BUUCTF misc 解题记录 二（超级详细）

原创

Vayn3 于 2021-04-14 23:40:02 发布 1441 收藏 10

分类专栏： 笔记 文章标签： 安全 python

本文链接：https://blog.csdn.net/qq_51090016/article/details/115712647

这里是从第四页开始的题目

[SUCTF2018]followme（kali下搜索整个文件夹）

先导出http

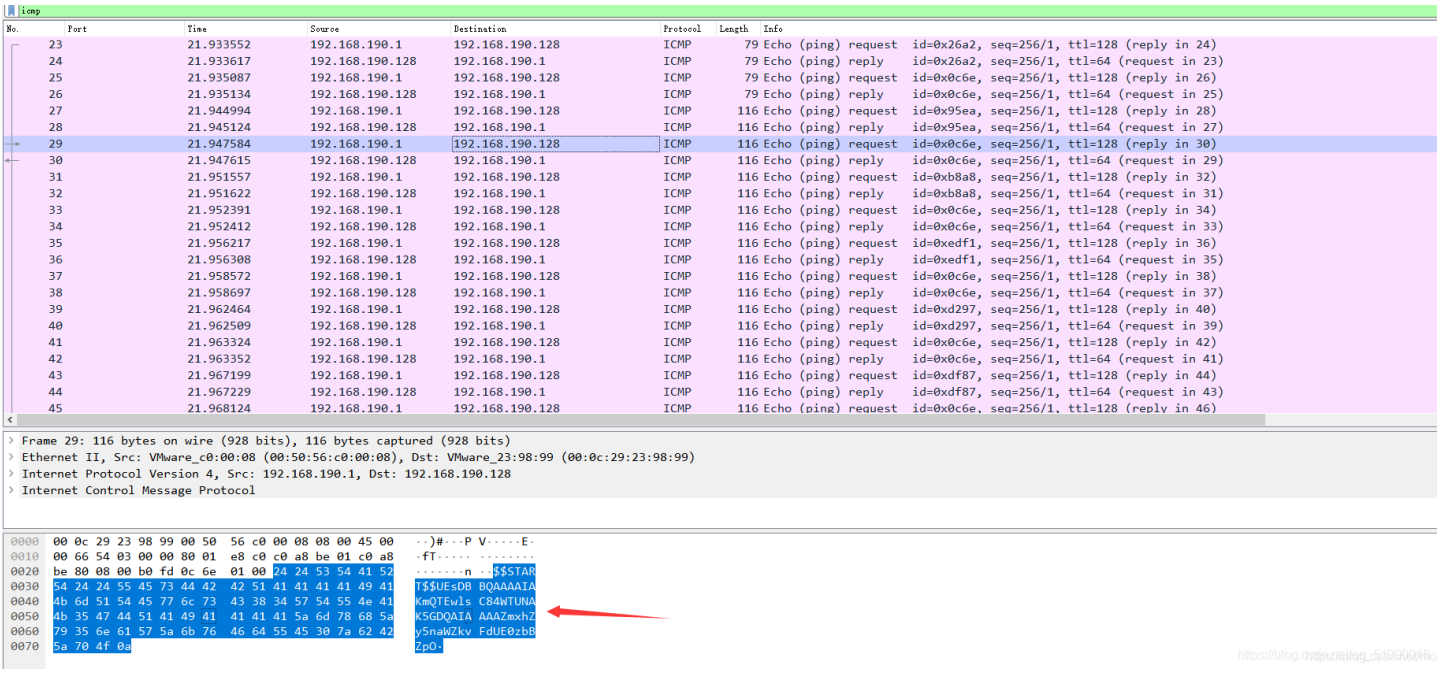
再放到kali

kali下输入grep -r ‘CTF’ ./文件名/

```
caijia@kali2020:~/桌面$ grep -r 'C    关于Qt(Q)...
./buu/login%3f_=6975b9a9f7a359d322e06c0e28db112b(123):name=admin&password=S
UCTF{password_is_not_weak}&referer=http%3A%2F%2F192.168.128.145%2Fbuild%2Fa
dmin%2F
匹配到二进制文件 ./buu/attachment.pcapng
```

