 | 192.168.78.134 | TCP | 1514 | 80 → 1329 [ACK] Seq=1461 Ack=562 Win=64240 Len=1460 [TCP segment of a reassembled data stream] |
| 17 | 0.124811 | 192.168.60.123 | 192.168.78.134 | TCP | 606 | 80 → 1329 [PSH, ACK] Seq=2921 Ack=562 Win=64240 Len=552 [TCP segment of a reassembled data stream] |
| 18 | 0.124821 | 192.168.60.123 | 192.168.78.134 | TCP | 1514 | 80 → 1329 [ACK] Seq=3473 Ack=562 Win=64240 Len=1460 [TCP segment of a reassembled data stream] |
| 19 | 0.124831 | 192.168.60.123 | 192.168.78.134 | TCP | 1514 | 80 → 1329 [ACK] Seq=4933 Ack=562 Win=64240 Len=1460 [TCP segment of a reassembled data stream] |

▼ Frame 4: 704 bytes on wire (5632 bits), 704 bytes captured (5632 bits) on interface 0
 > Interface id: 0 (\Device\NPF_{AF6B258F-F925-4A9A-AE35-47DFD3721881})
 Encapsulation type: Ethernet (1)
 Arrival Time: Dec 3, 2014 13:51:35.887100000 中国标准时间
 [Time shift for this packet: 0.00000000 seconds]

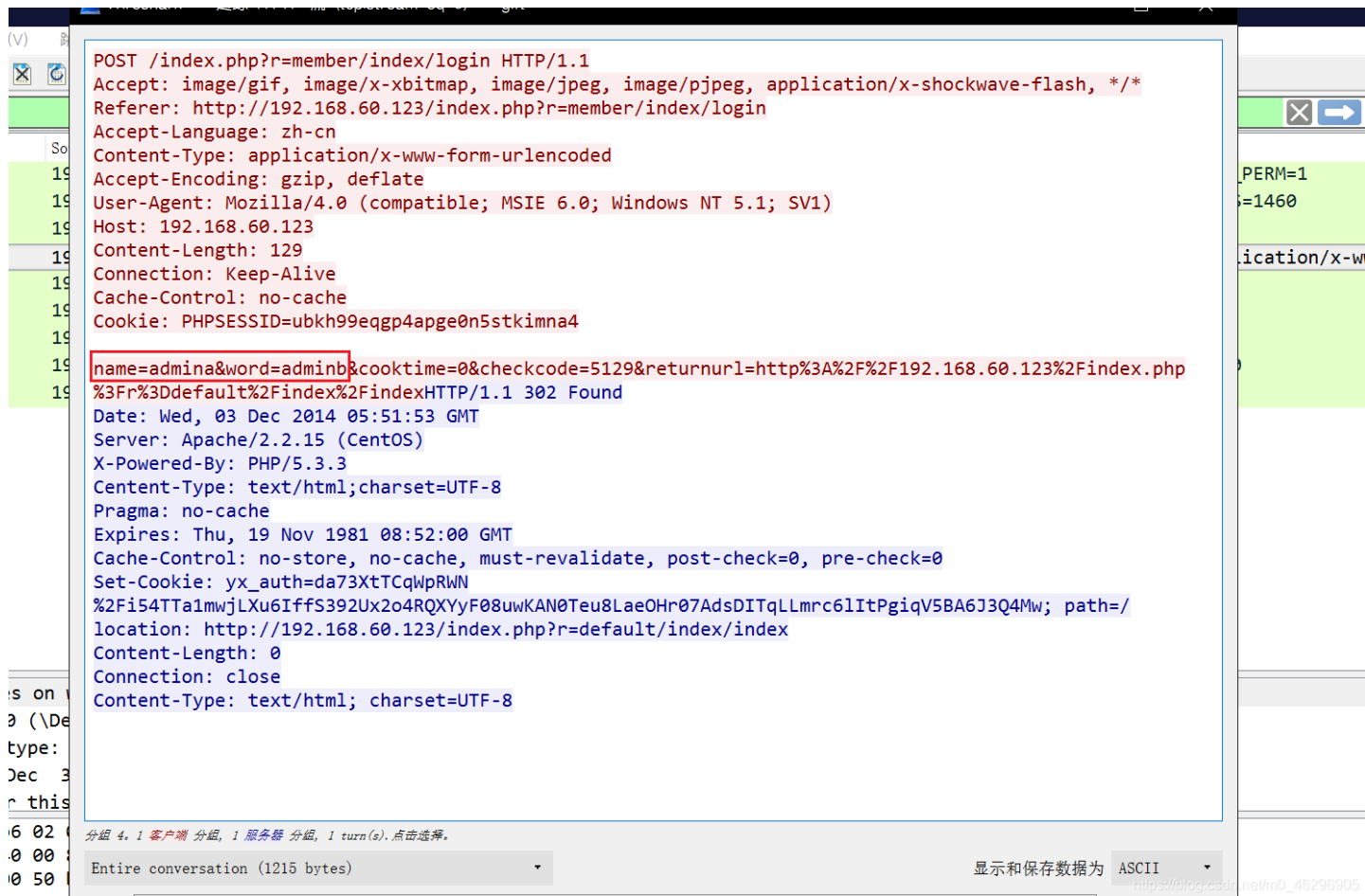
```

0000  00 50 56 f7 b6 02 00 0c 29 19 0c f1 08 00 45 00  .PV.... )....E.
0010  02 b2 1b 6e 40 00 80 06 d0 85 c0 a8 4e 86 c0 a8  ...n@... ..N...
0020  3c 7b 05 30 00 50 b2 89 44 9d 62 97 70 ae 50 18  <{.P.. D.b.p.P.
0030  fa f0 a0 98 00 00 50 4f 53 54 20 2f 69 6e 64 65  .....PO ST /inde
  
```

输入 http 过滤一下http流,右键追踪

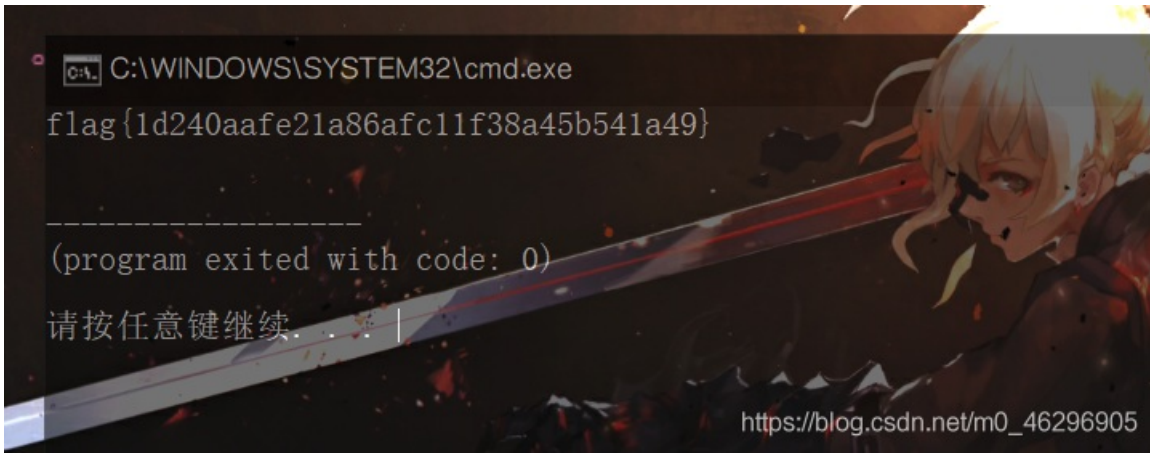


找到用户名是 `admina` 和密码是 `adminb`



题目说是 找到帐号密码，串在一起，用32位小写MD5哈希一下得到的就是答案，写个exp哈希一下吧

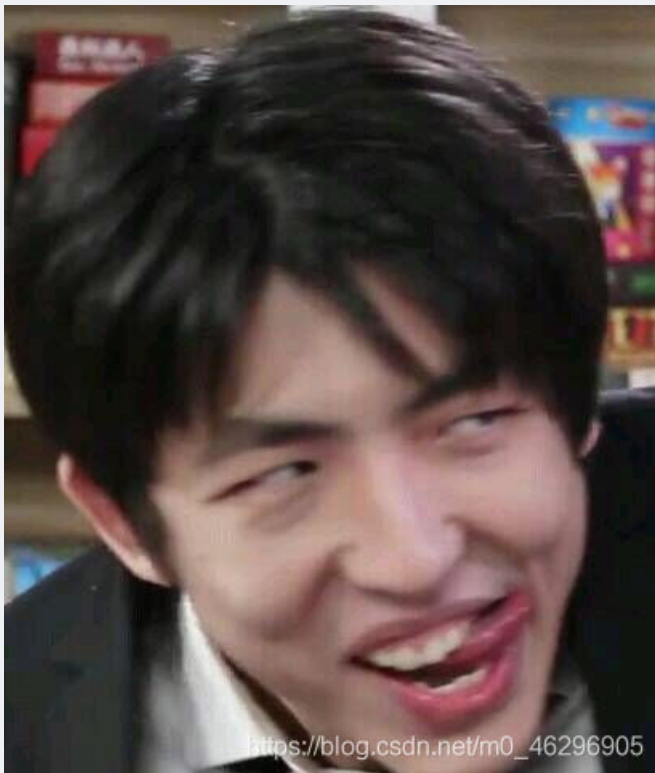
```
import hashlib
m=hashlib.md5()
m.update("adminaadminb".encode('utf-8'))
print('flag{'+m.hexdigest()+}'')
```



```
flag{1d240aafe21a86afc11f38a45b541a49}
```

0x01 刷新过的图片

题目：
浏览图片的时候刷新键有没有用呢



通过这道题知道了一个新的隐写方式——F5 隐写

(1) F5隐写

参考另一篇博客——【隐写术】F5隐写

先在电脑上安装一下，<https://github.com/matthewgao/F5-steganography>，解压即可用

安装目录下命令行输入

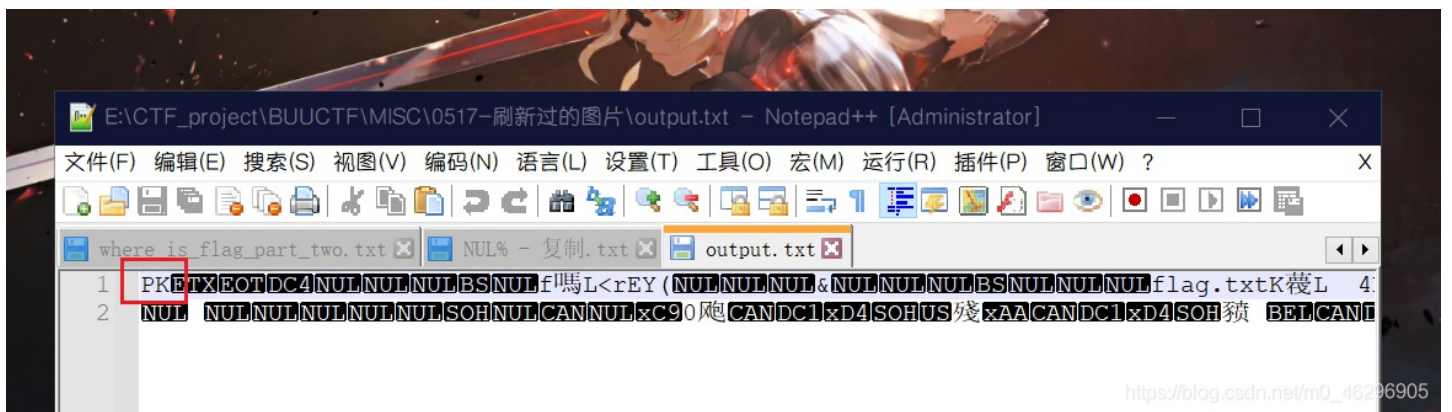
```
java Extract Misc.jpg
```

```
D:\CTF_Tools\F5-steganography>java Extract Misc.jpg
Huffman decoding starts
Permutation starts
309504 indices shuffled
Extraction starts
Length of embedded file: 190 bytes
(1, 31, 5) code used

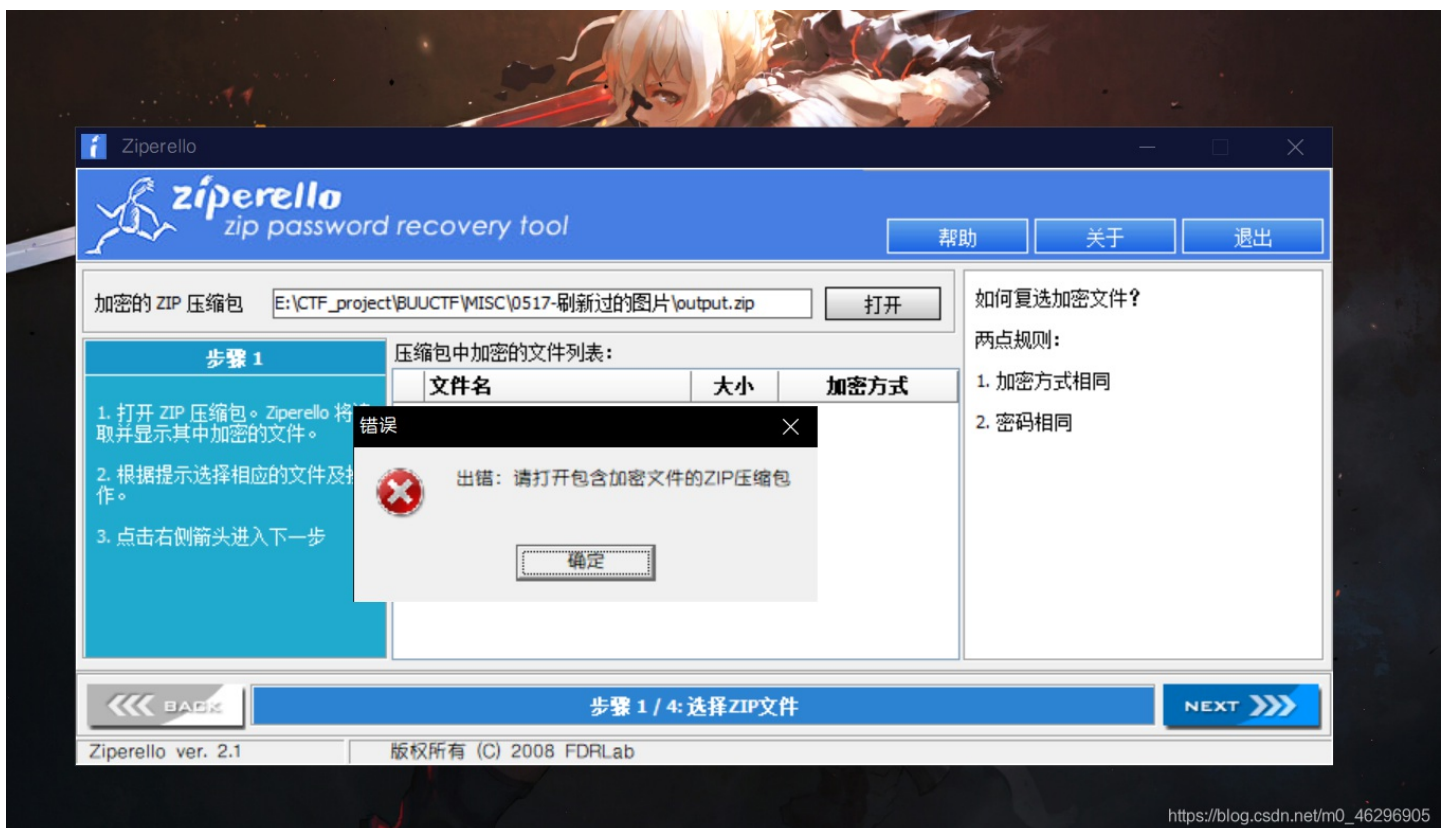
D:\CTF_Tools\F5-steganography> |
```

https://blog.csdn.net/m0_46296905

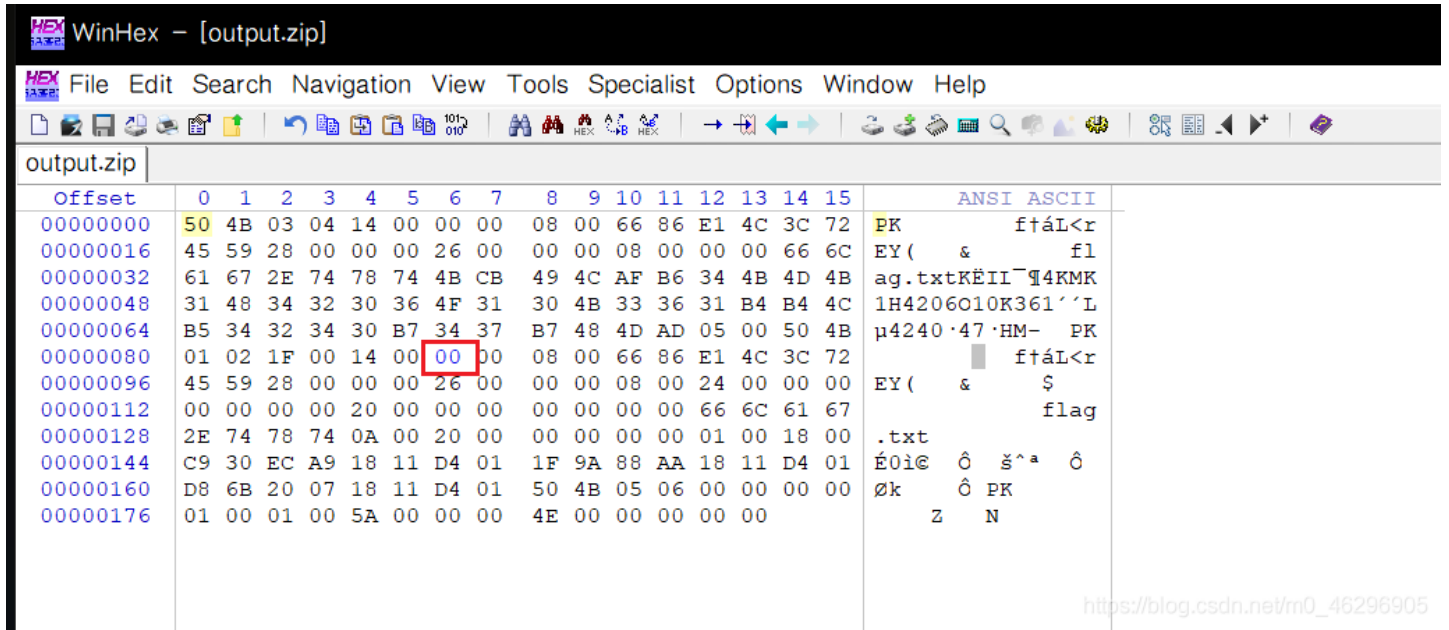
提取出output.txt文件



以PK开头，那就改后缀为zip，放到ziperello破解一下压缩密码，但显示无密码，推测是伪加密



winhex 打开output.zip文件，改完后保存



https://blog.csdn.net/m0_46296905

解压得到flag



https://blog.csdn.net/m0_46296905

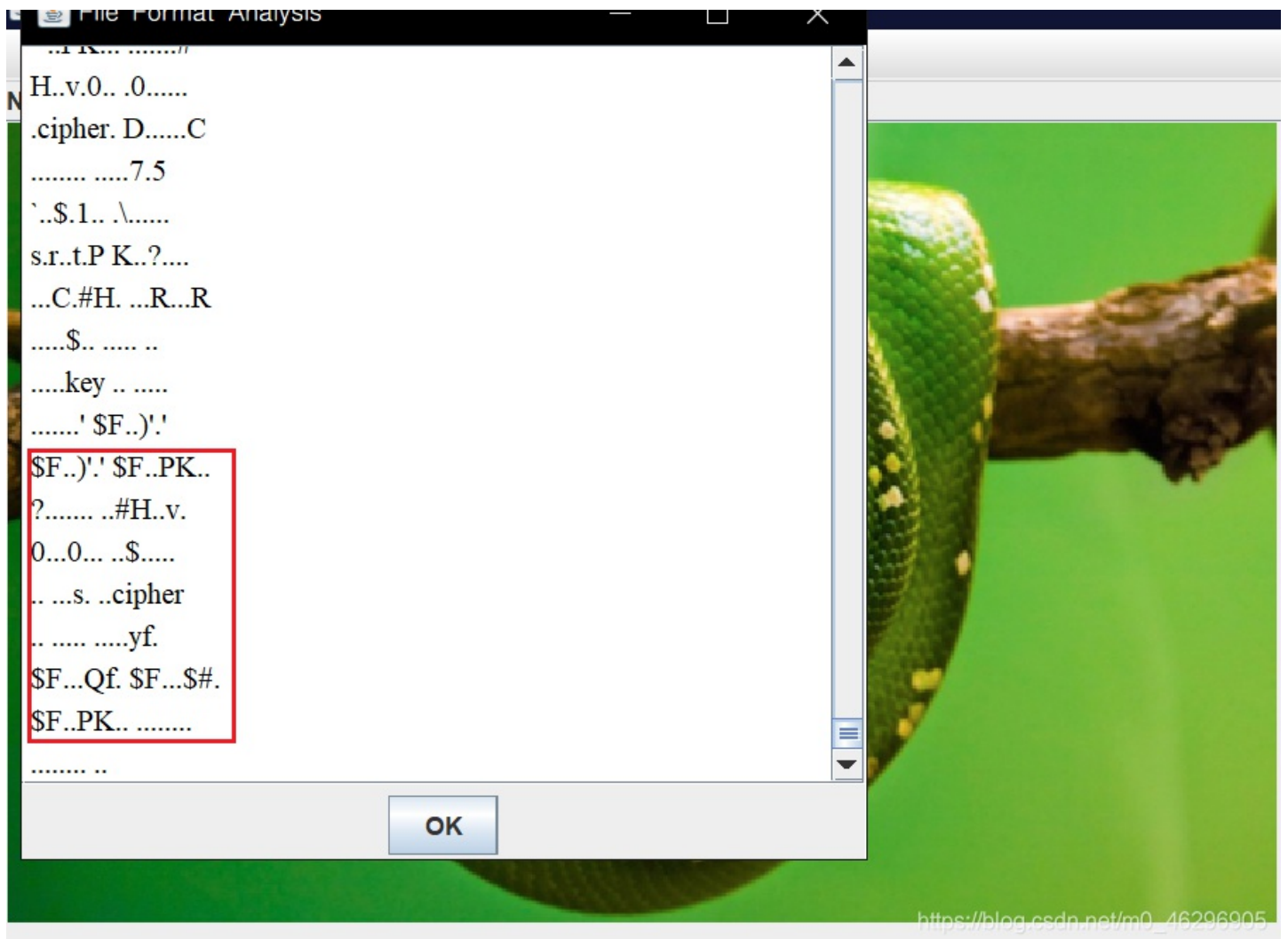
flag{96efd0a2037d06f34199e921079778ee}

0x02 snake

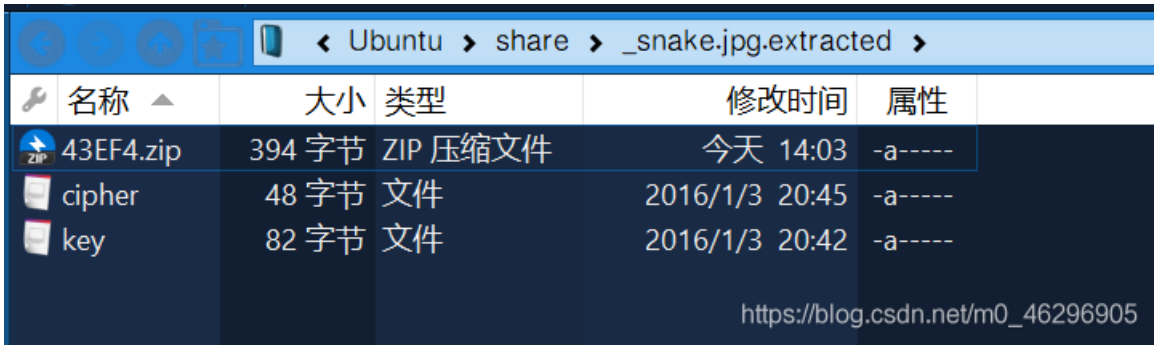
题目:



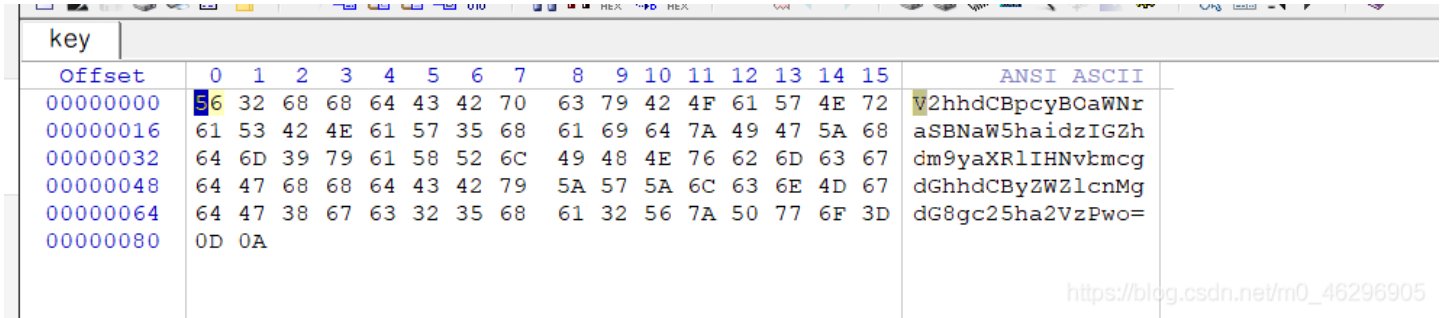
stegsolve打开看看Format Analysis，直接拉到文件最后，发现了压缩包的头文件字符PK，还发现了一些似乎是文件名称的字符，猜测有隐藏文件，



binwalk提取一下看看，得到三个文件，zip有密码，密码应该是通过另外两个文件解得的明文。



winhex打开key文件看看密钥，是个base64编码的字符串

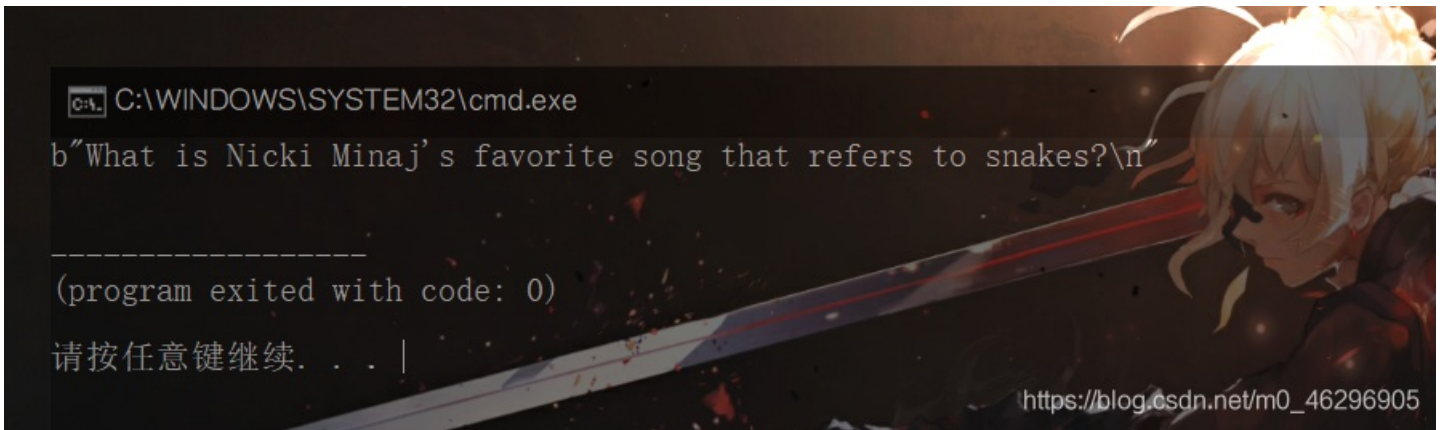


V2hhdCBpCyBOaWNraSBNaW5haidzIGZhdm9yaXRlIHNVbmcgdGhhdCByZWZlcnMgdG8gc25ha2VzPwo=

写个exp解一下

```
import base64
print(base64.b64decode('V2hhdCBpCyBOaWNraSBNaW5haidzIGZhdm9yaXRlIHNVbmcgdGhhdCByZWZlcnMgdG8gc25ha2VzPwo='))
```

得到密钥



What is Nicki Minaj's favorite song that refers to snakes?\n

Nicki Minaj是个歌手，出过的关于蛇的专辑是Anaconda，anaconda就是密钥了。

现在只剩不知道加密方式了，打开密文文件看看，没有发现什么

| cipher | | | | | | | | | | | | | | | | | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------|
| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI ASCII |
| 00000000 | DC | 44 | 15 | 8C | D6 | A2 | 83 | B5 | 43 | B4 | 12 | F7 | 16 | A7 | D1 | FD | ÜD €ÖçfµC' ÷ \$Ñý |
| 00000016 | D2 | 10 | D8 | EB | 9E | 89 | 37 | E2 | 35 | 60 | F9 | EE | 24 | 01 | 31 | BF | Ò øëž%7â5`ùî\$ 1¿ |
| 00000032 | 1C | E7 | 5C | AB | B6 | 8E | BF | DA | 83 | 73 | 0C | 72 | 8D | BC | 74 | 8D | ç\«Ź¿Úfs r ¼t |

参考别的博主的wp知道用的是serpent加密（serpent译为蛇），这个算法当时跟Rijndael算法一同参与AES投标，虽然落选，但却比Rijndael更为安全

[serpent在线解密](#)

解得flag

Serpent – Symmetric Ciphers Online

Input type: File

File: C:\fakepath\cipher Browse

Function: SERPENT

Mode: ECB (electronic codebook)

Key: anaconda

Plaintext Hex

> Encrypt! > Decrypt! ▶ 🔗

0%
File was uploaded.

Decrypted text:

| | | | | | | | | | | | | | | | | | |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|
| 00000000 | 43 | 54 | 46 | 7b | 77 | 68 | 6f | 5f | 6b | 6e | 65 | 77 | 5f | 73 | 65 | 72 | C T F { w h o _ k n e w _ s e r |
| 00000010 | 70 | 65 | 6e | 74 | 5f | 63 | 69 | 70 | 68 | 65 | 72 | 5f | 65 | 78 | 69 | 73 | p e n t _ c i p h e r _ e x i s |
| 00000020 | 74 | 65 | 64 | 7d | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | t e d } . . . k i t / 9 i n g . . . k i t 4 8 2 9 9 5 |

flag{who_knew_serpent_cipher_existed}

0x03 梅花香之苦寒来

题目：



右键看看图片属性。提示我们flag藏在图片的最后。画图??



winhex打开，图片后面全是一大串数据，看着好像有些规律

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI | ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|-------|
| 00668592 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 31 | 33 | 33 | 32 | 39 | 283237312c313329 | |
| 00668608 | 30 | 61 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 31 | 33 | 34 | 0a283237312c3134 | |
| 00668624 | 32 | 39 | 30 | 61 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 31 | 290a283237312c31 | |
| 00668640 | 33 | 35 | 32 | 39 | 30 | 61 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 35290a283237312c | |
| 00668656 | 33 | 31 | 33 | 36 | 32 | 39 | 30 | 61 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 3136290a28323731 | |
| 00668672 | 32 | 63 | 33 | 31 | 33 | 37 | 32 | 39 | 30 | 61 | 32 | 38 | 33 | 32 | 33 | 37 | 2c3137290a283237 | |
| 00668688 | 33 | 31 | 32 | 63 | 33 | 31 | 33 | 38 | 32 | 39 | 30 | 61 | 32 | 38 | 33 | 32 | 312c3138290a2832 | |
| 00668704 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 31 | 33 | 39 | 32 | 39 | 30 | 61 | 32 | 38 | 37312c3139290a28 | |
| 00668720 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 32 | 33 | 30 | 32 | 39 | 30 | 61 | 3237312c3230290a | |
| 00668736 | 32 | 38 | 33 | 32 | 33 | 37 | 33 | 31 | 32 | 63 | 33 | 32 | 33 | 31 | 32 | 39 | 283237312c323129 | |

| | | |
|----------|---|------------------|
| 00668752 | 30 61 32 38 33 32 33 37 33 31 32 63 33 32 33 32 | 0a283237312c3232 |
| 00668768 | 32 39 30 61 32 38 33 32 33 37 33 31 32 63 33 32 | 290a283237312c32 |
| 00668784 | 33 33 32 39 30 61 32 38 33 32 33 37 33 31 32 63 | 33290a283237312c |
| 00668800 | 33 32 33 34 32 39 30 61 32 38 33 32 33 37 33 31 | 3234290a28323731 |
| 00668816 | 32 63 33 32 33 35 32 39 30 61 32 38 33 32 33 37 | 2c3235290a283237 |
| 00668832 | 33 31 32 63 33 32 33 36 32 39 30 61 32 38 33 32 | 312c3236290a2832 |
| 00668848 | 33 37 33 31 32 63 33 32 33 37 32 39 30 61 32 38 | 37312c3237290a28 |
| 00668864 | 33 32 33 37 33 31 32 63 33 32 33 38 32 39 30 61 | 3237312c3238290a |
| 00668880 | 32 38 33 32 33 37 33 31 32 63 33 32 33 39 32 39 | 283237312c323929 |
| 00668896 | 30 61 32 38 33 32 33 37 33 31 32 63 33 33 33 30 | 0a283237312c3330 |
| 00668912 | 32 39 30 61 32 38 33 32 33 37 33 31 32 63 33 33 | 290a283237312c33 |
| 00668928 | 33 31 32 39 30 61 32 38 33 32 33 37 33 31 32 63 | 31290a283237312c |
| 00668944 | 33 33 33 32 32 39 30 61 32 38 33 32 33 37 33 31 | 3332290a28323731 |
| 00668960 | 32 63 33 33 33 33 32 39 30 61 32 38 33 32 33 37 | 2c3333290a283237 |
| 00668976 | 33 31 32 63 33 33 33 34 32 39 30 61 32 38 33 32 | 312c3334290a2832 |
| 00668992 | 33 37 33 31 32 63 33 33 33 35 32 39 30 61 32 38 | 37312c3335290a28 |
| 00669008 | 33 32 33 37 33 31 32 63 33 33 33 36 32 39 30 61 | 3237312c3336290a |
| 00669024 | 32 38 33 32 33 37 33 31 32 63 33 33 33 37 32 39 | 283237312c333729 |
| 00669040 | 30 61 32 38 33 32 33 37 33 31 32 63 33 33 33 38 | 0a283237312c3338 |
| 00669056 | 32 39 30 61 32 38 33 32 33 37 33 31 32 63 33 33 | 290a283237312c33 |
| 00669072 | 33 39 32 39 30 61 32 38 33 32 33 37 33 31 32 63 | 39290a283237312c |
| 00669088 | 33 34 33 30 32 39 30 61 32 38 33 32 33 37 33 31 | 3430290a28323731 |
| 00669104 | 32 63 33 34 33 31 32 39 30 61 32 38 33 32 33 37 | 2c3431290a283237 |
| 00669120 | 33 31 32 63 33 34 33 32 32 39 30 61 32 38 33 32 | 312c3432290a2832 |
| 00669136 | 33 37 33 31 32 63 33 34 33 33 32 39 30 61 32 38 | 37312c3433290a28 |
| 00669152 | 33 32 33 37 33 31 32 63 33 34 33 34 32 39 30 61 | 3237312c3434290a |
| 00669168 | 32 38 33 32 33 37 33 31 32 63 33 34 33 35 32 39 | 283237312c343529 |
| 00669184 | 30 61 32 38 33 32 33 37 33 31 32 63 33 34 33 36 | 0a283237312c3436 |
| 00669200 | 32 39 30 61 32 38 33 32 33 37 33 31 32 63 33 34 | 290a283237312c34 |
| 00669216 | 33 37 32 39 30 61 32 38 33 32 33 37 33 31 32 63 | 37290a283237312c |
| 00669232 | 33 34 33 38 32 39 30 61 32 38 33 32 33 37 33 31 | 3438290a28323731 |
| 00669248 | 32 63 33 34 33 39 32 39 30 61 32 38 33 32 33 37 | 2c3439290a283237 |
| 00669264 | 33 31 32 63 33 35 33 30 32 39 30 61 32 38 33 32 | 312c3530290a2832 |
| 00669280 | 33 37 33 31 32 63 33 35 33 31 32 39 30 61 32 38 | 37312c3531290a28 |

https://blog.csdn.net/m0_46296905

随便找了一串ASCII编码一下，发现输出坐标，推测应该是填充坐标得到flag，这就是所谓的画图吧



https://blog.csdn.net/m0_46296905

```
import binascii
import matplotlib.pyplot
import numpy
f=open('meihuai.jpg','rb').read()
f=f[6+21232:] #获取f从6+21232往后的代表坐标的字节型数据
f=binascii.unhexlify(f.decode()).decode()
f=f.split('\n') #以换行符划分字符串放入列表
o=open('out','w') #创建一个文件
for i in f:
    o.write(i.strip('(')+'\n')#每一个代表坐标的字符串元素去掉括号之后添加换行符写入文件
o.close()
x,y=numpy.loadtxt('out',delimiter=',',unpack=True)#获取坐标数据
matplotlib.pyplot.plot(x,y, '.')
matplotlib.pyplot.show()
```

得到一个二维码



得到flag

```
flag{40fc0a979f759c8892f4dc045e28b820}
```

0x04 菜刀666

题目:

流量分析, 你能找到flag吗

0x05 [BJDCTF2020]认真你就输了

是个zip压缩包, 还看到了flag所在的路径

The screenshot shows the WinHex interface with a zip file named '10.zip' open. The main window displays a hex dump of the file's contents. The right pane shows the corresponding ASCII text. A red box highlights the path '/charts/flag.txt' in the ASCII view, which is located at offset 00000032. The hex dump shows the following data:

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI | ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------|-------------|
| 00000000 | 50 | 4B | 03 | 04 | 14 | 00 | 00 | 00 | 08 | 00 | D4 | 99 | 0B | 49 | 1D | 3E | PK | Ô™ I > |
| 00000016 | EA | 91 | 13 | 00 | 00 | 00 | 11 | 00 | 00 | 00 | 12 | 00 | 00 | 00 | 78 | 6C | è` | xl |
| 00000032 | 2F | 63 | 68 | 61 | 72 | 74 | 73 | 2F | 66 | 6C | 61 | 67 | 2E | 74 | 78 | 74 | /charts/flag.txt | |
| 00000048 | 4B | CB | 49 | 4C | AF | F6 | B5 | 4C | 0D | 4B | CB | 34 | 0A | 48 | 2E | 56 | KEII | òµL KE4 H.V |
| 00000064 | AE | 05 | 00 | 50 | 4B | 03 | 04 | 0A | 00 | 00 | 00 | 00 | 00 | AC | 65 | 5C | Ⓔ PK | -e\ |
| 00000080 | 46 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 09 | 00 | 00 | F | |
| 00000096 | 00 | 64 | 6F | 63 | 50 | 72 | 6F | 70 | 73 | 2F | 50 | 4B | 03 | 04 | 14 | 00 | docProps/PK | |
| 00000112 | 00 | 00 | 08 | 00 | 00 | 00 | 21 | 00 | 83 | 6C | B2 | 07 | 94 | 01 | 00 | 00 | ! fl² " | |
| 00000128 | 68 | 03 | 00 | 00 | 10 | 00 | 00 | 00 | 64 | 6F | 63 | 50 | 72 | 6F | 70 | 73 | h | docProps |
| 00000144 | 2F | 61 | 70 | 70 | 2E | 78 | 6D | 6C | 9D | 93 | CD | 6E | DB | 30 | 10 | 84 | /app.xml "ínŪ0 " | |
| 00000160 | EF | 05 | FA | 0E | 02 | EF | 31 | E5 | C4 | 28 | 0A | 63 | C5 | A0 | B1 | 5B | i ú ilåÄ(cÅ ±[| |
| 00000176 | E4 | 90 | A2 | 06 | 24 | 27 | E7 | 35 | B5 | B2 | 88 | 50 | A4 | 40 | 32 | 82 | ä ç \$'ç5µ²^P²²02, | |
| 00000192 | DD | A7 | 2F | 15 | 41 | B6 | 9C | F8 | D4 | DB | EE | CE | 60 | F8 | F1 | 0F | ý\$/ AŹæŌŪif'øñ | |
| 00000208 | EE | 0F | 8D | 4E | 3A | 72 | 5E | 59 | 93 | B1 | F9 | 2C | 65 | 09 | 19 | 69 | i N:r^Y"±ù,e i | |
| 00000224 | 4B | 65 | F6 | 19 | DB | 16 | BF | 6E | BE | B3 | C4 | 07 | 34 | 25 | 6A | 6B | Keö Ū ;n³²Å 4%jk | |
| 00000240 | 28 | 63 | 47 | F2 | EC | 5E | 7C | FD | 02 | 1B | 67 | 5B | 72 | 41 | 91 | 4F | (cGòì^ ý g[rA'Ō | |
| 00000256 | 62 | 84 | F1 | 19 | AB | 43 | 68 | 97 | 9C | 7B | 59 | 53 | 83 | 7E | 16 | 65 | b,,ñ «Ch-æ{YSf~ e | |
| 00000272 | 13 | 95 | CA | BA | 06 | 43 | 6C | DD | 9E | DB | AA | 52 | 92 | D6 | 56 | BE | •Ê° ClÝžŪ²R'ÖV³ | |
| 00000288 | 35 | 64 | 02 | BF | 4D | D3 | 6F | 9C | 0E | 81 | 4C | 49 | E5 | 4D | 7B | 0A | 5d ;Móœ LIåM | |
| 00000304 | 64 | 43 | E2 | B2 | 0B | FF | 1B | 5A | 5A | D9 | F3 | F9 | E7 | E2 | D8 | C6 | dCâ² ý ZZŪòùçâØÆ | |
| 00000320 | 3C | 01 | 3F | DA | 56 | 2B | 89 | 41 | 59 | 23 | 7E | 2B | E9 | AC | B7 | 55 | < ?ŪV+%AY#~+é·U | |
| 00000336 | 48 | 7E | 1E | 24 | 69 | E0 | 53 | 11 | 62 | 50 | 4E | F2 | CD | A9 | 70 | 14 | H~ \$iàs bPNðÍ@p | |
| 00000352 | 29 | F0 | 69 | 0B | B9 | 44 | 4D | AB | 18 | 2C | 2A | D4 | 9E | 80 | 9F | 07 |)ðì ±DM« ,*ôžēŸ | |
| 00000368 | F0 | 48 | D8 | 1F | DA | 06 | 95 | F3 | 02 | BA | B0 | EC | 48 | 06 | EB | 12 | ðHØ Ū •ó °°iH è | |
| 00000384 | AF | FE | C6 | 63 | BB | 65 | C9 | 0E | 3D | F5 | 38 | 19 | EB | D0 | 29 | 34 | ~bÆc>eÉ =ç8 eð)4 | |
| 00000400 | 81 | 0D | B6 | A1 | 79 | AF | 75 | EB | 83 | 13 | 2F | D6 | BD | FA | 9A | 28 | Ź;ÿ_üef /Ō²úš(| |
| 00000416 | 78 | E0 | A7 | 21 | F0 | 0F | DE | 69 | AD | 16 | 62 | 01 | 7C | 28 | 2E | 8D | xà\$!ð ði- b (. | |
| 00000432 | FC | 04 | 12 | EB | 4B | C4 | 42 | 05 | 4D | FE | 4F | B5 | 41 | 17 | AE | 10 | ü eKÅB MpOµA Ⓔ | |
| 00000448 | 2F | A6 | C4 | EF | 0C | 6C | C2 | 98 | F7 | 7C | C9 | 7C | 0A | 78 | AA | 9E | /;Åi lÅ~÷ É x²ž | |
| 00000464 | 94 | A1 | AB | 42 | 5C | 09 | 77 | 56 | E3 | 55 | B1 | C0 | 9D | A6 | CF | 3B | " ;«B\ wVåU±À ; | |
| 00000480 | 1E | D9 | 3F | D0 | AE | 6C | D3 | A2 | 39 | 8A | A7 | 87 | 6D | BE | 06 | 3E | ù?ðølóç9šš±m³ > | |
| 00000496 | B6 | 10 | D7 | 7E | F5 | DB | B6 | B0 | 6B | 0C | 34 | DE | D2 | E5 | 10 | F2 | Ź x~çŪŹ°k 4Pòå ò | |
| 00000512 | 1A | 1D | 95 | F1 | 62 | 47 | FD | 3C | 80 | C7 | B8 | 5D | A7 | 7B | FF | AA | •ñbGý<eç,]Ź{ý² | |
| 00000528 | 46 | B3 | A7 | 72 | F4 | 7C | 16 | FA | 37 | F5 | 3C | 7C | 1C | 31 | 5F | CC | F²\$Źø úŹç< 1_İ | |
| 00000544 | D2 | BB | 34 | 05 | 3E | 99 | 01 | 3F | 7F | 11 | F1 | 0F | 50 | 4B | 03 | 04 | ò»4 >™ ? ñ PK | |
| 00000560 | 14 | 00 | 00 | 00 | 08 | 00 | 00 | 00 | 21 | 00 | AD | A7 | 95 | ED | 42 | 01 | ! -S·iB | |
| 00000576 | 00 | 00 | 60 | 02 | 00 | 00 | 11 | 00 | 00 | 00 | 64 | 6F | 63 | 50 | 72 | 6F | docPro | |