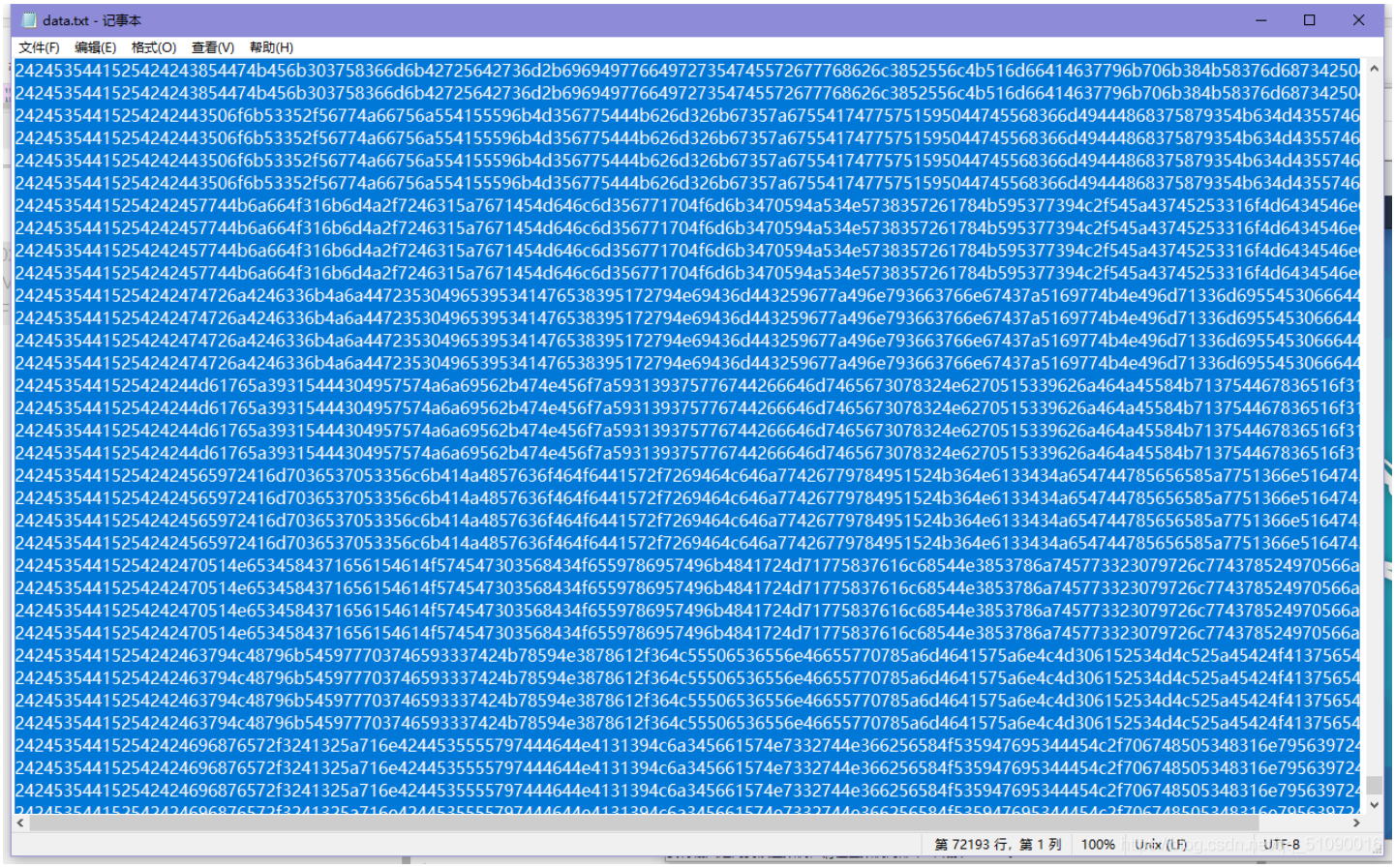
蜘蛛侠呀（tshark提取数据）

所有的icmp包后面都跟了一串数据，使用tshark把这些全部提取出来




```
tshark -r out.pcap -T fields -e data > data.txt
```

得到data.txt，发现有重复，用脚本去重




```
with open('data.txt', 'r') as file:
    res_list = []
    lines = file.readlines()
    print('[+]去重之前一共{0}行'.format(len(lines)))
    print('[+]开始去重，请稍等.....')
    for i in lines:
        if i not in res_list:
            res_list.append(i)
    print('[+]去重后一共{0}行'.format(len(res_list)))
    print(res_list)

with open('data1.txt', 'w') as new_file:
    for j in res_list:
        new_file.write(j)
```

再将十六进制数据转为字符

```
import binascii

with open('data1.txt','r') as file:
    with open('data2.txt','wb') as data:
        for i in file.readlines():
            data.write(binascii.unhexlify(i[:-1]))
```

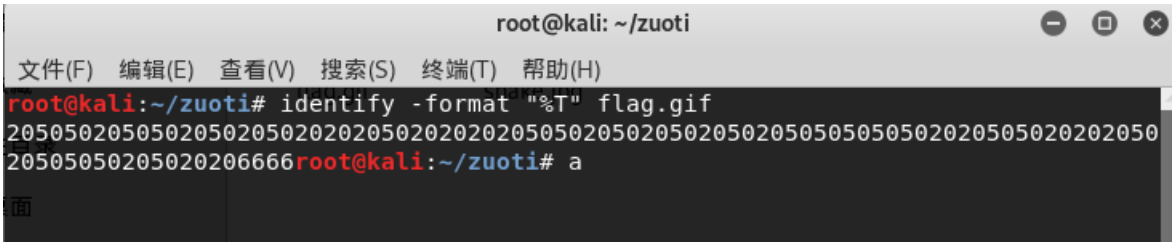
```
$$START$$-----BEGIN CERTIFICATE-----
$$START$$UEsDBBQAAAAIAKmQTEwlsC84WTUNAK5GDQAIAAAAZmxhZy5naWZkvFdUE0zbBZpO
$$START$$Qk3o3dBDNfSuCb0besfQqxgQKYKaRi8GBAUEDU2KiEEBUQFD70oTAVEDAqKigoJi
$$START$$eznff87lWWsuZs3FzFp75nlm7ynb1t7GyDh4AWgGyD4EiPHLSoqpKEiryYkoiaMU
$$START$$5I8dV5HHYlRMDNSNjbTNjPSsrQ0trUydHU86EWxO+VtZ+9s6+jq6RzsQAgheMfb2
$$START$$MZaWMW6+MR6usQGB/q6nY05HnotISI8KT48/d/nc+YxIUgoxLNoj1OGElZWuuba2
$$START$$tbGxi6Who5WWqZ/hSRUNa2kFEwkZrKa0mo2Crq2Ikr+iui1a2UZA2l9YIVYOc0EC
$$START$$ky+vU2nhwNQ80XISx8I5s1z9BmzcR4mhL6NIX8iZc6mpIx5Blc6+deEh9VFna1Oi
$$START$$a9JS6jOTC0O9L57yuEhwvHzCId8Ul2+kX6mo42vl5ucR7ut5JoaYEBeZRIohX75Y
$$START$$eJFcVZh/6yazq7HtaUvP677JrenFT8+Wdxfefn/1dldBSkVLzTQkICLvQmbRhdTj
$$START$$+s7+GkbYY1q6wkqxQopx8jrzWW59qTEX3YghxKTQCHJYbG4Bo7Xp4eLw/KfVd/v8
$$START$$AopYRWzFRbq9yYkMBd2xSIOWMO+AQNK5hNyE8yVXKju6Hw7dH1zTUdG21tK11Tep
$$START$$yMqytvRrCsYXRIUWRsbfrb+np65rp2fghz9ZeJaEFTymwyeRFhwoLKZlpaqHP3FK
$$START$$Fx+INfK0UdR4EnF29dV2XWEuqzj/euGV2OPG2vJqaefIhbT8y3m3G66VtN285UA4
$$START$$Y2sf5hCcc/Pes1dr3x5Ul7CZN9OCPCZS7TvSMt9sHbRlZRWlnKnAGd9oGDE56ecR
$$START$$f/NGSS277o5vcgsxo9XB4/xYdpKdJSGM9iDpUrlfLMMjICmWemf5UuYMo0BbRsXZ
$$START$$McCFEP6NUawtpZChp7fRlPSpqvR5c8lINcPjNBkjopRhivt28+Z+R+uf8f7FmVkF
$$START$$1LE/r3omO+8crMy96uv72/tkb31NkVdClFvsy8evyxODo71Pebilfv7+l1c1yM0l
$$START$$+JjVlnH1aWjCNd8Q8rf6xh9jBQeHf2PMTbZu+bz7+GPj0682e5WPX3592v37Yefn
$$START$$zu6vvf2/u99/E8KLEnPu36puPPz9X83DVxX3FgrrpsiVI0mMp3F5TwAAgNwRj4u1
$$START$$h7slnmCto4UFA/+v6RcEKgw4Amj8rw74vzABwI8AQAgSiWaTqRC4cMAYi9NPgUCR
$$START$$GmgmDgjmQioSWOR3IBifhUkqAGgBQQgpNpXiasIzuYWJXXGP3KhQhBjJr5wlNEjn
$$START$$lcAChsl4WhaftEnqLrNjkAuJ1qMzGHIWYzB5VrJ0DRr4shAxoIzGW1CoglHOTYFP
$$START$$cJ+pD1WcmD53uuig+6oe268nU77miWj4dIivDljni2khn5CAodoFA/kKT3Lwy5m8
$$START$$kzo6sde0Q2GFkvpxq+xiy1F+WfMLh4drlp2fFs3IgOXiOgN+KTwVjtKh26HkbchC
$$START$$cvIb8BJFkVLM6fsbI4LShomrWZfewa5KOd98NYMaggkq2TGMjhIE5QeFMK71dhU6
$$START$$mQ4i6t6ssNn7mTzXtX4+Sv0w8PTy0d9DqDiafMSXc+si10ojh8qrPIamAmyiOPSf
$$START$$CJ+1zL/2TTWXpZO317L/8Q0RLLhVF2YsINDdzByUa1gxkEhuR3mfzQWUgPlKw0og
$$START$$Jt8OcVCBvzkWTsgeK7wzZi+1H+Lowtk4hc90XYA5pKGzxEOJqiNux0hMoIOQEYoC
$$START$$NspjvQAhmMvFPopowjVQWDkTEBje5NmIOP/s/EBA7LK1BXfawYuX3GrIGDIgohy/
$$START$$FKw5BlgOyZPYGpC46uxVEV4e/sMiIqp+z5dvuvjaID+OuVgpcKJpzS/GfIejFl0P
$$START$$PaoZjO9BtY+ctVl1GEKNKNVAoBPL6J2oThF1v/Mvlgl+/Pe3RmBScj6A0rRTVrNZ
$$START$$wvFOe1nppCGxbNHEOLlsqqgbBn3pj0hO8KUg3CGHJnTBZ43yXrC7jSJ4k2e972+y
$$START$$tu3lo9882MsIOatai6TyVHlL1F00ylpEhsAlDo8SFrWNyMOWCCl/8BGECjDROtx4
$$START$$kMQG2HYR0g+VuaGHh7j4oRtgnhHR2eKGr7RHnsmQWJRTKIjzbbC8cDVbhptkXecT
$$START$$UZHphUgab1gMyHgjcd0v4p/q9mK6b+k2MUtEtDks39VkWaK0wXw5PC52S4IvdYHc
$$START$$2tRw+9rH5tYtuVf8rDdHI9WyT/girdr6+UY/ofqUUfT1V6r+O+9m1nCV/A2IGn8k
$$START$$2oznbsemeXt259LjfttjX39Rho8lACzeiv7X/io488MPsfRP+1s/2z4fbvWt9fxL
$$START$$HEv/+vSnVsZlRRgONwCcCc7I5ZHSm+FWd9jftSZC3NptqKmduFl5JisBLJedwcX9
$$START$$7sqVAbT8LWlFiGLz14QZM2fCApfsThyX3OH7e9/FZZPGgTDT6JKraFyHqTgJmJI6
$$START$$Ceqb02vCn4lRJllAJglwYTtCjLcu5RY/EtP2R/w4jNQ/Em5pdF8EdibaUepVC2VC
```