解压后进入xl的charts文件夹, 找到flag

```
flag{M9eVfi2Pcs#}
```

0x06 被偷走的文件

题目:

一黑客入侵了某公司盗取了重要的机密文件，还好管理员记录了文件被盗走时的流量，请分析该流量，分析出该黑客盗走了什么文件。

0x07 [GXYCTF2019]佛系青年

是个伪加密，修改后保存

| | | |
|----------|---|-----------------|
| 00034720 | 24 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00 | \$ |
| 00034736 | 31 2E 70 6E 67 0A 00 20 00 00 00 00 00 01 00 18 | 1.png |
| 00034752 | 00 64 2B 49 FB 5B 94 D5 01 B6 55 39 FB 5B 94 D5 | d+Iû["õ ¶U9û["õ |
| 00034768 | 01 23 22 26 FB 5B 94 D5 01 50 4B 01 02 1F 00 14 | #"&û["õ PK |
| 00034784 | 00 00 00 08 00 51 AB 65 4F 83 26 AB 0C 02 03 00 | Q«eOf&« |
| 00034800 | 00 14 0B 00 00 06 00 24 00 00 00 00 00 00 00 20 | \$ |
| 00034816 | 00 00 00 5C 84 00 00 66 6F 2E 74 78 74 0A 00 20 | \,, fo.txt |
| 00034832 | 00 00 00 00 00 01 00 18 00 6E A4 82 A5 DC 93 D5 | nα,¥Ü"õ |
| 00034848 | 01 F6 B6 E1 51 DC 93 D5 01 F6 B6 E1 51 DC 93 D5 | ø¶áQÜ"õ ø¶áQÜ"õ |
| 00034864 | 01 50 4B 05 06 00 00 00 00 02 00 02 00 AF 00 00 | PK |
| 00034880 | 00 82 87 00 00 00 00 | ,‡ |

https://blog.csdn.net/M10_46296905

得到fo.txt内容如下

```

_oO0o_
o8888888o
88" . "88
(| -_- |)
O\ = /O
___/^-___\___
. ' \\\ |// \.
/ \\\||| : |||// \
/ _||| | -:- ||||- \
| | \\\ - // | |
| \_| "'\---/' | |
\ .-\_ \`-` ___/- . /
__` . .' /--- --\ ` . . __
.'"' '< \_ \<|>/_ . ' >'""
| | : \- \`. ; \_ / ; . \ / - ` : | |
\ \ \`- . \_ \ /_ /_ . -` //
===== \`- . \_ \ /_ . -` ____ . -' =====
`-----'
.....
佛祖保佑                永无BUG
写字楼里写字间，写字间里程序员；
程序人员写程序，又拿程序换酒钱。
酒醒只在网上坐，酒醉还来网下眠；
酒醉酒醒日复日，网上网下年复年。
但愿老死电脑间，不愿鞠躬老板前；
奔驰宝马贵者趣，公交自行程序员。
别人笑我忒疯癫，我笑自己命太贱；
不见满街漂亮妹，哪个归得程序员？

```

佛曰：遮等諳勝能礙幡藐哆娑梵迦徑羅迦梵者梵楞蘇涅侄室實真鉢朋能。奢怛俱道怯都諳怖梵尼怯一罰心鉢謹鉢薩苦奢夢怯帝梵遠朋陀諳陀穆諳所訥知涅侄以薩怯想夷奢醞數羅怯諸

是与佛论禅加密，在线网站解密得到flag，

与佛论禅

```
flag{w0 fo ci Be1}
```

听佛说宇宙的真谛

参悟佛所言的真意

普度众生

无穷般若心自在，语默动静体自然

佛曰：遮等語勝能礙暗藐哆娑梵迦侄羅哆迦梵者梵楞蘇涅侄室實鉢朋能。奢但俱道怯都語怖梵尼怯一罰心鉢謹鉢薩苦奢夢怯帝梵遠朋陀諳陀穆語所訥知涅侄以薩怯想夷奢醞數羅怯諸

https://blog.csdn.net/m0_46296905

```
flag{w0_fo_ci_Be1}
```

0x08 [BJDCTF2020]藏藏藏

这题要用 `foremost` 提取隐写文件，linux里安装一下，

```
sudo apt-get foremost
```

待提取文件所在目录下命令行输入

```
foremost 藏藏藏.jpg
```

生成了一个output文件夹，



提取出内容得到flag



```
flag{you are the best!}
```

0x09 秘密文件

0x0A [BJDCTF2020]你猜我是个啥

winhex打开, 是个png的图片,

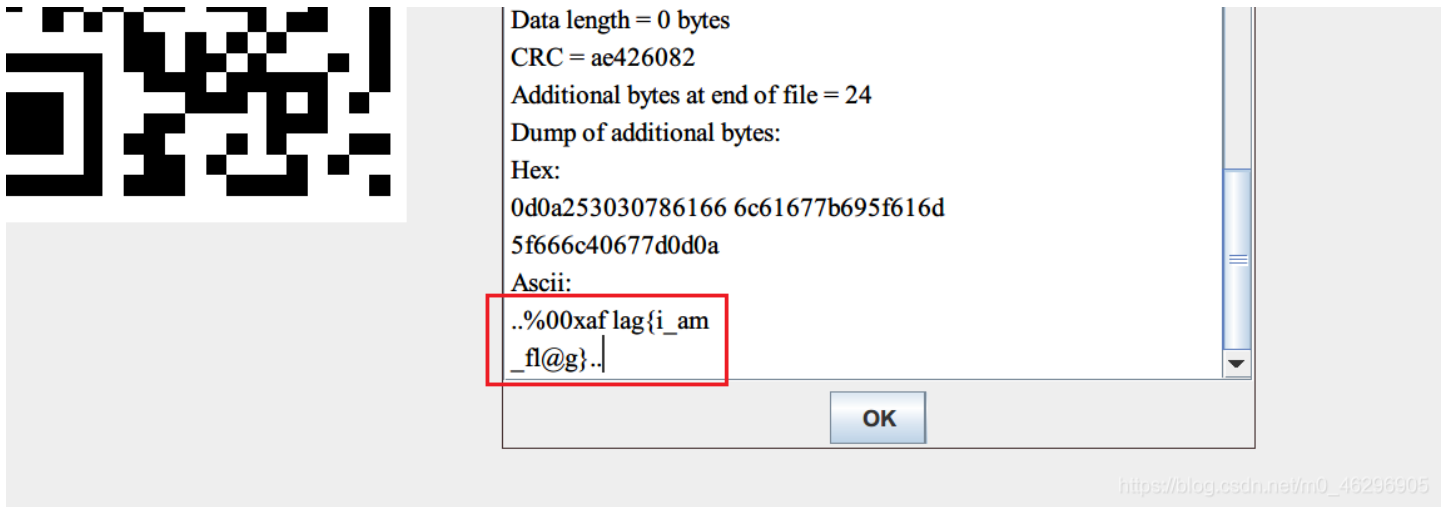
| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI | ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|--------------|
| 00000000 | 89 | 50 | 4E | 47 | 0D | 0A | 1A | 0A | 00 | 00 | 00 | 0D | 49 | 48 | 44 | 52 | PNG | IHDR |
| 00000016 | 00 | 00 | 00 | F5 | 00 | 00 | 01 | 00 | 08 | 06 | 00 | 00 | 00 | 6B | 99 | 30 | ø | k™0 |
| 00000032 | 7B | 00 | 00 | 04 | 7C | 49 | 44 | 41 | 54 | 78 | 9C | ED | DD | 41 | 6E | 1B | { | IDATxœiYAn |
| 00000048 | 31 | 14 | 05 | C1 | 28 | C8 | FD | AF | EC | C4 | 17 | 10 | 44 | 60 | 7E | 48 | 1 | Á(Èý~iÄ D`~H |
| 00000064 | B6 | AB | B6 | 01 | EC | 89 | A4 | 06 | 37 | CF | D4 | EB | EB | 9F | 5F | 40 | ¶«¶ i%¶ 7ÏÖëëY_@ | |
| 00000080 | C6 | EF | DD | 0F | 00 | 3C | 4B | D4 | 10 | 23 | 6A | 88 | 11 | 35 | C4 | 88 | ÆiY <KÖ #j^ 5Ä^ | |
| 00000096 | 1A | 62 | 44 | 0D | 31 | A2 | 86 | 18 | 51 | 43 | 8C | A8 | 21 | 46 | D4 | 10 | bd 1ç† QCE"!FÖ | |
| 00000112 | 23 | 6A | 88 | 11 | 35 | C4 | FC | 79 | F7 | 8F | AF | D7 | EB | 7F | 3D | 47 | #j^ 5Äü÷ ~xë =G | |
| 00000128 | C2 | CA | DF | C6 | 4C | BD | B6 | 53 | 7F | 9F | B3 | F2 | BC | 27 | BC | 0E | ÄÊBÆL»¶S Y³ò¼'¼ | |
| 00000144 | 55 | 9F | BC | B6 | 4E | 6A | 88 | 11 | 35 | C4 | 88 | 1A | 62 | 44 | 0D | 31 | UÝ¼¶Nj^ 5Ä^ bd 1 | |
| 00000160 | A2 | 86 | 18 | 51 | 43 | 8C | A8 | 21 | 46 | D4 | 10 | 23 | 6A | 88 | 11 | 35 | ç† QCE"!FÖ #j^ 5 | |
| 00000176 | C4 | BC | 9D | 89 | AE | 28 | 5F | 1F | 3E | 31 | 65 | 3C | 61 | CE | B9 | 62 | Ä¼ %@(_ >1e<aI'b | |
| 00000192 | F7 | FB | BB | FB | F7 | 4F | 7A | FA | 3D | 73 | 52 | 43 | 8C | A8 | 21 | 46 | ÷ú»ú÷Ozú=sRCE"!F | |
| 00000208 | D4 | 10 | 23 | 6A | 88 | 11 | 35 | C4 | 88 | 1A | 62 | 44 | 0D | 31 | A2 | 86 | Ö #j^ 5Ä^ bd 1ç† | |
| 00000224 | 18 | 51 | 43 | 8C | A8 | 21 | E6 | B1 | 99 | E8 | 8A | 13 | 6E | 90 | DC | 3D | QCE"!æ±™ëŠ n Ü= | |
| 00000240 | 3B | BC | 6D | CE | 39 | 75 | 9B | E8 | 84 | 9F | FE | F9 | 72 | 52 | 43 | 8C | ;¼mI9u>è„YpùrRCE | |
| 00000256 | A8 | 21 | 46 | D4 | 10 | 23 | 6A | 88 | 11 | 35 | C4 | 88 | 1A | 62 | 44 | 0D | "!FÖ #j^ 5Ä^ bd | |
| 00000272 | 31 | A2 | 86 | 18 | 51 | 43 | 8C | A8 | 21 | 66 | CB | 4C | 94 | B9 | 2F | 66 | 1ç† QCE"!fËL"/f | |
| 00000288 | BF | 6D | 7E | CA | F3 | 9C | D4 | 10 | 23 | 6A | 88 | 11 | 35 | C4 | 88 | 1A | ¿m~ËóœÖ #j^ 5Ä^ | |
| 00000304 | 62 | 44 | 0D | 31 | A2 | 86 | 18 | 51 | 43 | 8C | A8 | 21 | 46 | D4 | 10 | 23 | bd 1ç† QCE"!FÖ # | |

改一下后缀，提取二维码内容，没找到flag，



Stegsolve打开分析file format文件格式，或者直接winhex下拉到最后看字符，可以看到一个像flag的，提交后发现是正确的flag，

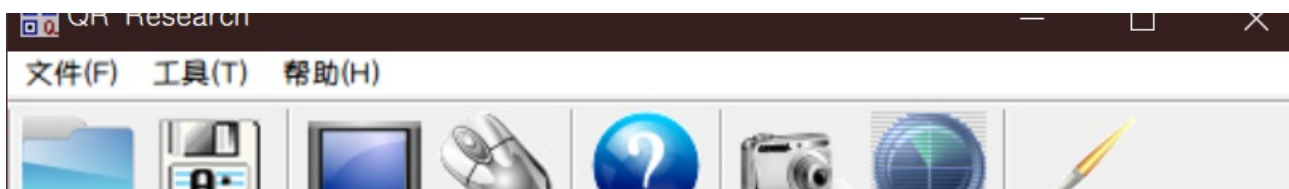
flag{i_am_fl@g}



0x0B [SWPU2019]神奇的二维码



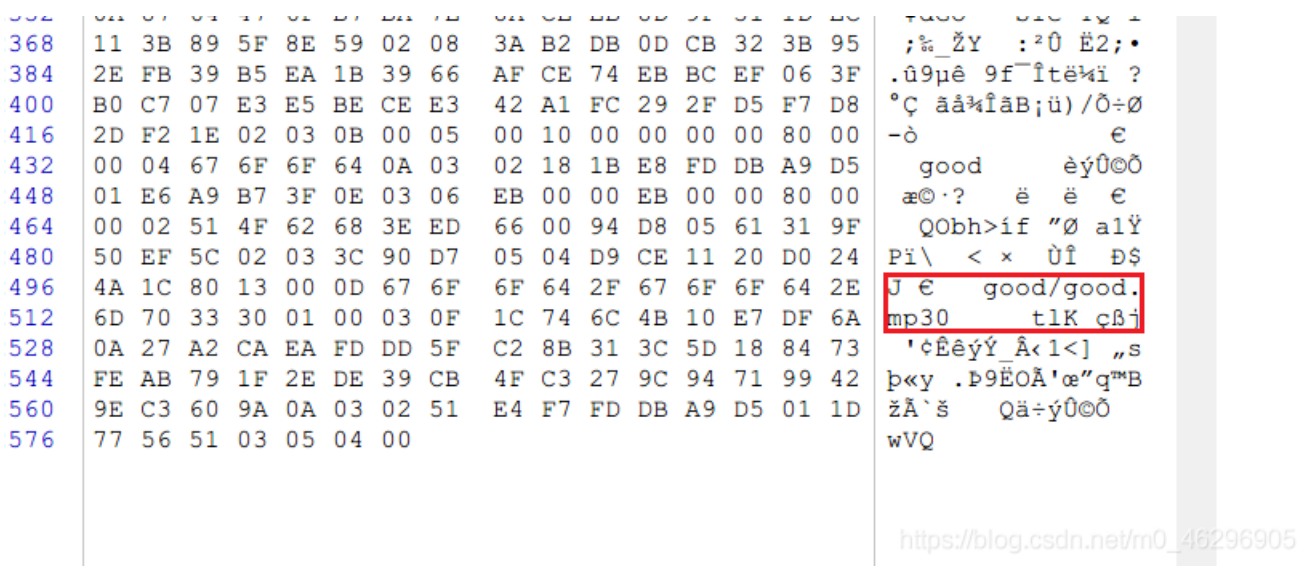
二维码里没找到flag，





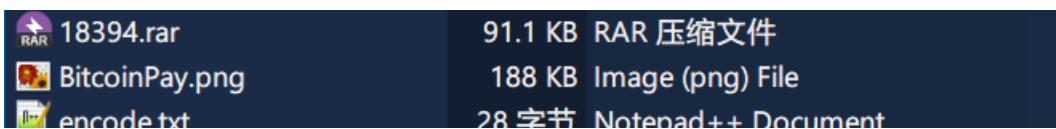
https://blog.csdn.net/m0_46296905

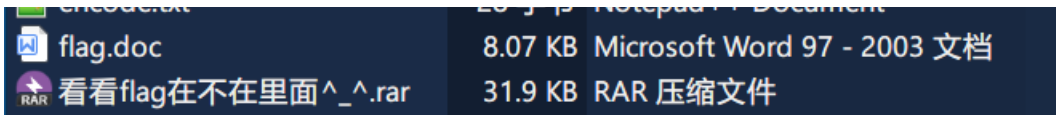
winhex打开，发现隐写了文件



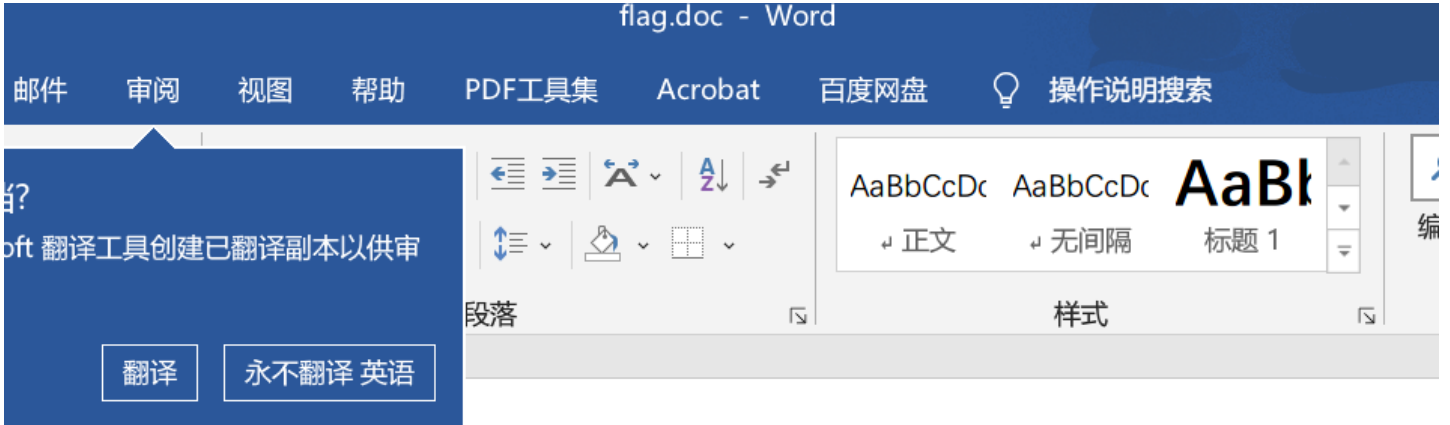
https://blog.csdn.net/m0_46296905

用 binwalk 提取出来，罗列提取出的文件如下





提取出的文件中有个经过多次base64编码后的文档，flag.doc，



Vm0wd2QyUX1VWGxWV0d4V1YwZDRWMV13WkRSV01WbDNXa1JTVjAxV2JET1hhMUpUVmpBeF
YySkVUbGhoTVVwVVZtcEJ1R115U2tWWJHaG9UV1Z3V1ZadGNFSmxSbGw1VTJ0V1ZXSkhh
Rz1VVmxaM1ZsWmFjVkJZOUmxSTmJFcEpWbTEwYTFkSFNrZGpSVGxhVmpOU1IxcFZXbUZrUj
A1R1UyMTRVMkpIZHpGV1Zfb3dWakZhV0Z0cmFHaFN1bXhXVm1wT1QwMHhjR1pYY1Vac1Vq
QTFSMWRyV25kV01ERkZVbFJHVjFaRmIzZFdha1poVjBaT2NtRkhhRk5sY1hoWFZtMXdUMV
F3TUhoa1JscF1ZbFZhY1ZadGRHRk5SbFowW1VaT1ZXS1ZXVEpWYkZKSfZqRmFSbU16WkZk
aGExcG9WakJhVDJ0dFJraGhSazVzWWxob1dGwnRNWGRVTZGM1RvaG9hbEpzY0ZsWmJGwM
hZMnhXY1ZGVVJsTk5XRUpIVmpKNFQxW1hTa2RqUmxxWF1saFNNMvpxU2t0V1ZrcFpXa1pr
YUdFeGNgbFhhMVpoVkrRkTmVgcE1UbWhTTW5oVvDWuk9RMWRzV1hoWGJYUk9VakZHT1ZaWE
5V0WhiRXAwVld4c1dtSkdXbWhaTW5oWF16R1djbHBHWkdsU2JrSmFWMnhXWVZReFdsaFRi
RnBZVmtWd1YxbHJXa3RUUmxxweFUydgFiR1pzV2xwWgExcHJZVWRGZUd0R2JGaGhNVnBvVm
tSS1QyTX1Ua1phUjJoVFRXNW9WV1pHWTNoaU1rbDRWMWhvWVZKR1NtR1diWGh6VFRGU1Zt
RkhPV2hpU1hCN1dUQmFjMWR0U2tkWGJXaGFUVzVvV0ZreFdrZFdWa3B6VkdzMVYySkdhM2
hXYTFwaFZUR1Z1RmR1U2s1WFJYQnhWVzB4YjFZeFVsaE9WazVPVFZad2VGvX1kREJXTVZw
eVkwWndXROV4YOROV2FrWkxWakpPU1dKR1pGZFNWWEJ2Vm10U1MxUX1UWGxVYTFwb1VqTk
NWRmxZY0ZkWFZscF1ZMFU1YVUxcmJEUldNV2h2V1ZaS1IxTnNaR1ZXyKzWn1ZHeGFZVmRG
T1ZaUFZtaFRUVWhDU2xac1pEUmpNV1IwVTJ0a1dHS1hhROZVVnpWd1YwWnJ1RmRyWkZkV2

不知道编码了几次，那就直接写个exp，循环100次解码，解到不能解的地方会自动报错停下来，

```
import base64
flag='Vm0wd2QyUX1VWGxWV0d4V1YwZDRWMV13WkRSV01WbDNXa1JTVjAxV2JET1hhMUpUVmpBeFYySkVUbGhoTVVwVVZtcEJ1R115U2tWWJHaG9UV1Z3V1ZadGNFSmxSbGw1VTJ0V1ZXSkhhRz1VVmxaM1ZsWmFjVkJZOUmxSTmJFcEpWbTEwYTFkSFNrZGpSVGxhVmpOU1IxcFZXbUZrUjA1R1UyMTRVMkpIZHpGV1Zfb3dWakZhV0Z0cmFHaFN1bXhXVm1wT1QwMHhjR1pYY1Vac1VqQTFSMWRyV25kV01ERkZVbFJHVjFaRmIzZFdha1poVjBaT2NtRkhhRk5sY1hoWFZtMXdUMVF3TUhoa1JscF1ZbFZhY1ZadGRHRk5SbFowW1VaT1ZXS1ZXVEpWYkZKSfZqRmFSbU16WkZkaGExcG9WakJhVDJ0dFJraGhSazVzWWxob1dGwnRNWGRVTZGM1RvaG9hbEpzY0ZsWmJGwMhZMnhXY1ZGVVJsTk5XRUpIVmpKNFQxW1hTa2RqUmxxWF1saFNNMvpxU2t0V1ZrcFpXa1prYUdFeGNgbFhhMVpoVkrRkTmVgcE1UbWhTTW5oVvDWuk9RMWRzV1hoWGJYUk9VakZHT1ZaWE5V0WhiRXAwVld4c1dtSkdXbWhaTW5oWF16R1djbHBHWkdsU2JrSmFWMnhXWVZReFdsaFRiRnBZVmtWd1YxbHJXa3RUUmxxweFUydgFiR1pzV2xwWgExcHJZVWRGZUd0R2JGaGhNVnBvVm tSS1QyTX1Ua1phUjJoVFRXNW9WV1pHWTNoaU1rbDRWMWhvWVZKR1NtR1diWGh6VFRGU1ZtRkhPV2hpU1hCN1dUQmFjMWR0U2tkWGJXaGFUVzVvV0ZreFdrZFdWa3B6VkdzMVYySkdhM2hXYTFwaFZUR1Z1RmR1U2s1WFJYQnhWVzB4YjFZeFVsaE9WazVPVFZad2VGvX1kREJXTVZweVkwWndXROV4YOROV2FrWkxWakpPU1dKR1pGZFNWWEJ2Vm10U1MxUX1UWGxVYTFwb1VqTkNWRmxZY0ZkWFZscF1ZMFU1YVUxcmJEUldNV2h2V1ZaS1IxTnNaR1ZXyKzWn1ZHeGFZVmRG T1ZaUFZtaFRUVWhDU2xac1pEUmpNV1IwVTJ0a1dHS1hhROZVVnpWd1YwWnJ1RmRyWkZkV2
```



```
lVbFZ3VTFadE1IZGxSVFZJVWxob1YxZEhhRmxXTUdSd1ZqRnNjbHBIT1dwaVJswXpwMnRhVDFkR1NuT1RiR2hYVfVdw2NsWkh1RXRrUjFKR1ZHeG
9hRTFXy0hsV2FrSmhVMjFSZVZScldtaFNia0pQV1cwmVewMXNXbkZUYm5Cc1VtczFTRlP0T1ZkWFiwE1WV3M1V21KVJVuW1pha1pWTFaR2RGSn
NaRTVoZwXZML1YxUkNWMk14V1hsVGEyaFdZa2RvVmxadGVHRk5NVnBZW1Vkr2FRmV1Ra3BYTfWVfZHeGFwVkpVUwXv1JWcDjXWHBHVm1WV1NsBg
lSbHBwVmtkNFdGZfH1Rz1pTVZKSFYyNUTXR0pW25GVVYzUmhVakZhU0dWR1RsVm1SbkF4V1Zkd1UxwXhXalpS5WxKV11XdGfHrmt5YzNoV01XUn
lUbfPrVTJfE1FscFdiVEIzW1VksmVWwVubGhPyXpWw1dXeG9VMVpXVm5GUMJvW1VwBTE0VjFZeU1VZFdWMBHHTBSTR1ZsWjZRVEZXYWtwTFZsWk
tWvKzZy0d4aE0wS1JwMWh3UjJrERsZFVibEpyVw1zMVQxU1ZwbmRXyKzSnfDrUKNXbF14UmpOVWjGwNjWmGRlU0dGRk9WZGhNVnBNVmtSR1YyUk
hwa2xVYXpsVF1rZDNV1pIZUZaT1YwWk1VMnRhYwXkDGVHRldiRnAZk4YWNWtNJaR3BoZwXawVZsZDRhMV4V25WUmFscFhZbGhVYUZwVjtdF
hSa3B5V2taV2FWSXhTb1pXUmXKRFUyc3hJmWR1VW10U00xS1FWVzB4TkZkR1duTmhTRTVVWpCV05Ga3dXbk5XTURGSV1V1NWMDfHY0d0wMvRwN
JaR1p3UjFSck5WZGhNV3QzVm0xd1MwMudVWGHYymxKV1URndWmxyV25kV2JGcHpwMnRrVGsxV1draFZiRkp6V1ZaV1ZVMUvhejA9'
for i in range(100):
    print("第{0}次解码: ".format(i+1))
    flag=base64.b64decode(flag).decode()
    print(flag)
```

```
C:\WINDOWS\SYSTEM32\cmd.exe
alphVTFfeFdsFRhM1JwVWtaS1YxUlh0Vks5YkZweFvTMudVMkpWVmpaW1ZweGhZVWRGZUdOR2JGaFhTRUpJV1ZSS1QyTxhaSFZVYkZkEfvRktWV1pHV
第9次解码:
Vm0wd2QyUX1Wa2hVW0doVYyWZG9XR113Wkc5V2JHeDBaRWhrVmxKc2NEQ1VWbU0xVmpBeFdHvkdXbFp0Ym1oUVZtcEdZV1JIVmtsavJttuk9ZV3hhZVZac
Wa2MxVDJGR1NuUmhSemxWVm14YU0xWnNXbUZrULRGV1ZXeHdWMDfWY0ZsV1Z6QXhVekpHUjFOdVVsWm1hMBBZvkZWYwQxUkdjRmRYV1hSwwVqRktTVLZQ
ZtMTBWM1F5VW50V2JrNV1ZbFZhY2xWc1VrZFhIR3QzV2tSU1ZrMUVSa1pXY1hoM1ZqRmFSbU16WkZwV1JWcGhXbFphVDJ0c2NFZGhSMnhUVFcx1dsWX
kZaU1ExW1hTa2RpUkZKV1RXNVN1bFpxUm1GU2JVvJZzVvPhYudFeGNGbFhXSEJTWVRKT2MxZHVubFJpUjJfKVVZGukJkMDFsUFQwPQ==
第10次解码:
Vm0wd2QyVkhVWGHUV0doWfYwZG9WbGx0ZEhkV1JscDBUvM1VjAxWGVGW1ZNbmhQVmpGYWRHVk1iRmROYWxaeVZtMTRZV015VGtsa1JtU1haV3hhVZac
WVzAxUzJGR1NuU1Zia0pYVfVad1RGcFdXbXRYUjFKSVVteHdWMDfWY0ZsV1Z6QXhWRpHUjFOdVVsWm1SaBbXVm1OV2QyUnNwbk5YY1Vac1VsUkdXbGt3
YxWmtNRmxXVWtka1JttU1lZbGhTY1ZSV1pGTk5SbFowVfZSQ1ZXskdiRfJTW5Se1ZqRmFSbU6YUzAaGExcF1XWHBHytJ0c1duT1RiR1JUvFRBd01RPT0
第11次解码:
Vm0wd2VHUxhTWGhXV0doV11tdHdWR1p0Tvc5V01XeFZVMnhPVjFadGVIbFdNa1ZyVm14YwYmYTk1jRmRXZwXaUvZqQmtTMU14VG50aFJtaG9UV1Z3V1Zac
WVzAxVDJGR1NuU1ZiRkpWvmtWd2R5VnNXbUZrULRGV1kZDRVMDfVmxw1ZfB3dZVEZaZVZ0c1pGaG1SMmhZV1ZkMf1WUkdjRmRYV1hScVRWZFNRR1ZQ
==
第12次解码:
Vm0weGQxSxhWwGHVYmtwUFZtMw9WmXVU2x0V1ZteH1WmjVrVmxac2NIcFdWe1ZQVjBkS1IxTnNhRmhoTVVwVYzTMTRZV1JXUm5KaFJtUnBWMFpLZVZkV
WVEowYTFZeVNrZFhIR2hYwVd0YVRGcFdXbXRxQTVdSMFVtMTBubF14U2tsV1ZFa3hVakZXYzFkc1ZsS1dSM001
第13次解码:
Vm0xd1IxVXhUbkpPVm1oV11US1NWmxyV25kV1ZscHpWvZVPV0dKR1NsaFhhMUpUvM14YWRWRnJhRmRpV0ZKeVdXdGFZV1JXUm5OaFJtU1RUVEpvU1Zac
WVEkxUjFwC1drV1JWR3M5
第14次解码:
Vm1wR1UxTnJovmhWYTJSVv1rWndVv1pzVW50WgJGS1hXa1JTVmxadVFraFdiWFJyWwtaYWRWRnNhRmRTTJoRVZtMXp1R05YU2tkV2JGw1hWakZLVZkV
第15次解码:
VmpGU1NrNVhVa2RUyKzWUVZsUnNXbFJXWkRSV1ZuQkhWbXRRYkZadVFsaFdsM2hEvM1zeGNXSkdWbFZXVjFKeVdsWmFkMWRHU25GU1ZEQTk=
第16次解码:
VjFSsk5XUkdTbFpQV1RsW1RWZDRVnBHvmtkbFZuQ1hWR3hDVmSxcWJGV1VWV1JyW1Zad1dGSnFRVDA9
第17次解码:
V1RJNWRGS1ZPVt1ZTVd4UvpGVkd1VnBXVGxCVk1qbFVUvWRzRVZwWFJqQT0=
第18次解码:
WTI5dFJV0U9YMWxQZfVGeVpWT1BVMj1UTUdkeVpXRjA=
第19次解码:
V29tbnR0X11pDdUFyZVNPd29TMdyZWp0
第20次解码:
comEON_YOUAreS0SoS0great
第21次解码:
Traceback (most recent call last):
  File "E:\py_work\Re\test2.py", line 5, in <module>
    flag=base64.b64decode(flag).decode()
  File "D:\Python3.9\lib\base64.py", line 87, in b64decode
    return binascii.a2b_base64(s)
```

https://blog.csdn.net/m0_46296905

解码了20次，得到一个字符串

```
comEON_YOUAreS0SoS0great
```

还有一个是encode.txt

```
YXNkZmdoamtSMTIzNDU2Nzg5MA==
```

同样base64解码一下，

```
import base64
print(base64.b64decode('YXNkZmdoamtSMTIzNDU2Nzg5MA==').decode())
```

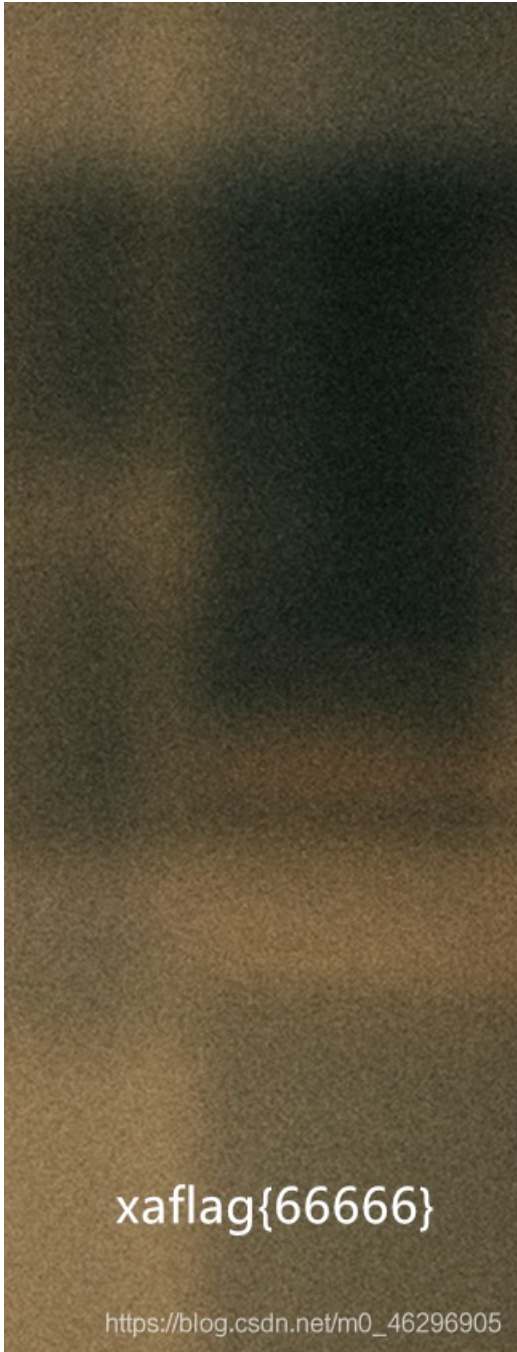
解码出来得到字符串

```
asdfghjkl1234567890
```


0x0C [BJDCTF2020]一叶障目

无法浏览的png文件，应该是CRC校验有误，网上找到的CRC修正python源码，

```
#coding=utf-8
import zlib
import struct
#读文件
file = '1.png' #注意，1.png图片和脚本在同一个文件夹下哦~
fr = open(file,'rb').read()
data = bytearray(fr[12:29])
crc32key = eval(str(fr[29:33]).replace('\x', '').replace("b'", '0x').replace("'", ''))
#crc32key = 0xCBD6DF8A #补上0x, copy hex value
#data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xf4\x00\x00\x01\xf1\x08\x06\x00\x00\x00') #hex下copy grep hex
n = 4095 #理论上0xffffffff,但考虑到屏幕实际, 0x0fff就差不多了
for w in range(n):#高和宽一起爆破
    width = bytearray(struct.pack('>i', w))#q为8字节, i为4字节, h为2字节
    for h in range(n):
        height = bytearray(struct.pack('>i', h))
        for x in range(4):
            data[x+4] = width[x]
            data[x+8] = height[x]
            #print(data)
        crc32result = zlib.crc32(data)
        if crc32result == crc32key:
            print(width,height)
            #写文件
            newpic = bytearray(fr)
            for x in range(4):
                newpic[x+16] = width[x]
                newpic[x+20] = height[x]
            fw = open(file+'.png', 'wb')#保存副本
            fw.write(newpic)
            fw.close
```



得到flag

```
flag{66666}
```

0x0D [BJDCTF2020]鸡你太美

得到两个gif文件，一个损坏的，一个完好的，

本能地想逐帧查看，但这题好像不是考察这个的，winhex打开看看

第二个少了个gif文件头，所以才显示损坏，直接复制那个完好的gif的文件头 47 49 46 38，光标移至开头，ctrl+v 粘贴，默认是插入

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI | ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-------------------|
| 00000000 | 47 | 49 | 46 | 38 | 39 | 61 | 68 | 01 | 80 | 02 | F7 | 00 | 31 | 00 | FF | 00 | GIF8 | 9ah € ÷ 1 ý |
| 00000016 | 04 | 04 | 0A | 0B | 09 | 14 | 0F | 0C | 1A | 15 | 21 | 2C | 15 | 2C | 3A | 16 | | !,,: |
| 00000032 | 10 | 1B | 16 | 11 | 21 | 17 | 16 | 21 | 19 | 17 | 28 | 1B | 17 | 23 | 1B | 17 | | ! ! (# |
| 00000048 | 28 | 1B | 1C | 2D | 1C | 19 | 2A | 1C | 1A | 2A | 1F | 19 | 2B | 20 | 16 | 24 | | (- * * + \$ |
| 00000064 | 20 | 1D | 29 | 20 | 1D | 2D | 21 | 1D | 31 | 23 | 20 | 35 | 24 | 22 | 37 | 24 | |) -! 1# 5\$"7\$ |
| 00000080 | 28 | 3C | 28 | 20 | 31 | 28 | 23 | 3B | 29 | 18 | 22 | 29 | 1D | 2C | 2A | 27 | | (<(1(#;) ") ,*' |
| 00000096 | 3E | 2B | 29 | 34 | 2D | 2C | 3F | 34 | 1F | 2B | 34 | 26 | 39 | 34 | 31 | 3A | | >+)4-,?4 +4&941: |
| 00000112 | 35 | 2D | 43 | 35 | 36 | 50 | 37 | 36 | 44 | 3C | 5B | 6A | 3E | 2B | 38 | 3F | | 5-C56P76D<[j]>+8? |
| 00000128 | 32 | 46 | 40 | 3A | 51 | 40 | 3D | 47 | 42 | 4A | 53 | 45 | 3E | 53 | 46 | 2C | | 2F@:Q@=GBJSE>SF, |
| 00000144 | 37 | 49 | 33 | 46 | 4C | 56 | 69 | 4D | 4B | 57 | 4E | 3C | 47 | 4F | 3E | 50 | | 7I3FLViMKWN<GO>P |
| 00000160 | 50 | 42 | 59 | 52 | 4F | 63 | 53 | 2F | 36 | 54 | 4D | 59 | 58 | 52 | 64 | 5F | | PBYROcS/6TMYXRd_ |
| 00000176 | 48 | 59 | 61 | 79 | 86 | 62 | 50 | 61 | 64 | 57 | 6D | 64 | 59 | 62 | 64 | 5A | | HYaytbPadWmdYbdZ |
| 00000192 | 68 | 67 | 61 | 6E | 68 | 41 | 42 | 69 | 61 | 74 | 6B | 65 | 7A | 73 | 58 | 5E | | hganhABiatkezsX^ |
| 00000208 | 73 | 74 | 7F | 74 | 80 | 96 | 75 | 5D | 6B | 75 | 62 | 68 | 76 | 67 | 79 | 76 | | st t@-u]kubhvgyv |
| 00000224 | 6B | 80 | 77 | 62 | 70 | 77 | 6C | 86 | 77 | 75 | 83 | 7A | 68 | 6E | 7B | 76 | | k@whnw1twifzbnlv |

得到flag



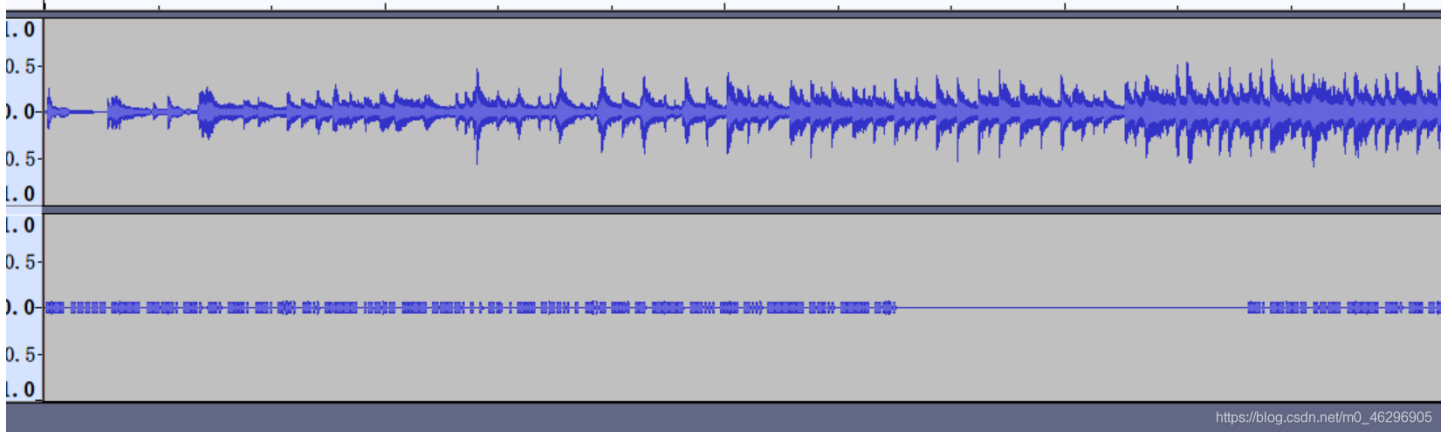
```
flag{zhi_yin_you_are_beautiful}
```

0x0E 穿越时空的思念

题目:

嫦娥当年奔月后，非常后悔，因为月宫太冷清，她想：早知道让后羿自己上来了，带了只兔子真是不理智。于是她就写了一首曲子，诉说的是怀念后羿在的日子。无数年后，小明听到了这首曲子，毅然决定冒充后羿。然而小明从曲子中听不出啥来，咋办。。（该题目为小写的32位字符，提交即可）

里面很明显嵌入了摩斯密码，Audacity 打开，解摩斯密码



```
flag{f029bd6f551139eedeb8e45a175b0786}
```

0x0F [BJDCTF2020]just_a_rar

是一个名为 4位数.rar 的压缩文件，ARCHPR 破解一下

ARCHPR 4.54 - 20%

文件(F) 恢复(R) 帮助(H)

打开 开始! 停止 基准测试 升级 帮助 关于 退出

加密的 ZIP/RAR/ACE/ARJ 文件 攻击类型

E:\CTF_project\BUUCTF\MISC\0524-[BJDCTF2020]just_a_rar.rar 暴力

口令已成功恢复!

Advanced Archive Password Recovery 统计信息:

| | |
|------------|-------------|
| 总计口令 | 765 |
| 总计时间 | 3s 771ms |
| 平均速度(口令/秒) | 202 |
| 这个文件的口令 | 2016 |
| 十六进制口令 | 32 30 31 36 |

保存... 确定

2021/5/24 21:02:51 - 口令已成功恢复!
2021/5/24 21:02:51 - '2016' 是这个文件的一个有效口令

当前口令: 2016 平均速度: 203 p/s
已用时间: 3s 剩余时间: 39s
口令长度 = 4, 总计: 10,000, 已处理: 2,021

20%

ARCHPR version 4.54 (c) 1997-2012 ElcomSoft Co. Ltd.