去除首尾两行，将base64解码以字节流形式写成zip

```python
import base64

with open('data3.txt','rb') as file:
    with open('res.zip','wb') as new_file:
        new_file.write(base64.b64decode(file.read()))
```

终于得到gif：
这一系列操作实在看的我头疼

然后是时间隐写？？Ubuntu下使用identify



20替换为0
50替换为1

01101101010001000011010101011111001100010111010 0

```
>>>int('01101101010001000011010101011111001100010111010 0',2)
120139720634740
```

```
>>> hex(120139720634740)
'0x6d44355f3174'
```

```
>>> binascii.unhexlify('6d44355f3174')
b'mD5_1t'
```

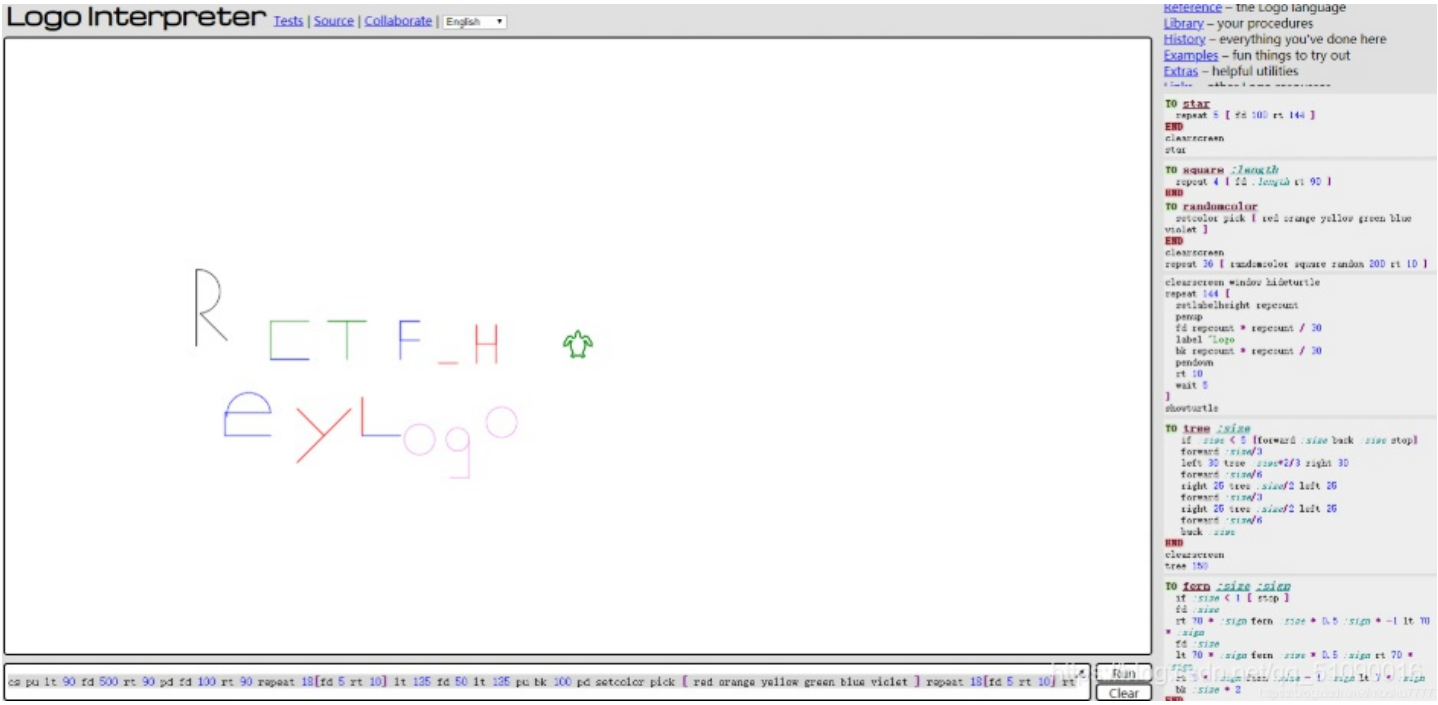
```
>>> hashlib.md5('mD5_1t'.encode('utf-8')).hexdigest()
'f0f1003afe4ae8ce4aa8e8487a8ab3b6'
```