解压出来
右键查看属性得到flag





flag{Wadf_123}

0x10 [BJDCTF2020]纳尼

浏览不了的gif文件，winhex打开看看，发现是少了个文件头，加一下

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | ANSI ASCII |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| 00000000 | 47 | 49 | 46 | 38 | 39 | 61 | 80 | 04 | 88 | 02 | F7 | 00 | 00 | 02 | 02 | 02 | GIF89a |
| 00000016 | 0A | 01 | 02 | 00 | 09 | 01 | 01 | 02 | 0B | 0A | 02 | 0B | 02 | 0A | 09 | 09 | ^ |
| 00000032 | 08 | 0A | 11 | 00 | 00 | 12 | 00 | 0B | 11 | 08 | 03 | 02 | 02 | 12 | 0A | 01 | ÷ |
| 00000048 | 13 | 01 | 02 | 1B | 02 | 09 | 17 | 2E | 06 | 00 | 34 | 01 | 02 | 3A | 01 | 01 | |
| 00000064 | 3C | 00 | 0B | 38 | 0F | 13 | 01 | 02 | 2C | 00 | 0D | 23 | 01 | 02 | 33 | 08 | |
| 00000080 | 00 | 33 | 00 | 09 | 33 | 01 | 01 | 3C | 08 | 02 | 3C | 45 | 00 | 0D | 47 | 01 | |
| 00000096 | 03 | 5D | 02 | 02 | 59 | 05 | 05 | 4C | 12 | 04 | 64 | 01 | 01 | 6C | 00 | 01 | |
| 00000112 | 62 | 02 | 0A | 6E | 00 | 0D | 67 | 08 | 02 | 71 | 00 | 00 | 75 | 00 | 0D | 7E | |
| 00000128 | 43 | 1B | 02 | 03 | 42 | 02 | 04 | 4A | 06 | 04 | 46 | 03 | 01 | 59 | 01 | 02 | |
| 00000144 | 63 | 02 | 00 | 6C | 0B | 01 | 66 | 05 | 04 | 75 | 04 | 34 | 7E | 00 | 67 | 5D | |
| 00000160 | 64 | 65 | 46 | 8C | 35 | 02 | 8C | 3C | 01 | 88 | 3A | 09 | 93 | 35 | 00 | 94 | |
| 00000176 | 3B | 02 | 9A | 39 | 00 | 9F | 36 | 09 | 8A | 41 | 08 | B5 | 5F | 00 | 9E | 6C | |
| 00000192 | 00 | AD | 65 | 03 | AB | 6A | 00 | AE | 67 | 0A | B4 | 65 | 01 | BC | 66 | 02 | |
| 00000208 | B4 | 69 | 01 | BB | 6A | 02 | B7 | 67 | 0B | B2 | 68 | 15 | B6 | 71 | 0C | B2 | |
| 00000224 | 5F | 2B | B8 | 67 | 3A | C4 | 64 | 02 | C8 | 62 | 00 | C0 | 60 | 14 | DE | 8E | |
| 00000240 | 2B | CF | 8D | 36 | DB | 8C | 3A | D6 | 92 | 3D | DA | 92 | 35 | E4 | 8D | 2A | |
| 00000256 | E5 | 95 | 3A | DA | 8F | 45 | DC | 92 | 45 | D7 | 93 | 4A | E1 | 93 | 41 | FC | |
| 00000272 | B4 | 5B | FA | B8 | 59 | FF | AE | 65 | FF | AE | 6C | EC | BE | 69 | ED | B6 | |
| 00000288 | 66 | FD | B4 | 63 | FE | B8 | 63 | FD | B4 | 6A | FF | B9 | 6A | FE | B6 | 73 | |
| 00000304 | EA | B5 | 7F | EA | C2 | 7A | 01 | 3B | 8A | 01 | 38 | 85 | 01 | 3D | 92 | 02 | |

码文散布在不同的帧，用 PS 逐帧提取一下

```
Q1RGe3dhbmdfYmFvX3FpYW5nX21zX3NhZH0=
```

base64解码

```
import base64
print(base64.b64decode('Q1RGe3dhbmdfYmFvX3FpYW5nX21zX3NhZH0=').decode())
```

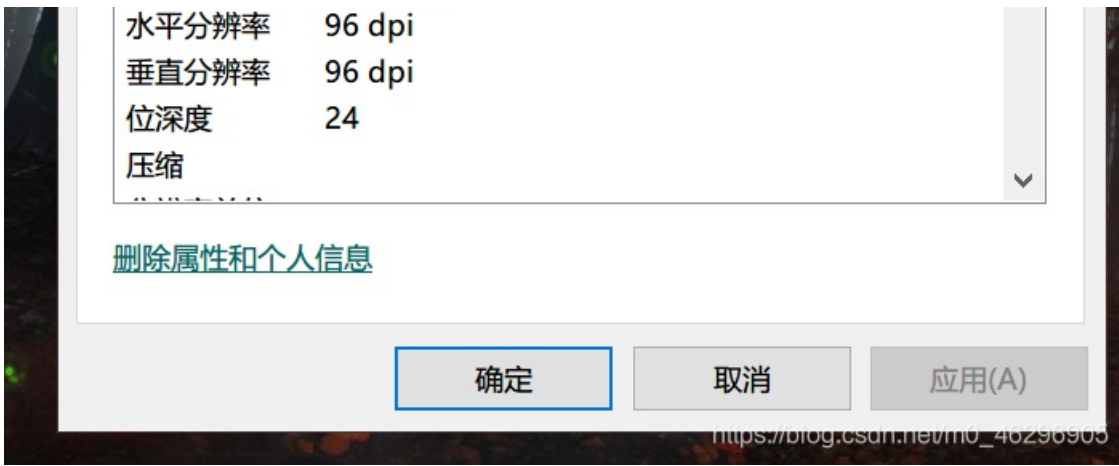


```
flag{wang_bao_qiang_is_sad}
```

0x11 [ACTF新生赛2020]outguess

主要是 `mmm.jpg` 右键看属性找到奇怪的备注，是 `核心价值观编码`





在线解码得到 `outguess` 隐写的 key

—— 核心价值观编码 ——

社会主义核心价值观：富强、民主、文明、和谐；自由、平等、公正、法治；爱国、敬业、诚信、友善

abc

编 码

解 码

公正民主公正文明公正和谐

outguess隐写

1.outguess工具安装

```
git clone https://github.com/crorvick/outguess
./configure && make && make install #无权限时加sudo
```

2.outguess工具基本使用

(1) 写入:

```
outguess -k "my secret key" -d hidden.txt demo.jpg out.jpg
#加密之后, demo.jpg会覆盖out.jpg,
#hidden.txt中的内容是要隐藏的东西
```

(2) 提取:

1. 当有key时:

```
outguess -k "my secret key" -r out.jpg hidden.txt
#解密之后, 解密内容放在hidden.txt中
```

2. 当无key时:

```
outguess -r out.jpg hidden.txt
```

(3) outguess --help

```
outguess [options] [<input file> [<output file>]]
  -[sS] <n>      iteration start, capital letter for 2nd dataset
  -[iI] <n>      iteration limit
  -[kK] <key>    key
  -[dD] <name>   filename of dataset
  -[eE]          use error correcting encoding
  -p <param>    parameter passed to destination data handler
  -r            retrieve message from data
  -x <n>        number of key derivations to be tried
  -m           mark pixels that have been modified
  -t           collect statistic information
  -F[+-]       turns statistical steganalysis foiling on/off.
```

这题求出 key 是 abc，就直接输入命令

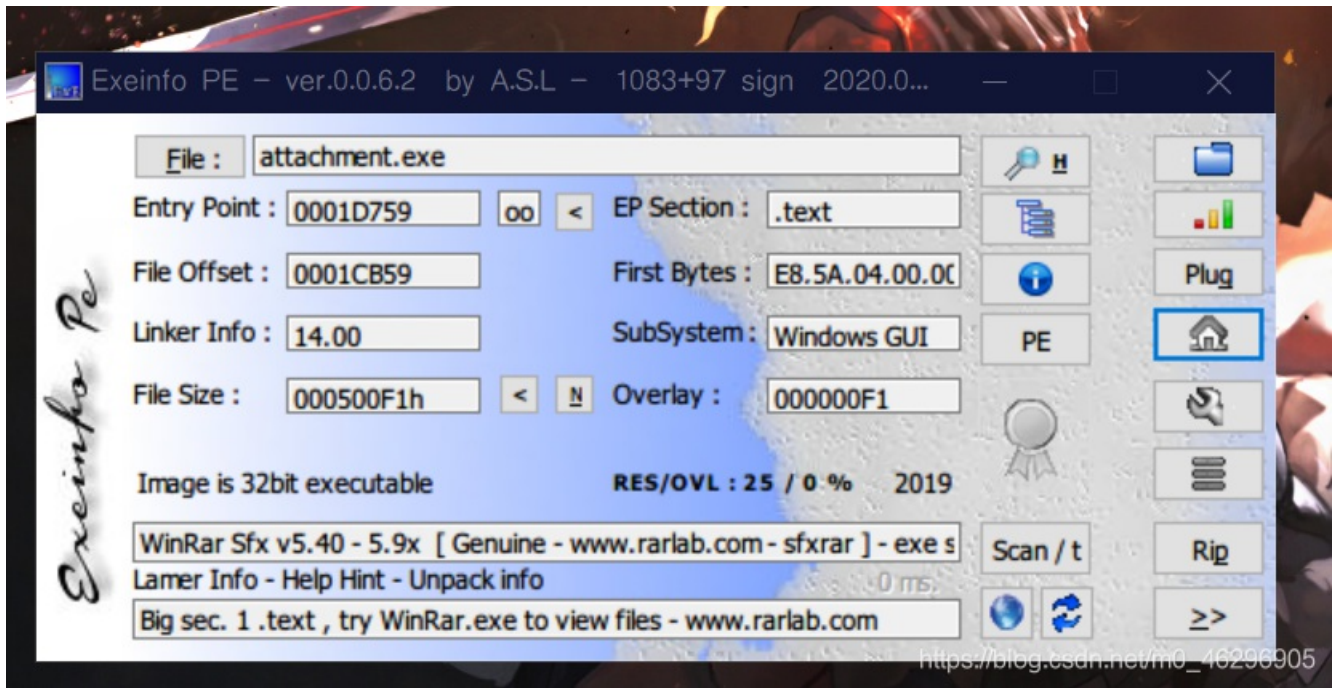
```
outguess -k "abc" -r mmm.jpg flag.txt
```

得到flag

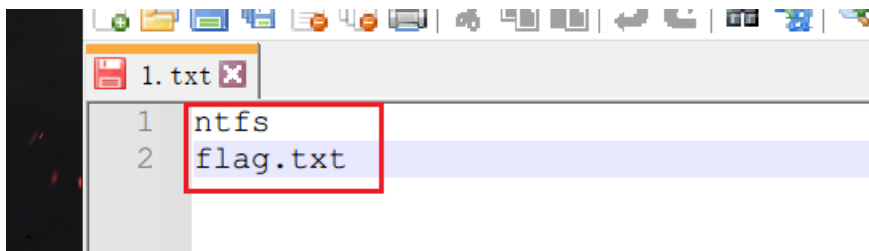
```
flag{gue33_Gu3Ss!2020}
```

0x12 [SWPU2019]我有一只马里奥

确认是一个exe文件



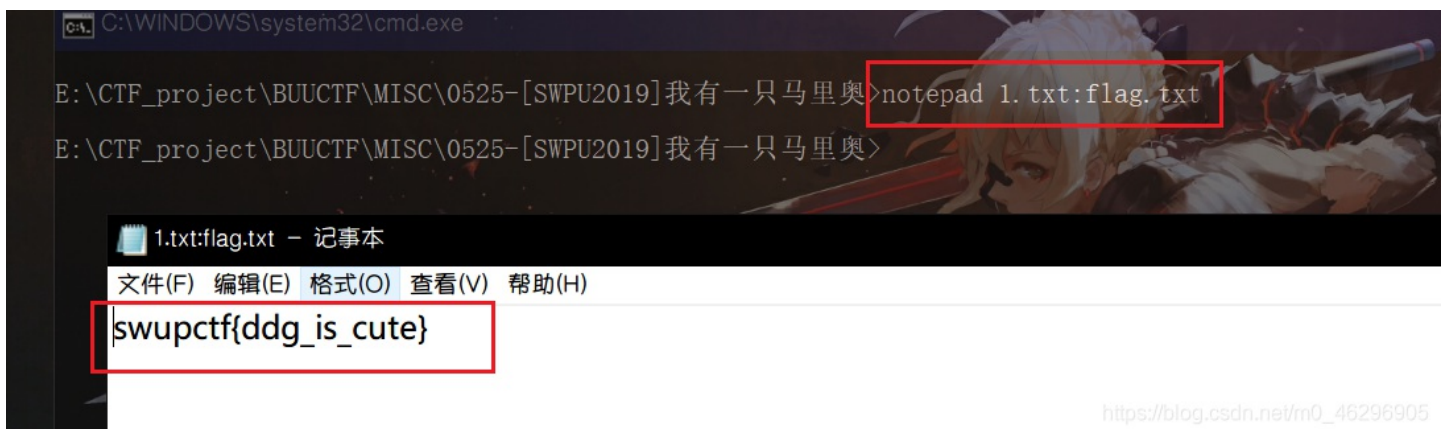
运行一下得到一个文本文件，告诉我们这题在1.txt里用 NTFS流隐写 了一个flag.txt



可以用工具NtfsStreamsEditor提取

也可以在1.txt所在目录下打开cmd输入:

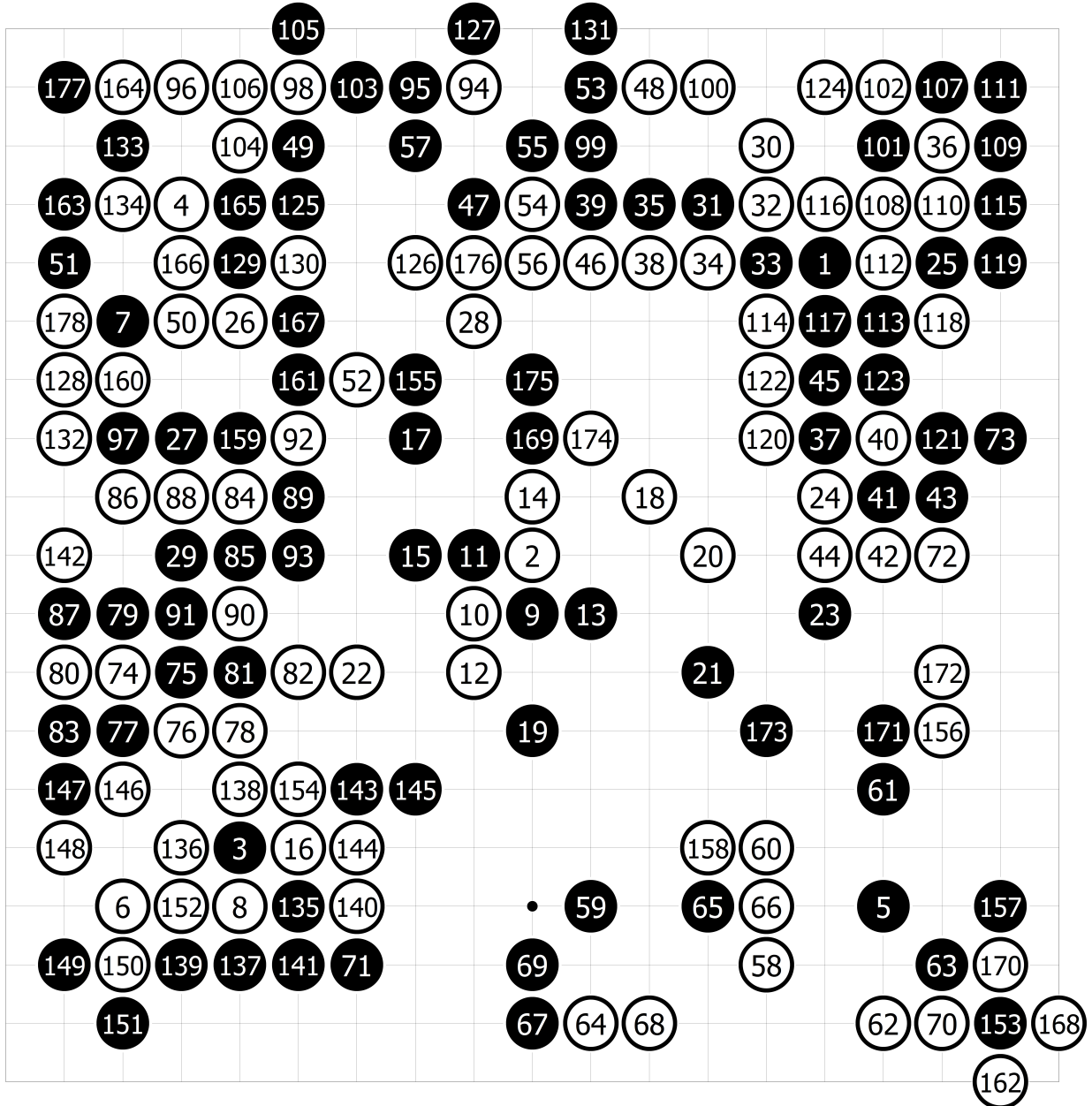
```
notepad 1.txt:flag.txt
```



```
flag{ddg_is_cute}
```

0x13 谁赢了比赛？

题目：小光非常喜欢下围棋。一天，他找到了一张棋谱，但是看不出到底是谁赢了。你能帮他看看到底是谁赢了么？



foremost提取一下

```

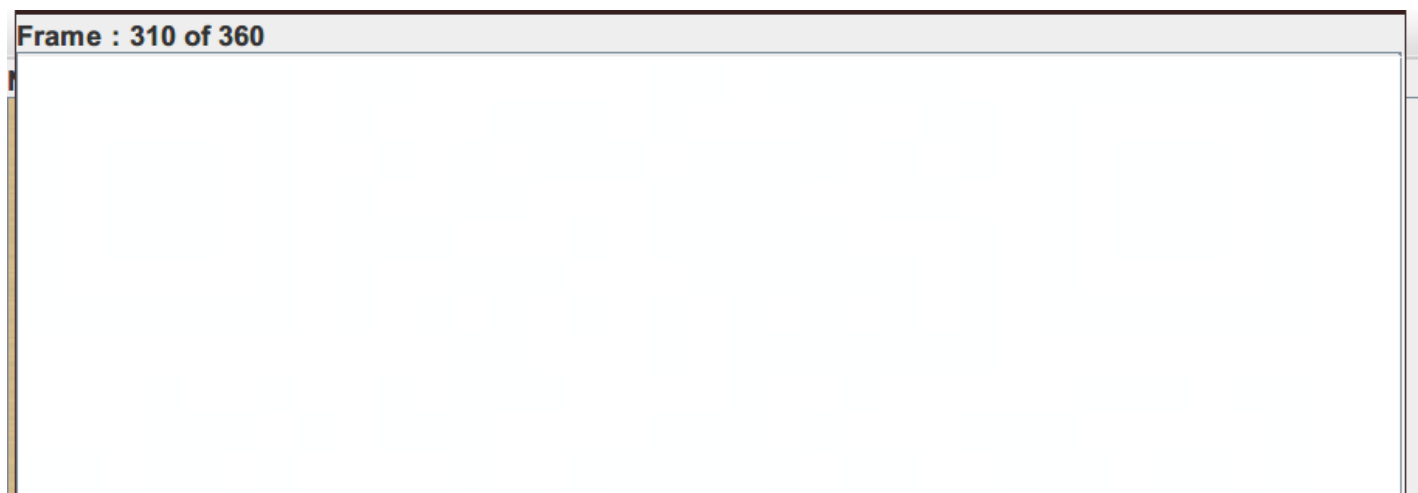
01617696 | 3D 52 75 6B 39 41 D1 A4 2B 9E 09 6E 11 17 4D 8E =Ruk9AÑª+ž n MŽ
01617712 | 11 9D 03 90 8B E6 70 3F 8E 20 B4 43 18 02 FC 78 <æp?Ž ´C üx
01617728 | DF F4 C3 DB 84 BB BB E1 AB 0C 0F 61 50 E9 E3 D5 BðÃŪ„»»á« aPéãŒ
01617744 | 7E EF 79 61 A8 0A 08 13 C6 A5 A1 F0 41 59 CC 8C ~iya" EY;ðAYÏE
01617760 | 4A CE 97 CF 8C A3 32 F2 36 C7 03 56 D4 BA 5E A0 JĪ-ĪE£2ð6Ç VŌ^
01617776 | 2F 36 5B 8B 04 39 FC 0F 13 E5 1F 08 13 BC 56 C0 /6[< 9ü á ¼VÀ
01617792 | 8D D5 3E 71 2B B3 4C 2B 9A 85 C8 FA 49 36 B8 47 Ō>q+³L+š...ÈúI6,G
01617808 | 34 A2 BB E1 27 42 19 76 5B 8B 09 73 87 4E AB 35 4ç>á'B v[< s+N«5
01617824 | 32 74 55 B4 D5 AC 0A 25 9E 5B 78 96 6C 88 68 03 2tU'Ō- %Ž[x-l^h
01617840 | 56 BA 07 E1 91 DB 1A 74 20 90 2D 00 1F 00 00 00 V° á'Ū t -
01617856 | 1F 00 00 00 02 37 01 89 BD 1D 71 13 47 1D 30 08 7 ½½ q G q
01617872 | 00 20 00 00 00 66 60 61 67 2F 74 78 74 00 80 81 51ææ tot 4š
    
```