flag{f0f1003afe4ae8ce4aa8e8487a8ab3b6}

[RCTF2019]draw（logo解释器）

将那些字符复制到这个网站：https://www.calormen.com/jslogo/

如图：



[MRCTF2020]不眠之夜（montage和gaps）

拼图，把最开始的文件和中间显示不了的文件去掉，120个，每张都是200 x 100，应该长：10张图片，宽：12张图片，那么拼起来的总图就应该是长：2000 x 宽：1200

放到kali里，先用montage：`montage *jpg -tile 10x12 -geometry 200x100+0+0 flag.jpg`

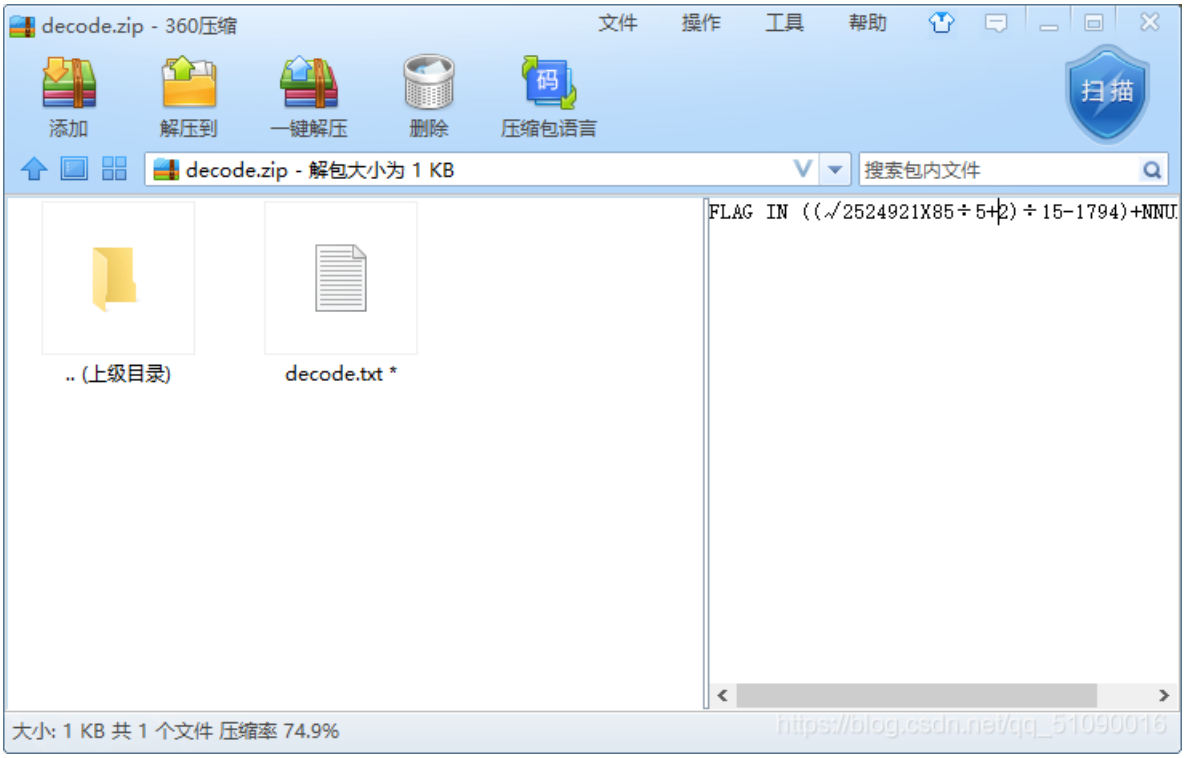
再用gaps：

```
gaps --image=flag.jpg --generations=40 --population=120 --size=100
```

为什么我费劲九牛二虎之力安装完还出错了？？真的给我弄吐了，还百度不到原因

[安洵杯 2019]easy misc（盲水印）

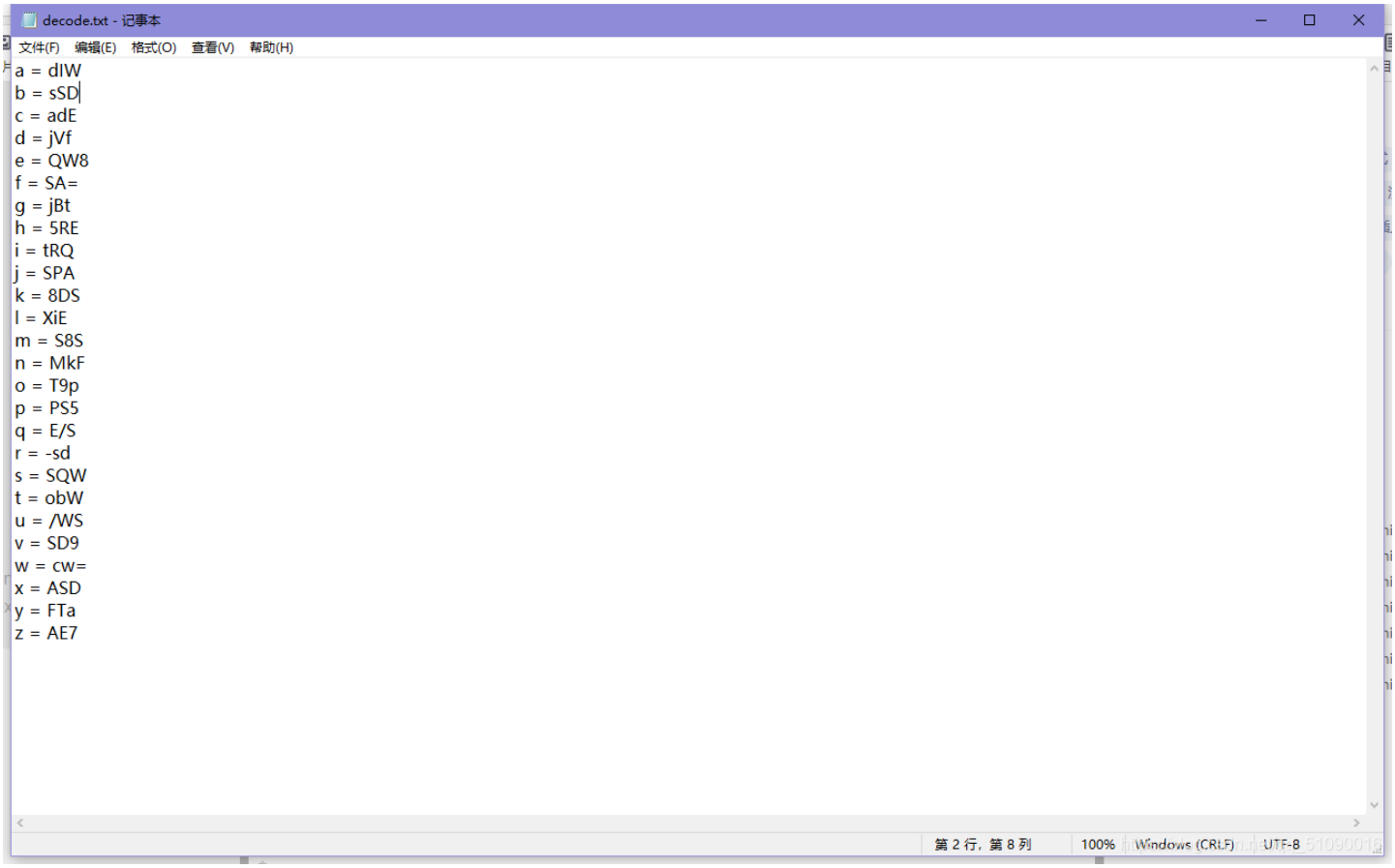
看压缩包：




```
FLAG IN ((√2524921X85÷5+2)÷15-1794)+NNULLULL
```