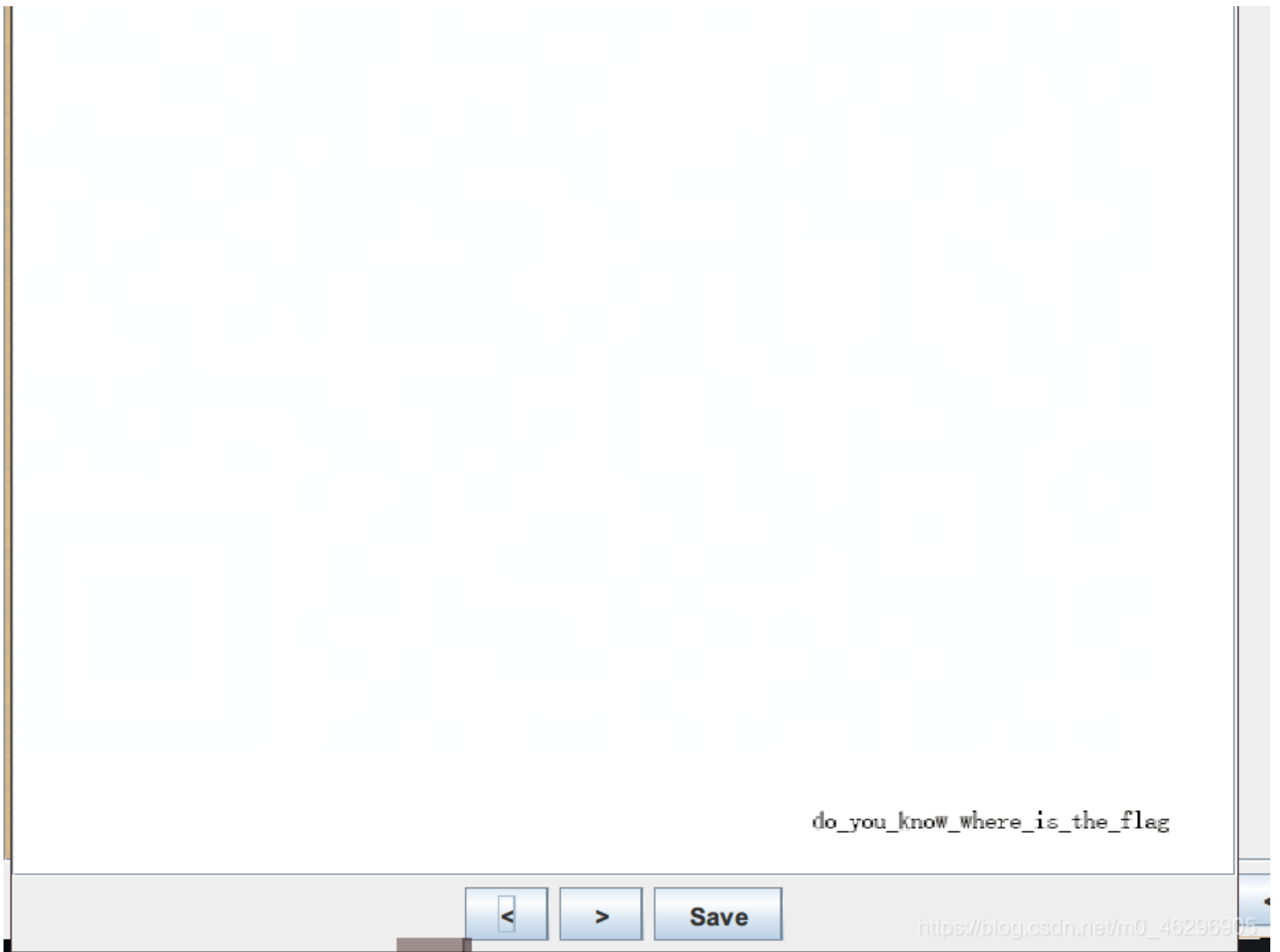
```
01617872 | 00 20 00 00 00 00 00 01 07 2E 74 70 74 00 00 F1 | flag.txt"
01617888 | AC 1A 77 68 65 72 65 20 64 6F 20 79 6F 75 20 74 | - where do you t
01617904 | 68 69 6E 6B 20 74 68 65 20 66 6C 61 67 20 69 73 | hink the flag is
01617920 | 3F C4 3D 7B 00 40 07 00 | ?A={ @
https://blog.csdn.net/m0_46296905
```

得到一个rar压缩包，里面是flag.txt和一个gif，用 ARCHPR 破解压缩密码如下：



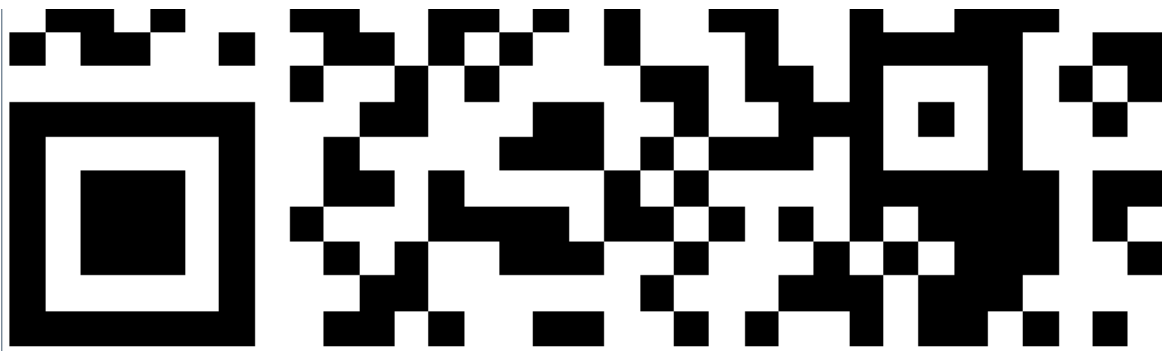
flag不在txt文件中，那就看gif文件，用 stegsolve 的 frame format 逐帧看，在第310帧的时候发现了可疑的地方，提取出来





各个通道查看一下，找到了二维码





do you know where is the flag

https://blog.csdn.net/m0_46296905

扫一下得到flag

QR Research

文件(F) 工具(T) 帮助(H)

纠错等级: H(30%) 掩码: Auto

版本: Auto 尺寸: 4

已解码数据 1:

位置:(6.8,57.0)-(831.9,57.0)-(6.8,879.6)-(832.2,878.1)

颜色正常, 正像

版本: 4

纠错等级: H, 掩码: 7

内容:

`flag(shanxiajingwu_won_the_game)`

https://blog.csdn.net/m0_46296905

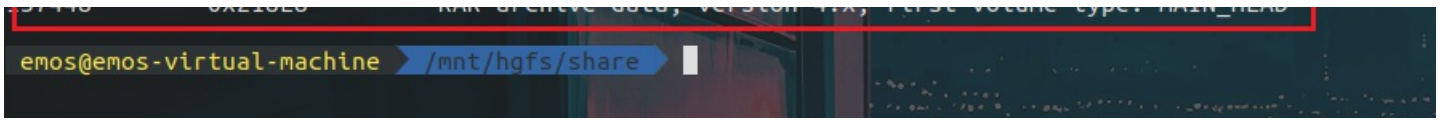
0x14 [GXYCTF2019]gakki

题目:

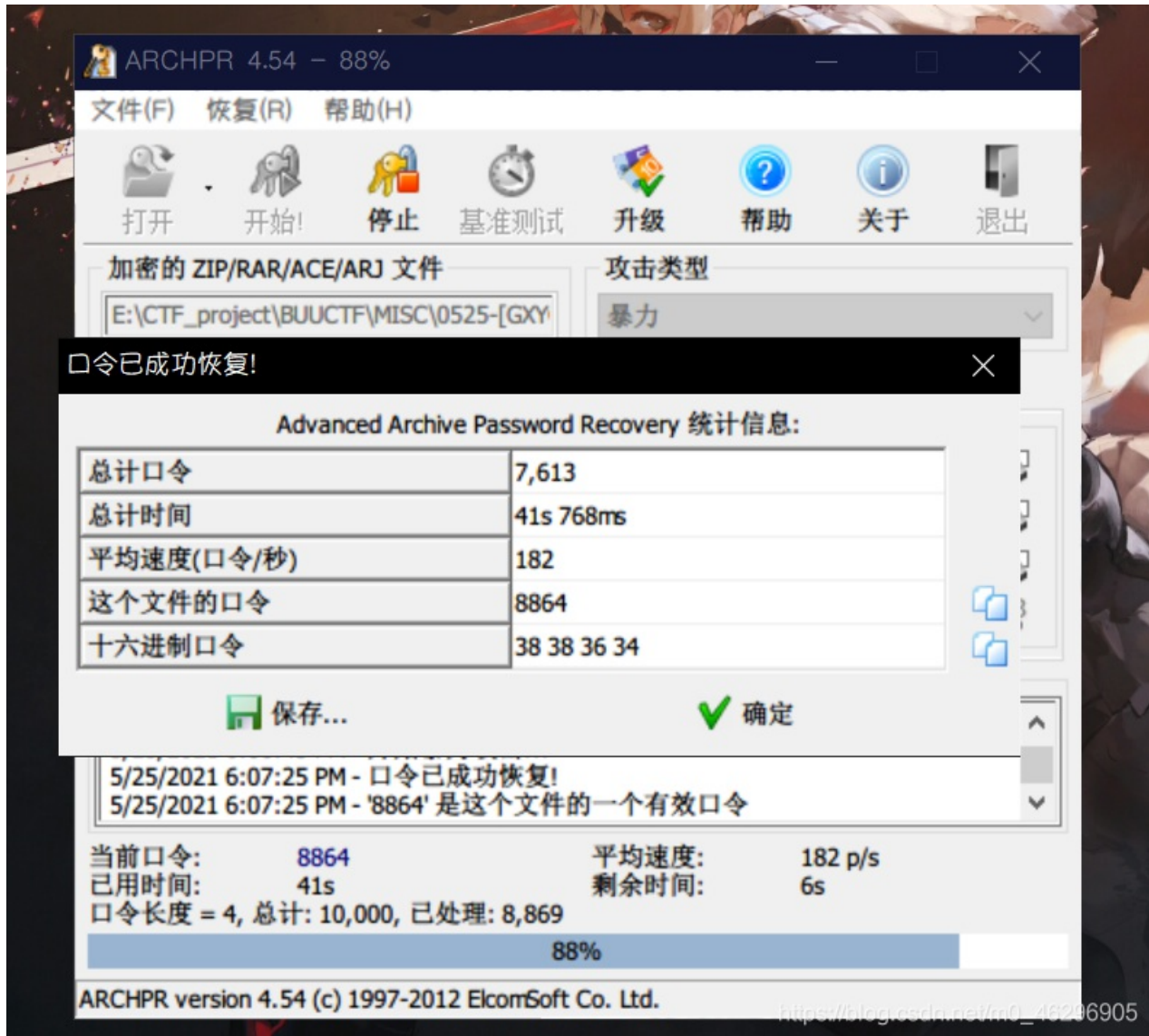


binwalk分析一下，发现隐写了rar压缩包，

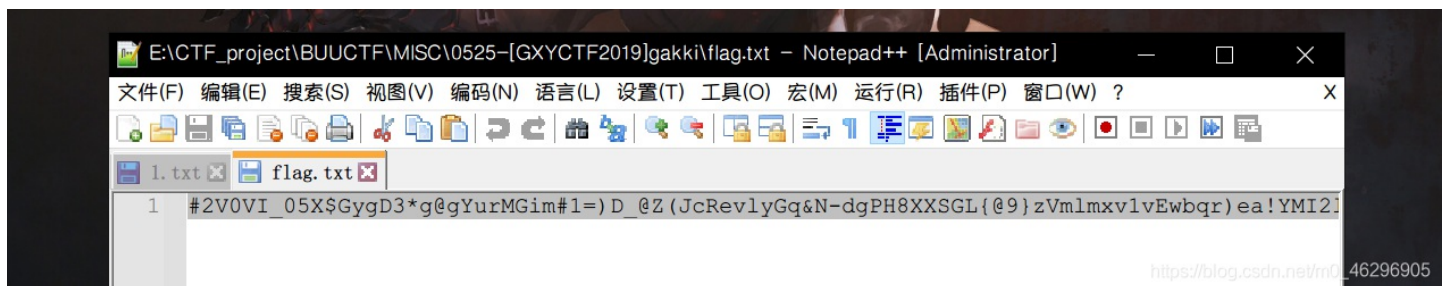
```
emos@emos-virtual-machine /mnt/hgfs/share binwalk wolaoppo.jpg
DECIMAL      HEXADECIMAL  DESCRIPTION
-----
0            0x0         JPEG image data, JFIF standard 1.01
30          0x1E         TIFF image data, big-endian, offset of first image directory: 8
37448       0x218E8     RAR archive data, version 4.x, first volume type: MAIN HEAD
```



提取出的rar是有压缩密码的，直接ARCHPR破解一下，



解压得到一串奇怪的字符



参考别的博主的wp知道是要统计字母频率，然后按频率顺序输出，

```

f=open("flag.txt",'r').read()#读文件
l=[]
for i in range(256):
    l.append(0)#初始化频率表,下标作为ASCII码,值是频数
for i in f:
    l[ord(i)]+=1 #读一个对应下标就加一
b=l[:]#对列表结果复制到另一个列表
b.sort()#按频率排序
b.reverse()#倒置一下,不倒置不能输出正序的fLag
flag=''
for i in b:
    flag+=chr(l.index(i))#索引出对应的字符拼成fLag
print(flag)

```

参考一下别的博主的wp, 两个都用的是字典:

```

alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()_+- ={}[]"
f = open("flag.txt", "r")
data = f.read()
result = {d:0 for d in alphabet}

def sort_by_value(d):
    items = d.items()
    backitems = [[v[1],v[0]] for v in items]
    backitems.sort(reverse=True)
    return [ backitems[i][1] for i in range(0,len(backitems))]

for d in data:
    for alpha in alphabet:
        if d == alpha:
            result[alpha] = result[alpha] + 1

print(sort_by_value(result))

```

```

alphabet = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890!@#$%^&*()_+- =\\{\\}[]"
strings = open('./flag.txt').read()

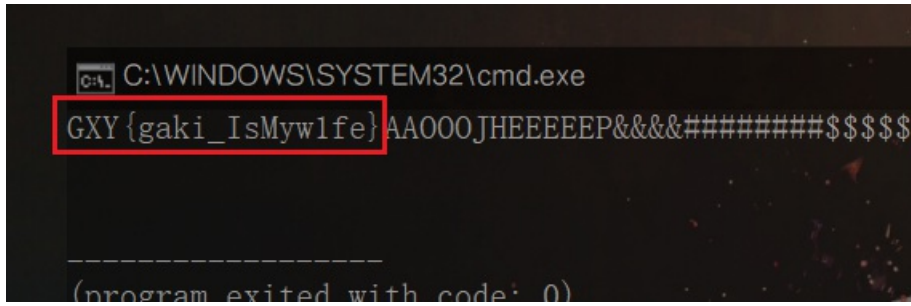
result = {}
for i in alphabet:
    counts = strings.count(i)
    i = '{0}'.format(i)
    result[i] = counts

res = sorted(result.items(),key=lambda item:item[1],reverse=True)
for data in res:
    print(data)

for i in res:
    flag = str(i[0])
    print(flag[0],end="")

```

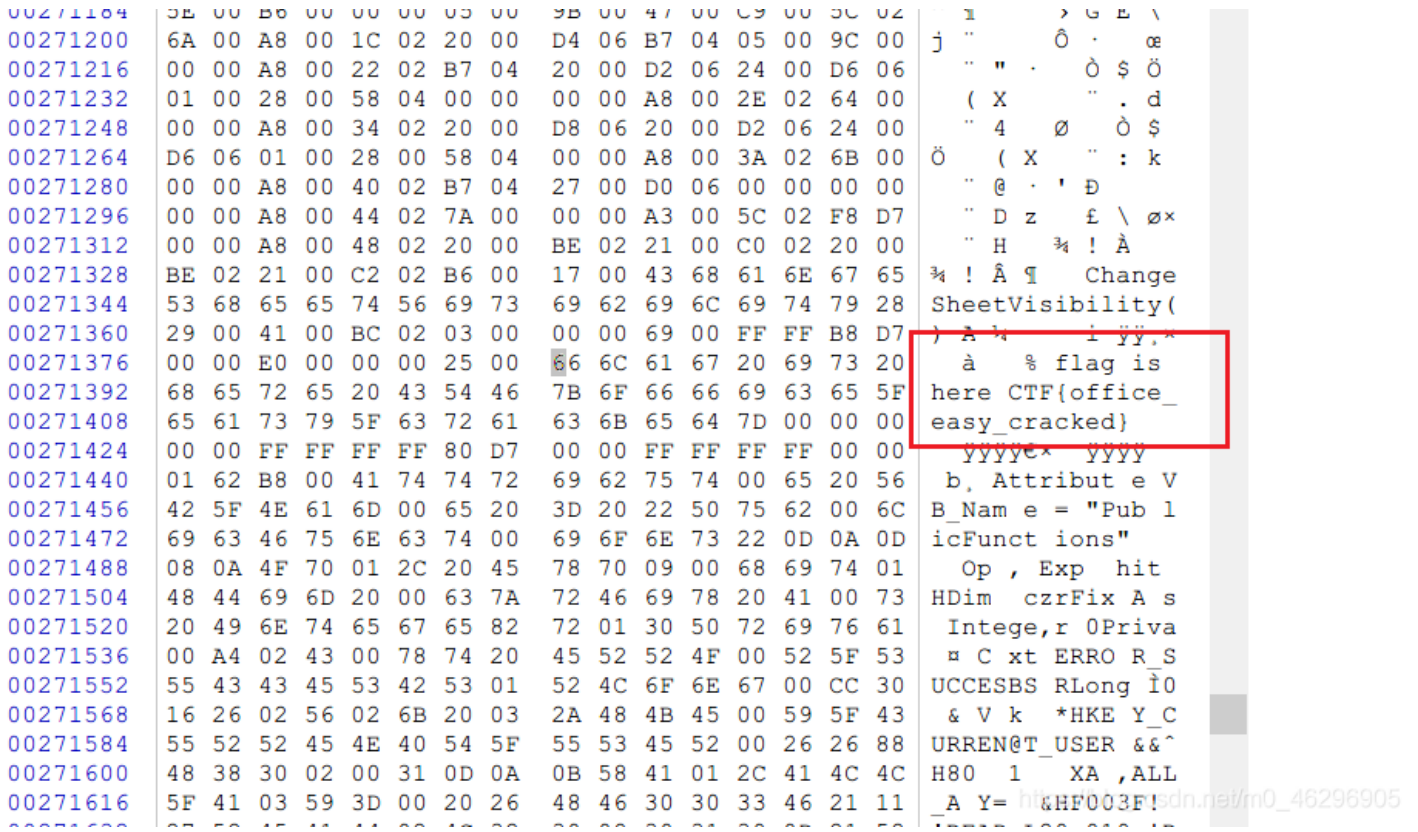

最终flag如下:



flag{gaki_IsMyw1fe}

0x15 [HBNIS2018]excel破解

有密码保护的excel, winhex打开, search一下flag

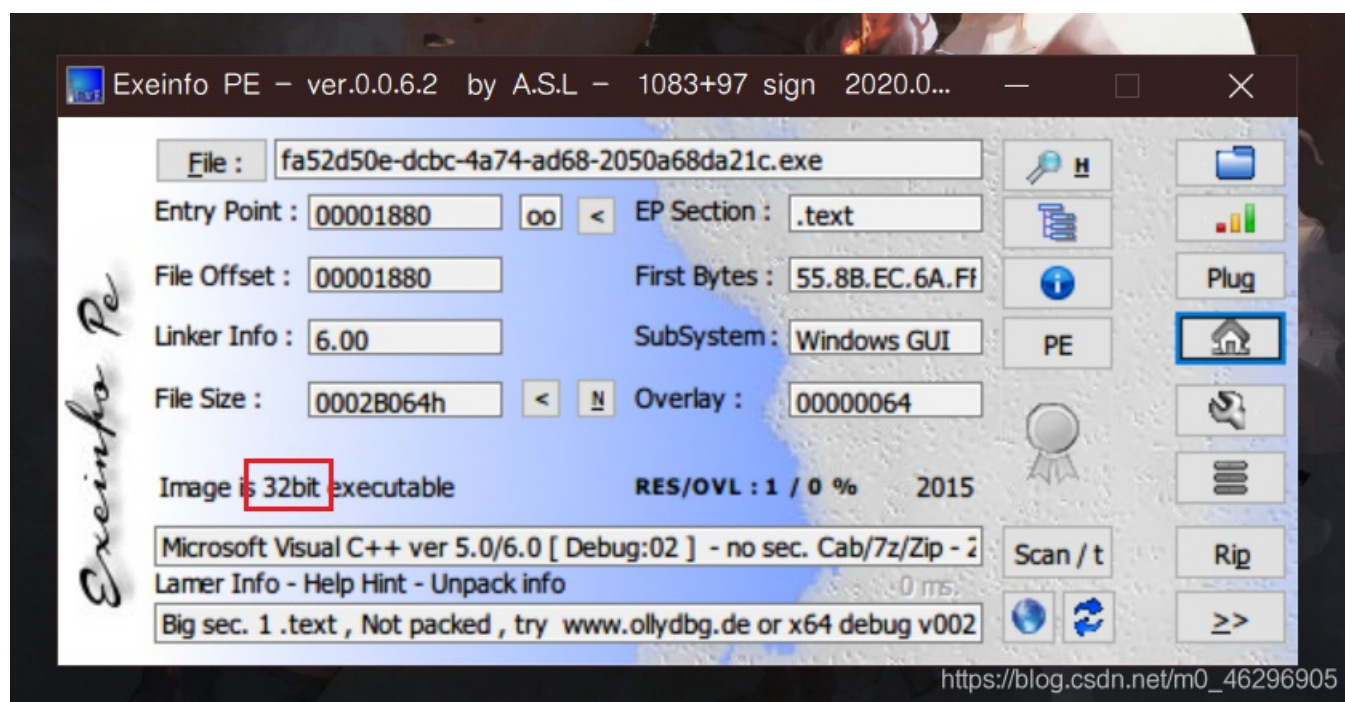


flag{office_easy_cracked}

0x16 Mysterious

题目: 自从报名了CTF竞赛后, 小明就辗转于各大论坛, 但是对于逆向题目仍是一知半解。有一天, 一个论坛老鸟给小明发了一个神秘的盒子, 里面有开启逆向思维的秘密。小明如获至宝, 三天三夜, 终于解答出来了, 聪明的你能搞定这个神秘盒子么?