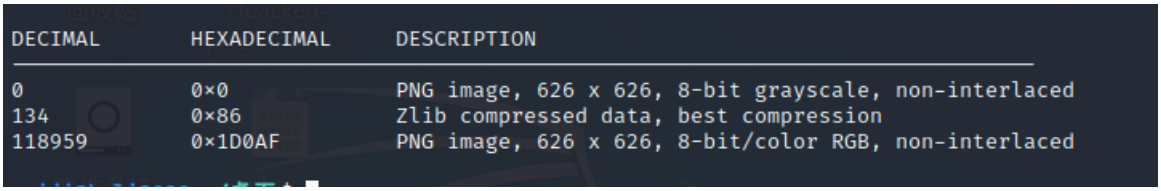
前面计算得7，后面猜测是掩码

得到密码，打开解压



这又是啥。。果然做题全靠百度，原来是字频隐写
按照wp的做，先看png

binwalkt一下



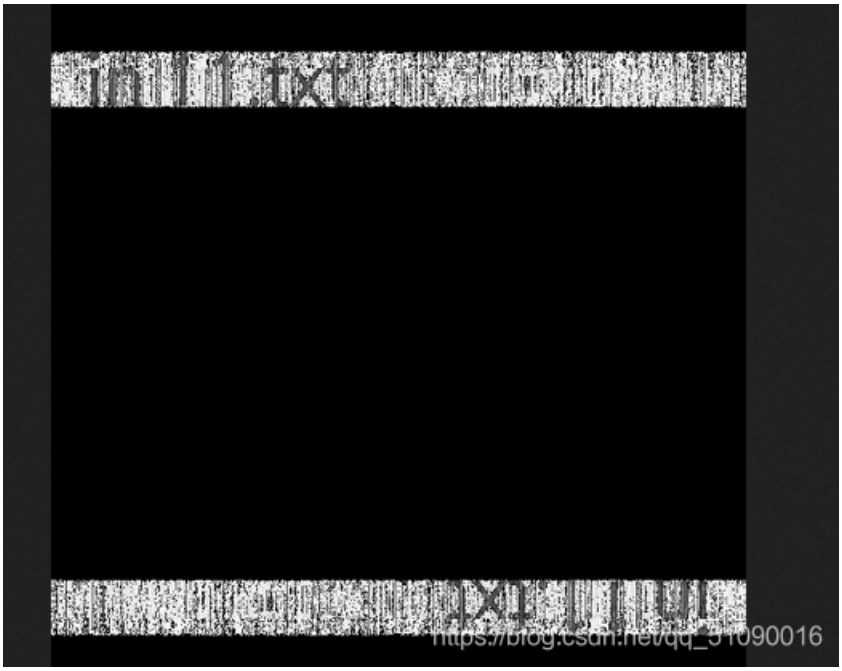
还有个png，foremost一下

两个一样的图片？想到盲水印，安装脚本地址：
https://github.com/chishaxie/BlindWaterMark

输入：

```
python2 bwm.py decode 00000000.png 00000232.png output.png
```

得到



提示in 11.txt（所以费了这么大力气就这）

hint中提示取前16位，利用python脚本获取11.txt中前16高频字母

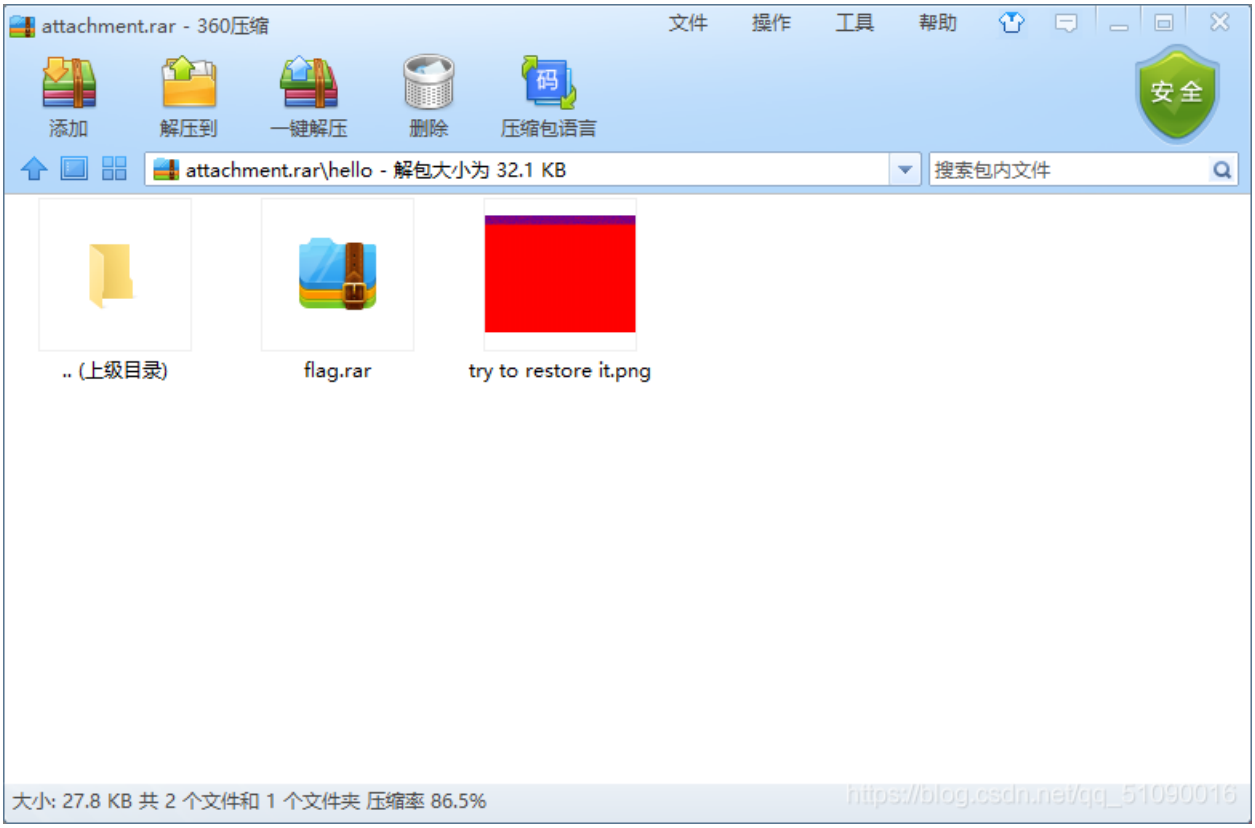
```
import re

file = open('C:/Users/Fiona/Desktop/11.txt')
line = file.readlines()
file.seek(0,0)
file.close()

result = {}
for i in range(97,123):
 count = 0
 for j in line:
  find_line = re.findall(chr(i),j)
  count += len(find_line)
 result[chr(i)] = count
res = sorted(result.items(),key=lambda item:item[1],reverse=True)

num = 1
for x in res:
  print('频数第{0}: '.format(num),x)
  num += 1
```