放到exeinfo里看看，发现是32位可执行文件



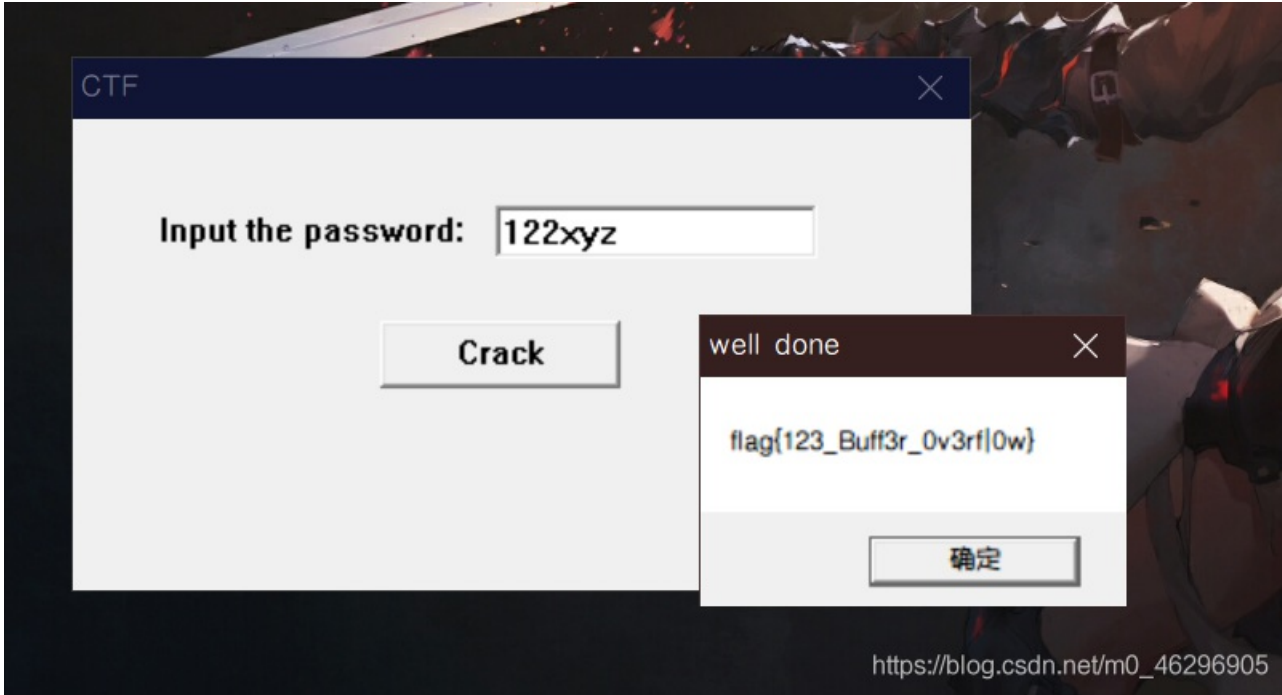
从读取用户输入的API函数 `GetDlgItemTextA` 为突破口找到关键函数，分析一下发现输入 `122xyz` 即可得到flag

```

int __stdcall sub_401090(HWND hWnd, int a2, int a3, int a4)
{
    int v4; // eax
    char Source[260]; // [esp+50h] [ebp-310h] BYREF
    CHAR Text[5]; // [esp+154h] [ebp-20Ch] BYREF
    char v8[252]; // [esp+159h] [ebp-207h] BYREF
    __int16 v9; // [esp+255h] [ebp-10Bh]
    char v10; // [esp+257h] [ebp-109h]
    int Value; // [esp+258h] [ebp-108h]
    CHAR String[260]; // [esp+25Ch] [ebp-104h] BYREF

    memset(String, 0, sizeof(String));
    Value = 0;
    if ( a2 == 16 )
    {
        DestroyWindow(hWnd);
        PostQuitMessage(0);
    }
    else if ( a2 == 273 )
    {
        if ( a3 == 1000 )
        {
            GetDlgItemTextA(hWnd, 1002, String, 260);
            strlen(String);
            if ( strlen(String) > 6 )//长度受限
                ExitProcess(0);
            v4 = atoi(String);//字符串转整型("122"-->122)
            Value = v4 + 1;
            if ( v4 == 122 && String[3] == 'x' && String[5] == 'z' && String[4] == 'y' )//输入密码是"122xyz"
            {
                strcpy(Text, "flag");
                memset(v8, 0, sizeof(v8));
                v9 = 0;
                v10 = 0;
                _itoa(Value, Source, 10);
                strcat(Text, "{");
                strcat(Text, Source);
                strcat(Text, "_");
                strcat(Text, "Buff3r_0v3rf|0w");
                strcat(Text, "}");
                MessageBoxA(0, Text, "well done", 0);
            }
            SetTimer(hWnd, 1u, 0x3E8u, TimerFunc);
        }
        if ( a3 == 1001 )
            KillTimer(hWnd, 1u);
    }
    return 0;
}

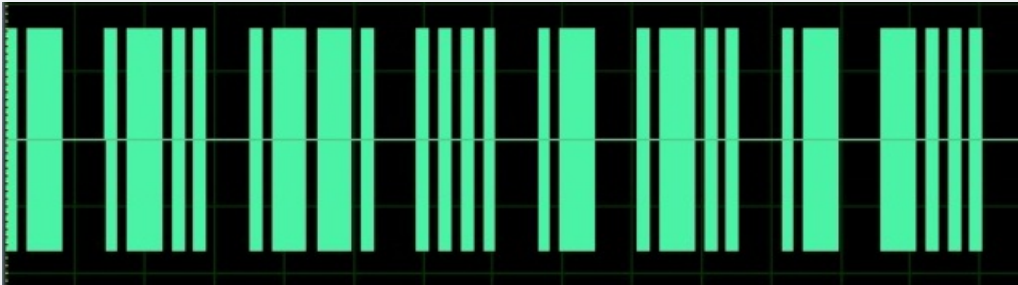
```



```
flag{123_Buff3r_0v3rf|0w}
```

0x17 [HBNIS2018]来题中等的吧

题目:



又是摩斯密码

```
.- .-... .- .-... .- .-... .- .-...
```

```
flag{alphalab}
```

0x18 [ACTF新生赛2020]base64隐写

根据题目知道是base64隐写
换python2的源跑一下下面的脚本


```
# -*- coding: cp936 -*-
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
with open('ComeOn!.txt', 'rb') as f:
    bin_str = ''
    for line in f.readlines():
        stegb64 = ''.join(line.split())
        rowb64 = ''.join(stegb64.decode('base64').encode('base64').split())
        offset = abs(b64chars.index(stegb64.replace('=', ''))[-1] - b64chars.index(rowb64.replace('=', ''))[-1])
        equalnum = stegb64.count('=') #no equalnum no offset
        if equalnum:
            bin_str += bin(offset)[2:].zfill(equalnum * 2)
    print ''.join([chr(int(bin_str[i:i + 8], 2)) for i in xrange(0, len(bin_str), 8)])
```

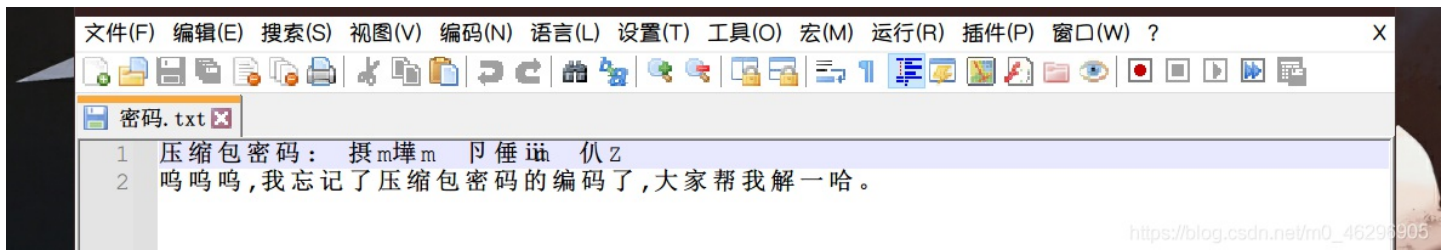


flag{6aseb4_f33!}

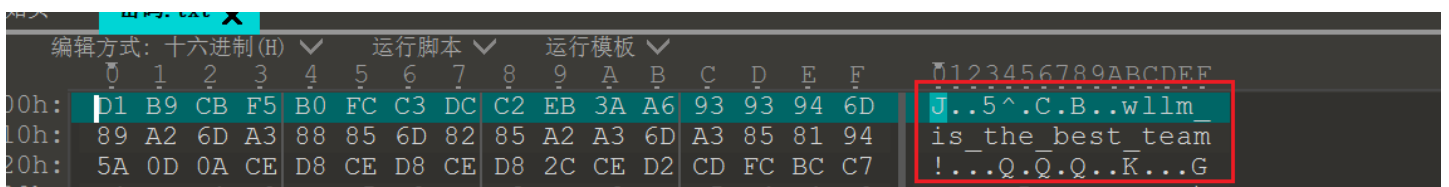
0x19 [SWPU2019]伟大的侦探

先用 ziperello 试着破解一下压缩密码，发现失败。

密码.txt 是可以解压出来的，里面本该是密码的部分变成了乱码



用010editor打开，试着换个编码方式，发现密码 wllm_is_the_best_team!



```

00h:  C1 CB D1 B9 CB F5 B0 FC C3 DC C2 EB B5 C4 B1 E0 A..S^..C.B..D.\
40h:  C2 EB C1 CB 2C B4 F3 BC D2 B0 EF CE D2 BD E2 D2 B.A...3.K^..K.SK
50h:  BB B9 FE A1 A3 ]...~t

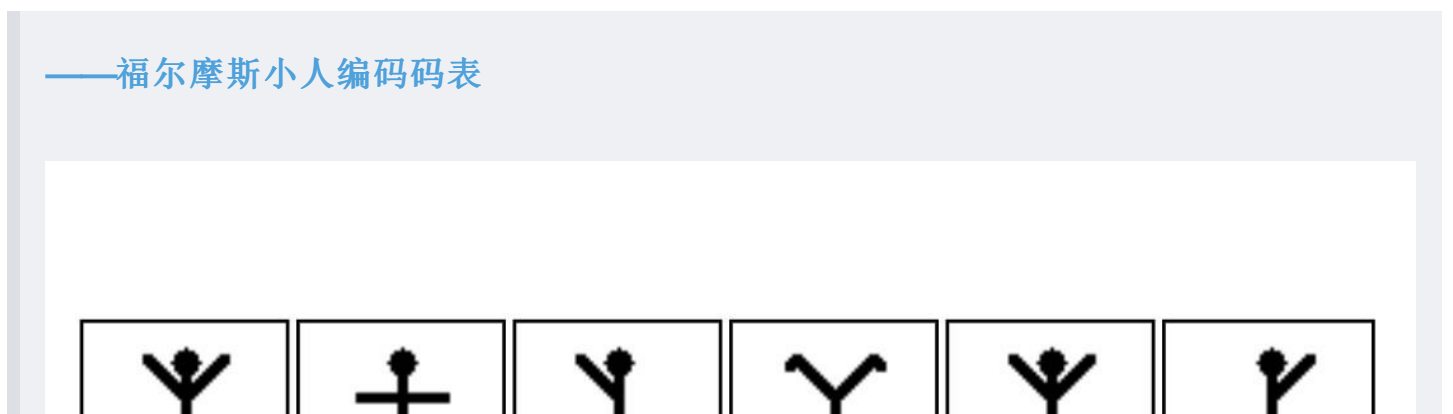
```

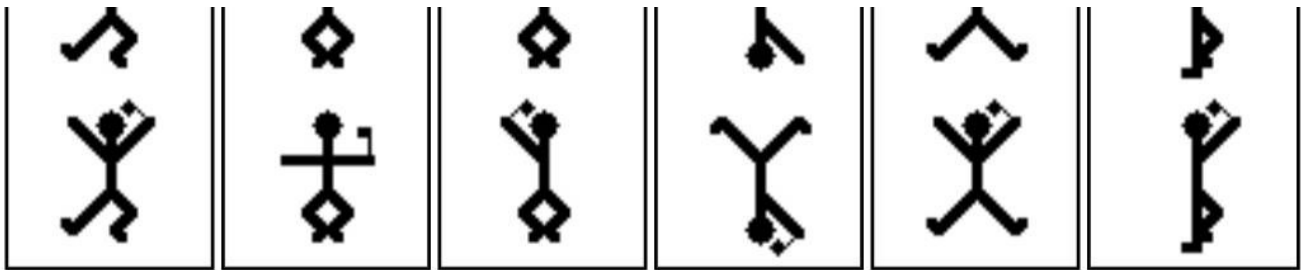
https://blog.csdn.net/m0_46296905

得到的图片是 [福尔摩斯小人密码](#)



给出码表





a

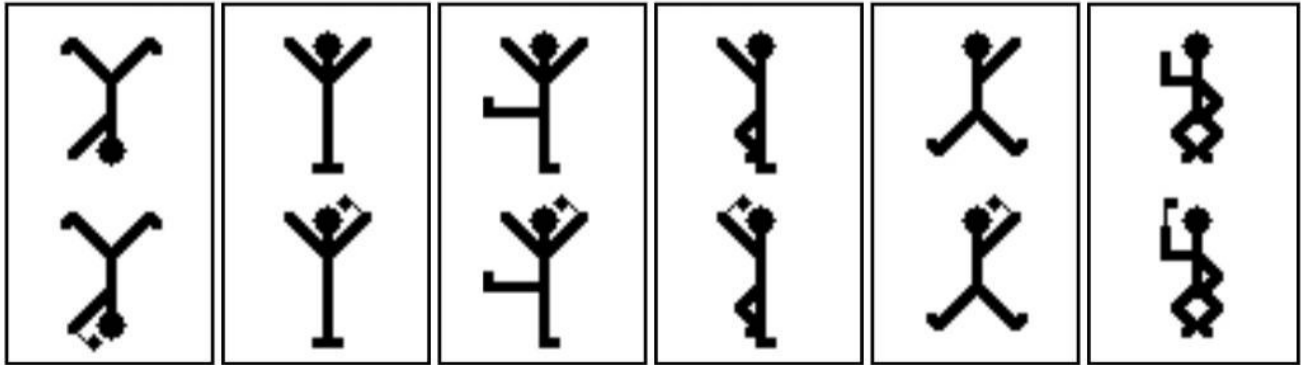
b

c

d

e

f



g

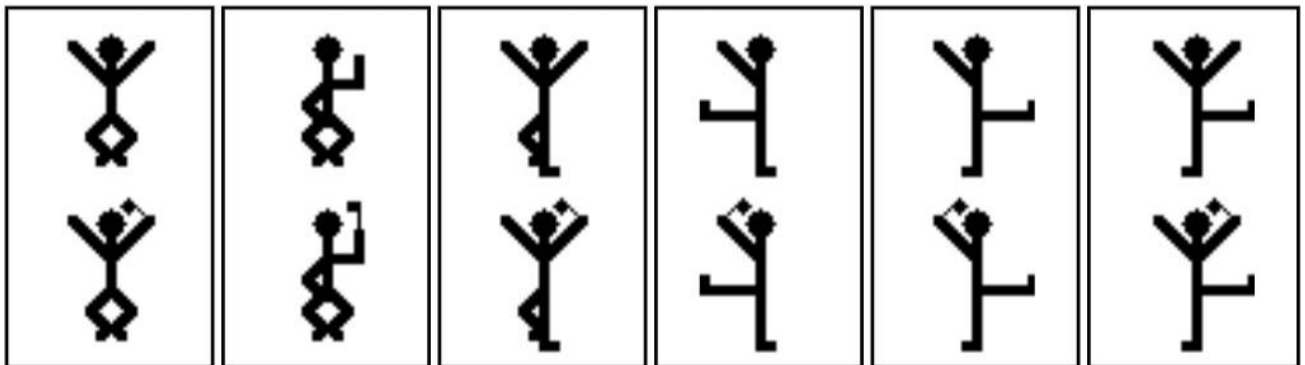
h

i

j

k

l



m

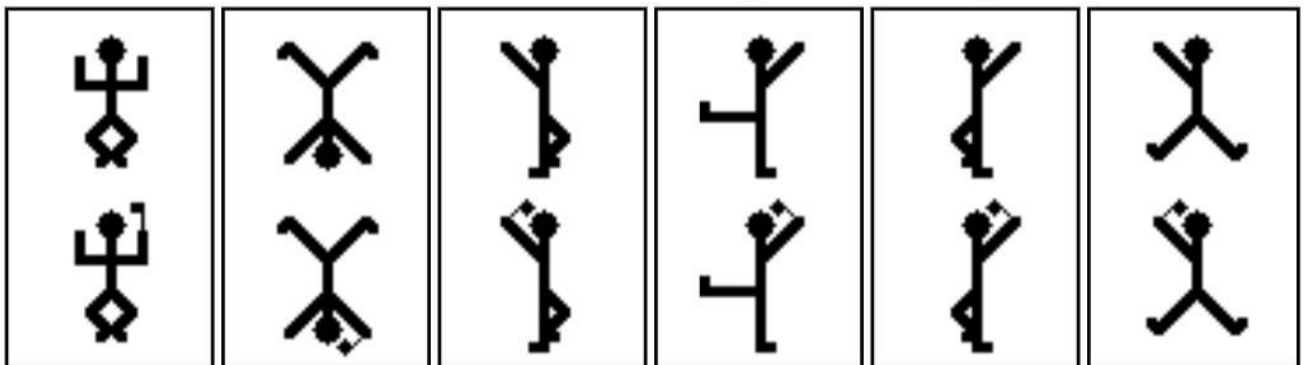
n

o

p

q

r



s

t

u

v

w

x

https://blog.csdn.net/m0_46298903

对照得到flag

flag{iloveholmesandwllm}

去掉png的头即可



flag{57cd4cfd4e07505b98048ca106132125}

0x1C 喵喵喵

题目：喵喵喵，扫一扫



根据题目推测隐写了一个二维码，果然在0通道发现如下隐写的黑白方块

File Analyse Help

Blue plane 0



StegSolve 1.3 by Caesum

File Analyse Help

Green plane 0



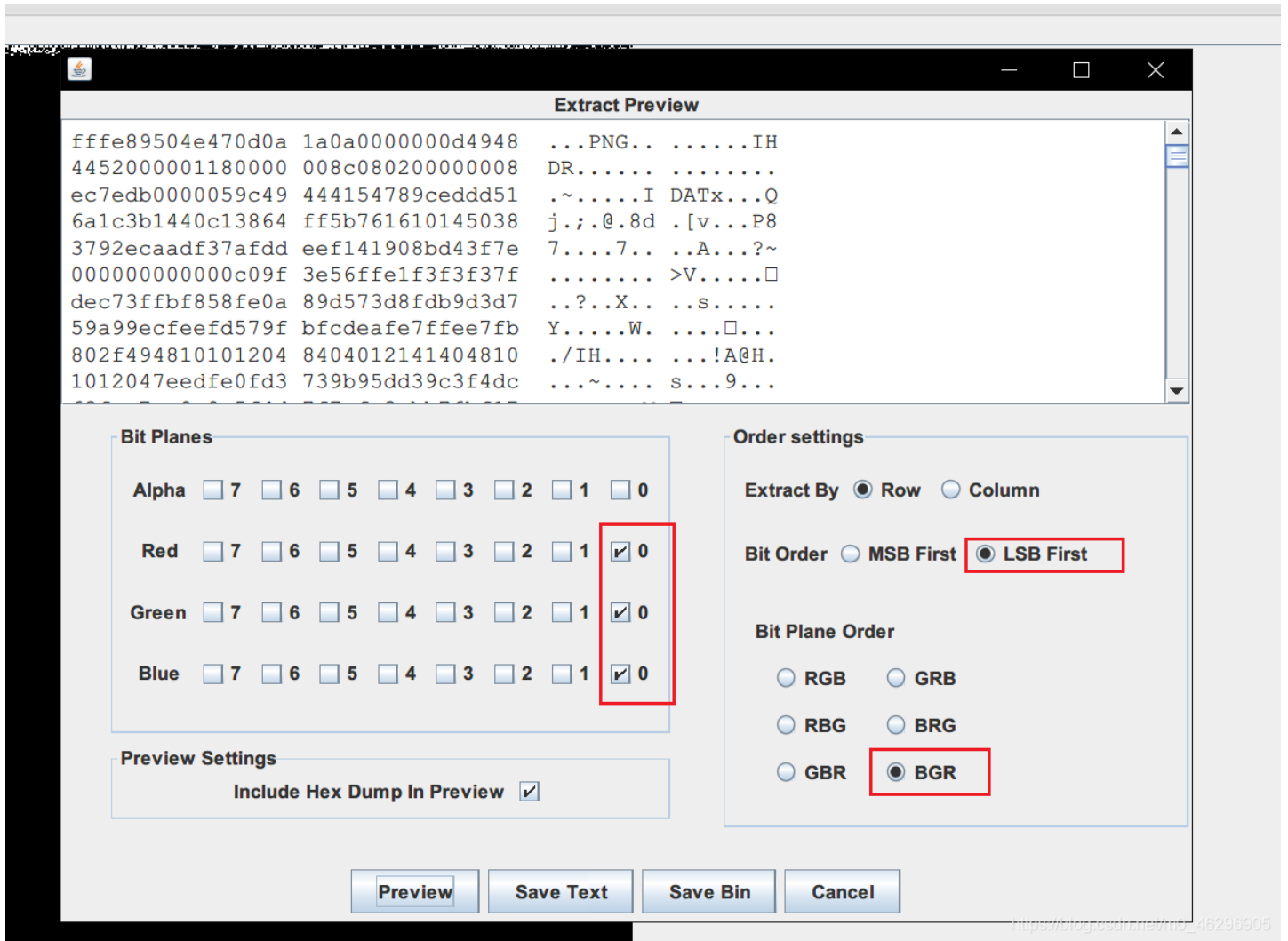
StegSolve 1.3 by Caesum

File Analyse Help

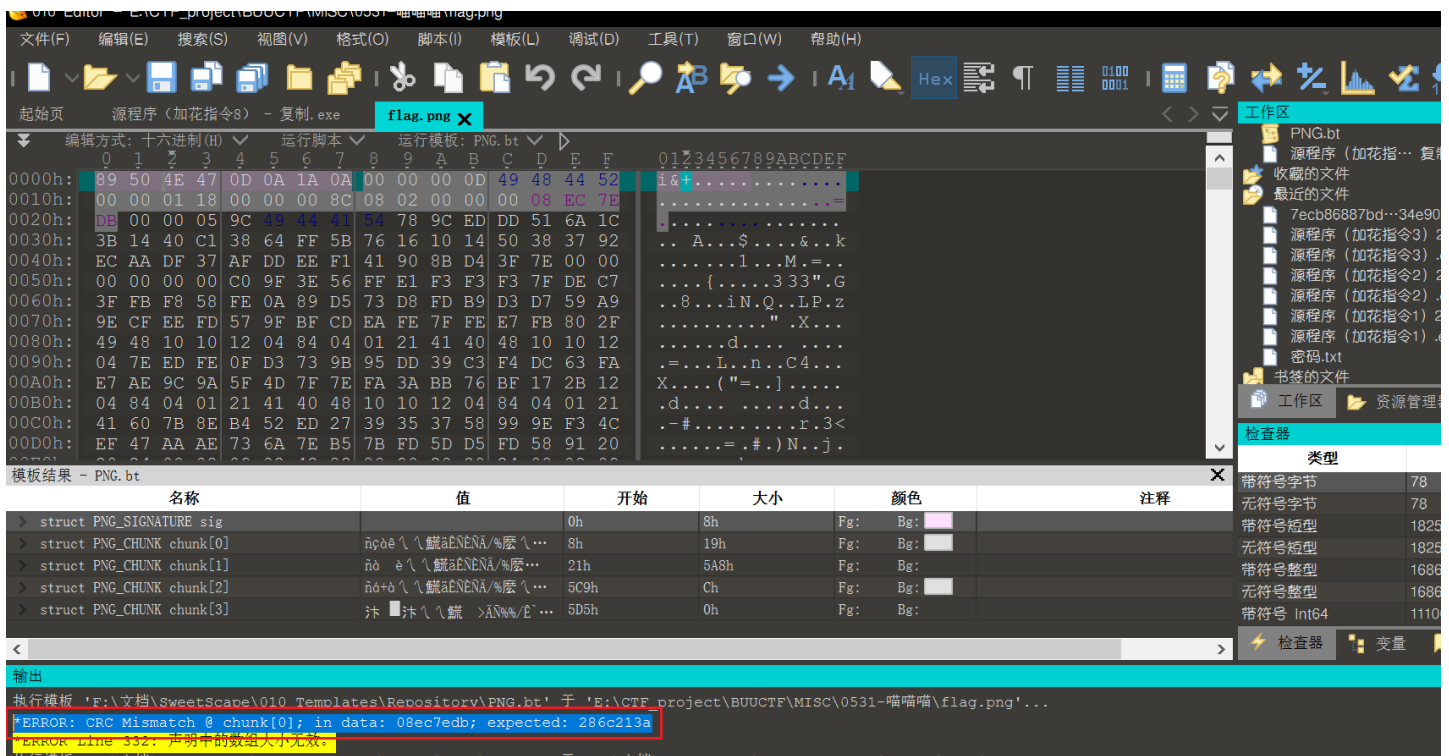
Red plane 0



一开始在默认的RGB位平面顺序下没有看到文件头，就一个个尝试不同的位平面顺序，最终在选择位平面顺序是BGR的时候出现了PNG文件头，直接导出



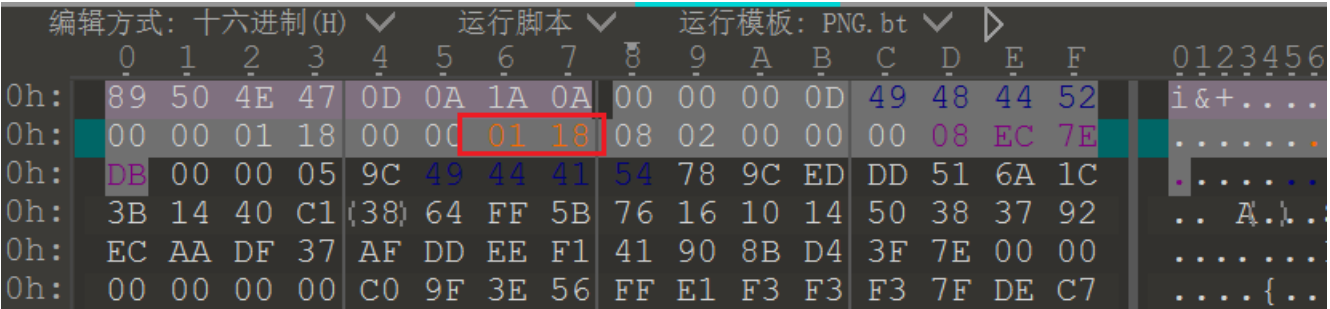
还要用 010 Editor 把前面的多余数据 FF FE 去掉，同时出现CRC不匹配的情况，



结合预览图片发现只有半张二维码，



改一下高试试



得到如下二维码

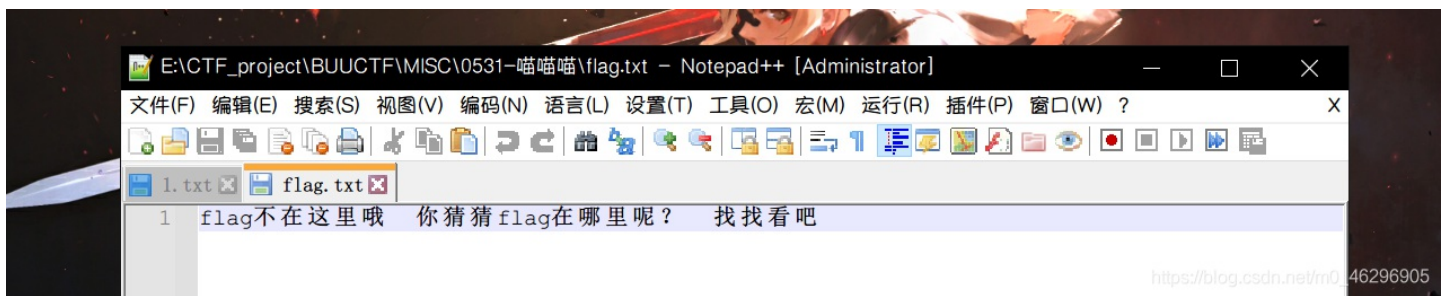


是个百度网盘网址





又是个隐写文件



先试试 [NTFS文件流隐写](#) ,

- 使用NtfsStreamsEditor工具扫描需要注意：
- 1.解压需要在win7中使用winrar解压软件解压。
 - 2.需要在win7系统中进行搜索。



```
# Embedded file name: flag.py
import base64

def encode():
    flag = '*****'
    ciphertext = []
    for i in range(len(flag)):
        s = chr(i ^ ord(flag[i]))
        if i % 2 == 0:
            s = ord(s) + 10
        else:
            s = ord(s) - 10
        ciphertext.append(str(s))

    return ciphertext[::-1]

ciphertext = ['96',
'65',
'93',
'123',
'91',
'97',
'22',
'93',
'70',
'102',
'94',
'132',
'46',
'112',
'64',
'97',
'88',
'80',
'82',
'137',
'90',
'109',
'99',
'112']
```

贴上解密脚本

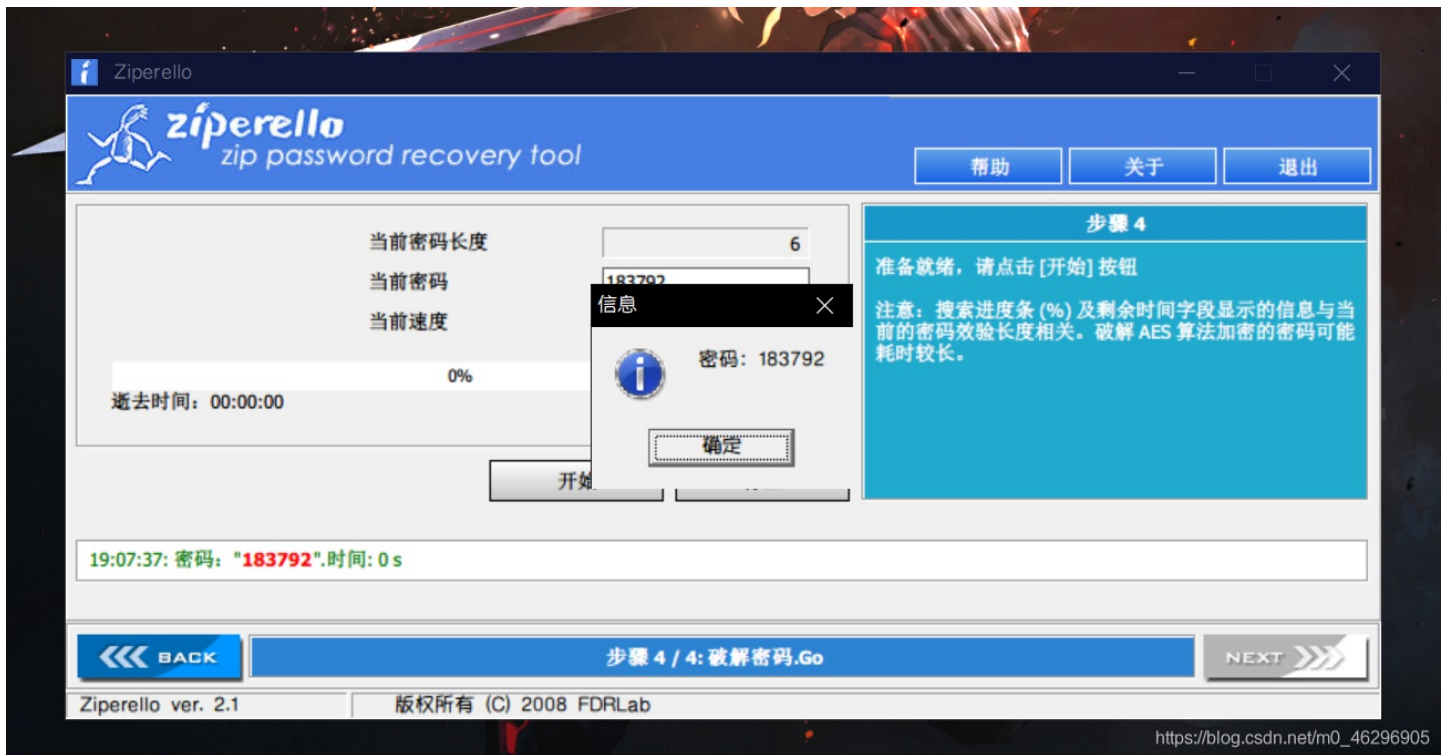

```
ciphertext = ['96','65','93','123','91','97','22','93','70','102','94','132','46','112','64','97','88','80','82',
,'137','90','109','99','112']
ciphertext.reverse() #逆置
flag=''
for i in range(len(ciphertext)):
    if i % 2 == 0:
        flag += chr((int(ciphertext[i]) - 10)^i)
    else:
        flag += chr((int(ciphertext[i]) + 10)^i)
print(flag)
```

得到flag

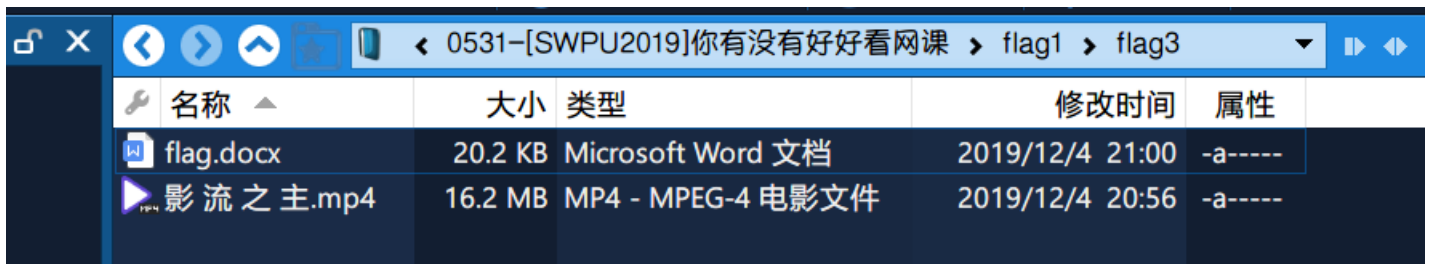
```
flag{Y@e_C13veR_C1Ever!}
```

0x1D [SWPU2019]你有没有好好看网课?

两个压缩包，flag3.zip 可以破解出来密码



得到两个文件，一个文档，一个无声的视频，



文档里面特别强调数字，猜测是在视频5分20秒和7分11秒的地方观察，





段落



样式

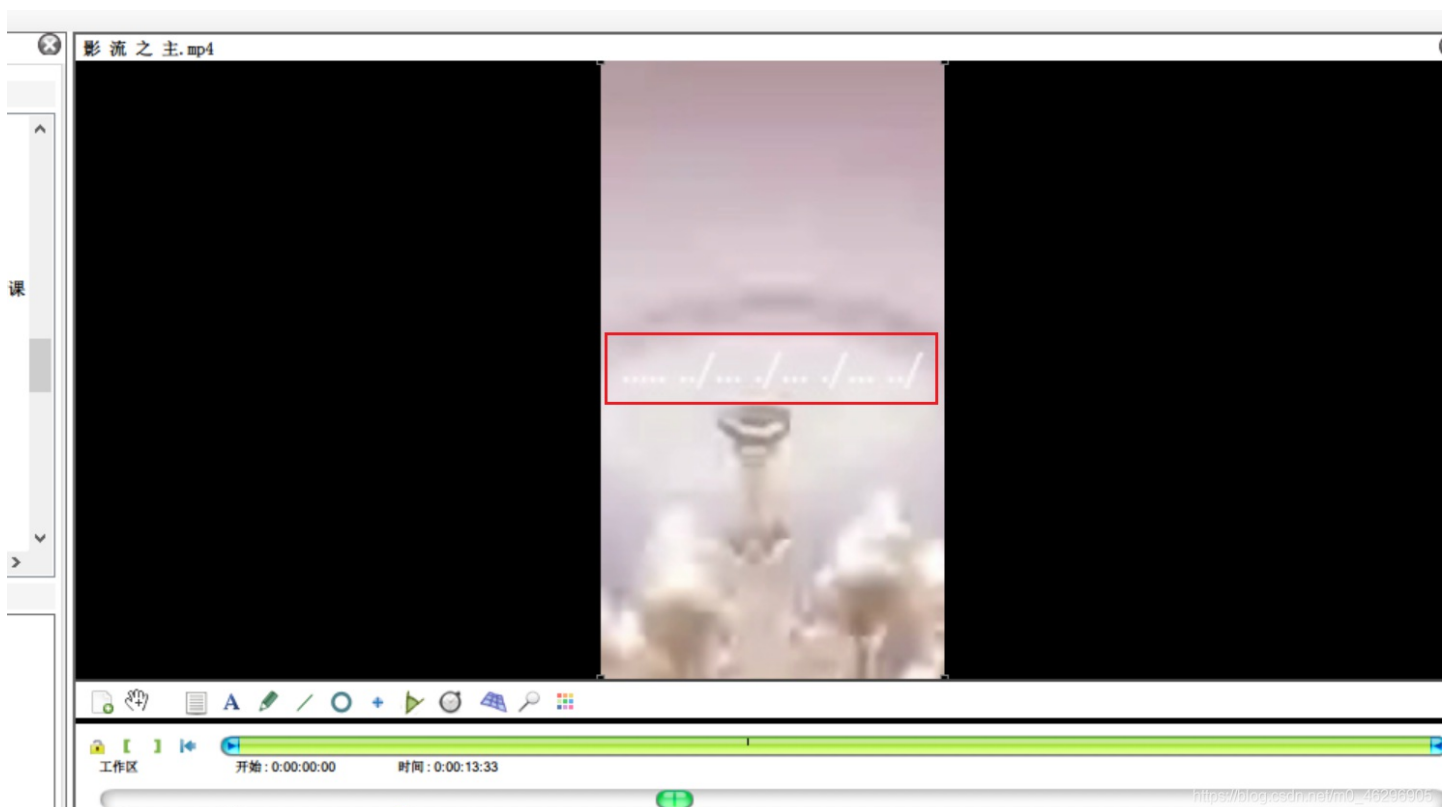
从小 5 就 20 列文虎克, ←

我每年的 7 月 11 日的生日愿望就是拥有一个 🔍 ←

←



用视频分析工具kinovea一帧一帧看一下,





得到两个有用信息，一个是敲击码，一个是base64

```
..... ./... ./... ./... ./
dXBfdXBfdXA=
```

敲击码

码表如下

| | | | | |
|---|---|---|-----|-----|
| 1 | 2 | 3 | 4 | 5 |
| 1 | A | B | C/K | D E |
| 2 | F | G | H I | J |
| 3 | L | M | N O | P |
| 4 | Q | R | S T | U |
| 5 | V | W | X Y | Z |

以 / 为字母间的分割，space空格 来分隔行列，如 对应5行2列的w

解密得到解压密码

```
wllmup_up_up
```

得到如下图片



搜索 flag 没搜到，那么试试 swpu，搜索成功

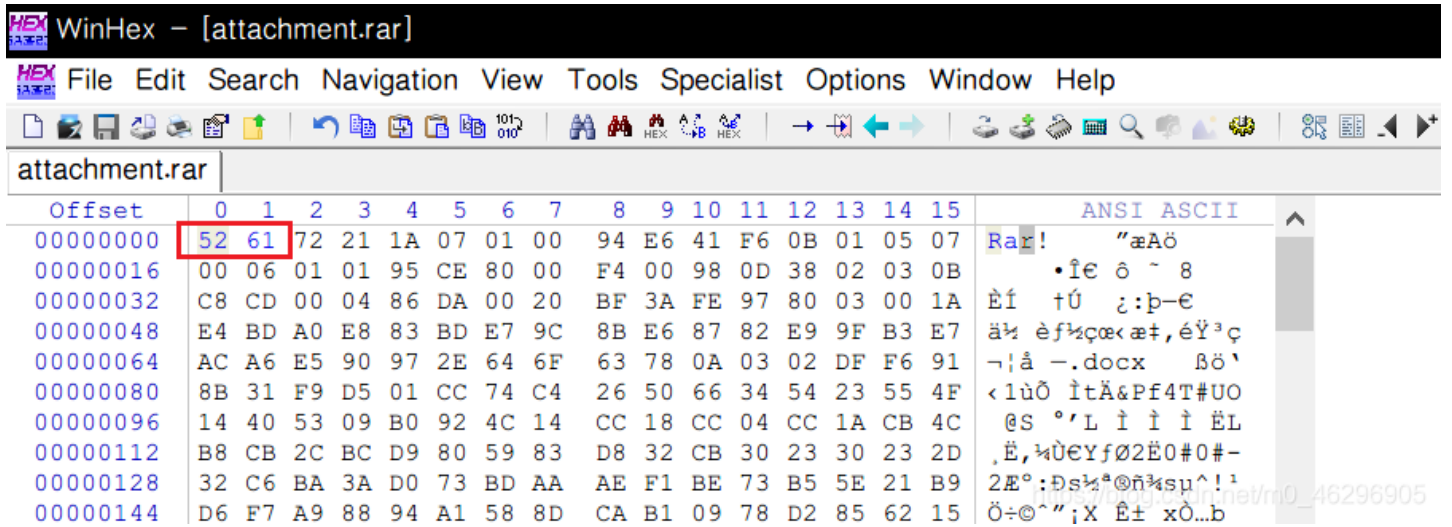
```
HEX File Edit Search Navigation View Tools Specialist Options Window Help
Real flag.jpg
Offset 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ANSI ASCII
00005328 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005344 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005360 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005376 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005392 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005408 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005424 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005440 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005456 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005472 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005488 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005504 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005520 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005536 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005552 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005568 0A 0A 0A 0A 0A 0A 0A 0A 0A 73 77 70 75 63 74 66 swpuctf
00005584 7B 41 32 65 5F 59 30 75 5F 4F 6B 3F 7D 0A 0A 0A {A2e_Y0u_Ok?}
00005600 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005616 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005632 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005648 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005664 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005680 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005696 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005712 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005728 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005744 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005760 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005776 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005792 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005808 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005824 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005840 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005856 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005872 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005888 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005904 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
00005920 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A
```

https://blog.csdn.net/m0_46296905

得到flag

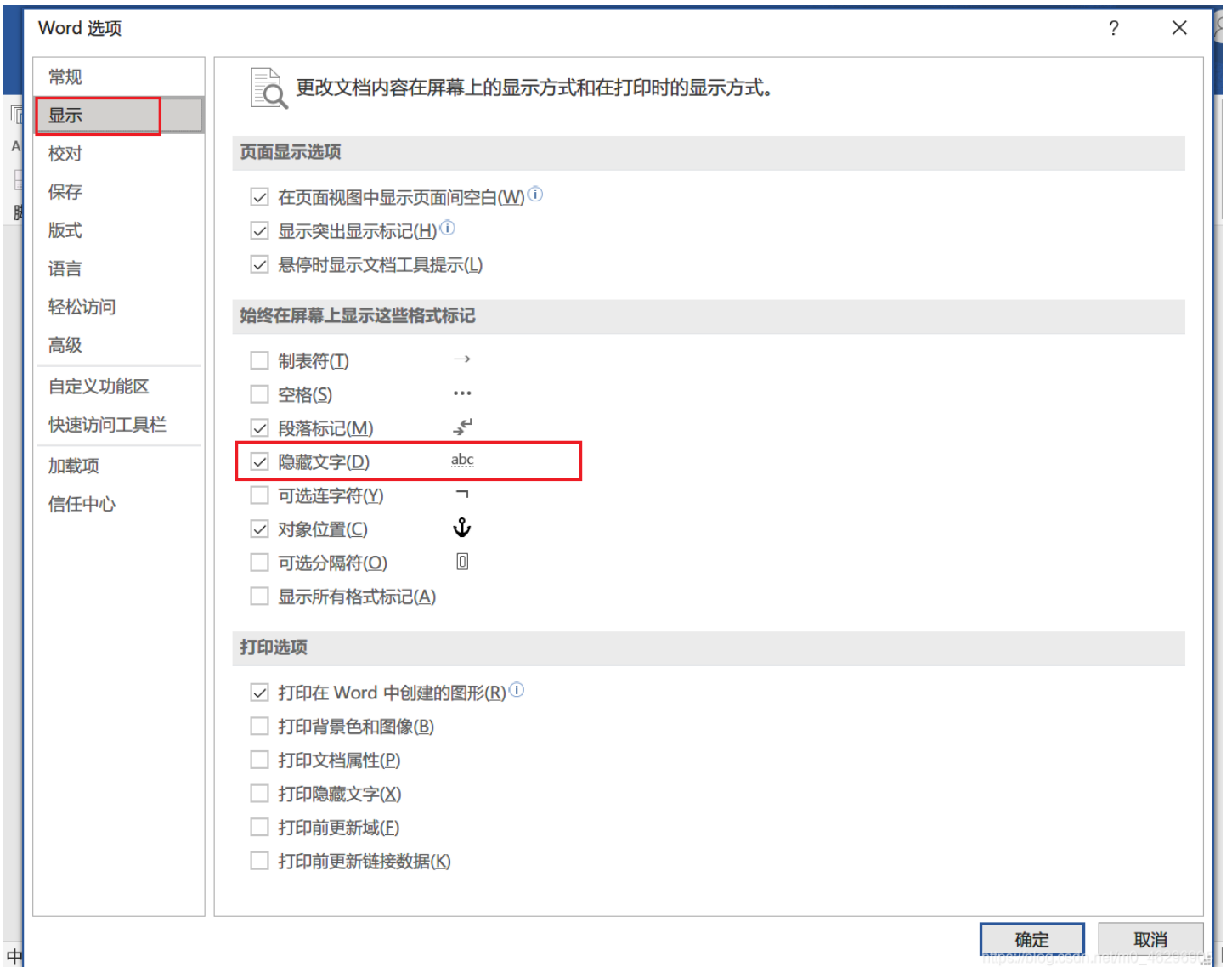
0x1E [MRCTF2020]你能看懂音符吗

文件头损坏，改一下文件头



解压后发现一个文档，看看有没有隐藏文字，点击 **文件**，按照下图操作





然后发现一串音符密码，但不知道为什么复制不了