即：etaonrhsidluygw
对照decode.txt里面的数得到一串base64的编码：
QW8obWdlWT9pMkFSQWtRQjVfXiE/WSFTajBtcw==

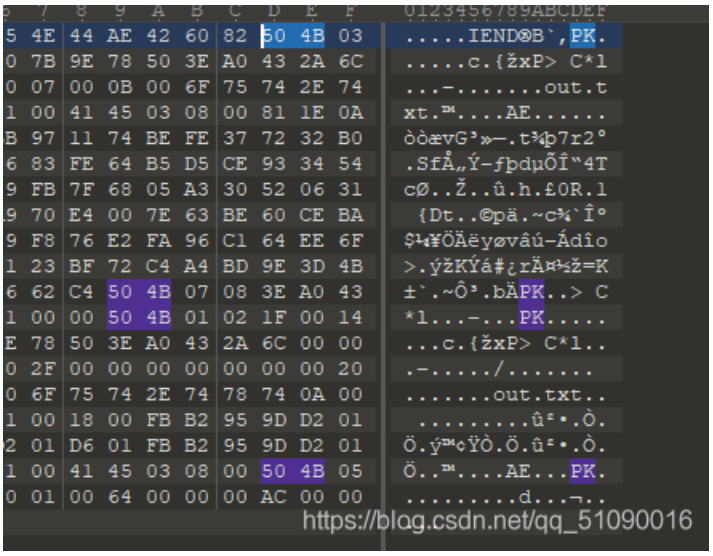
解密后
Ao(mgHY?i2ARAkQB5_^!?Y!Sj0ms
这是base85

得到
flag{have_a_good_day1}

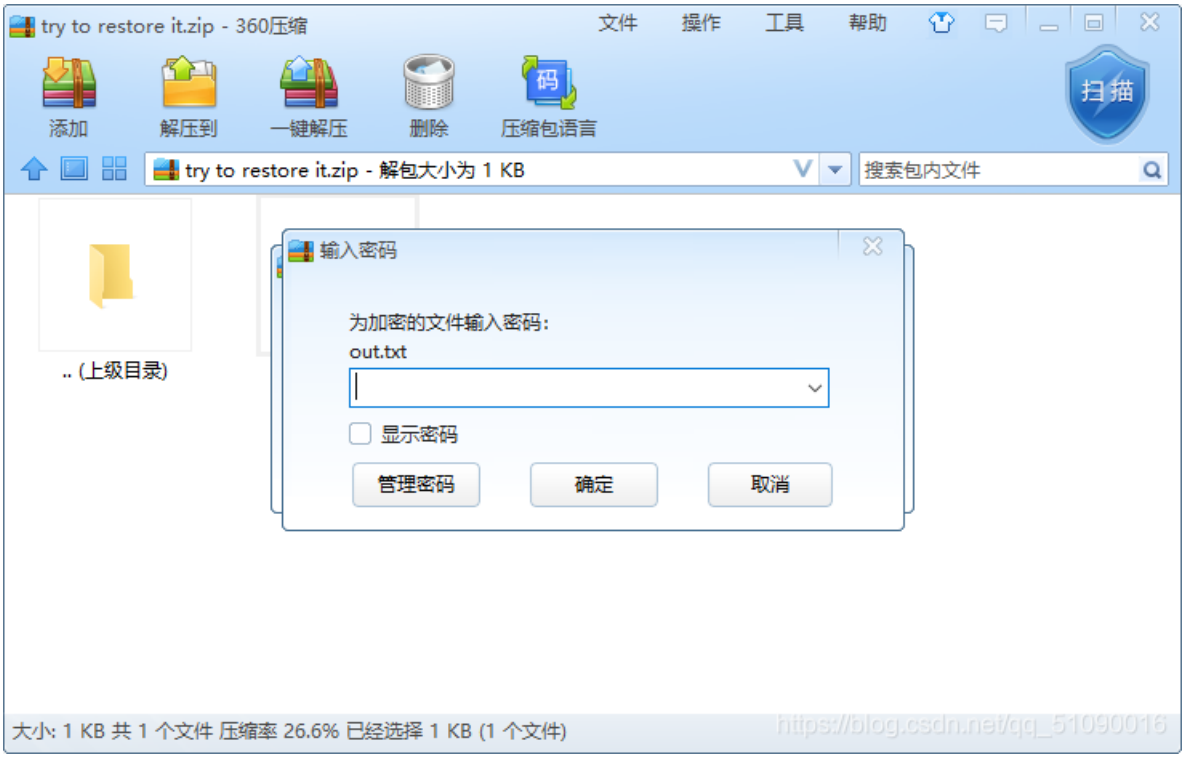
[MRCTF2020]Hello_ misc



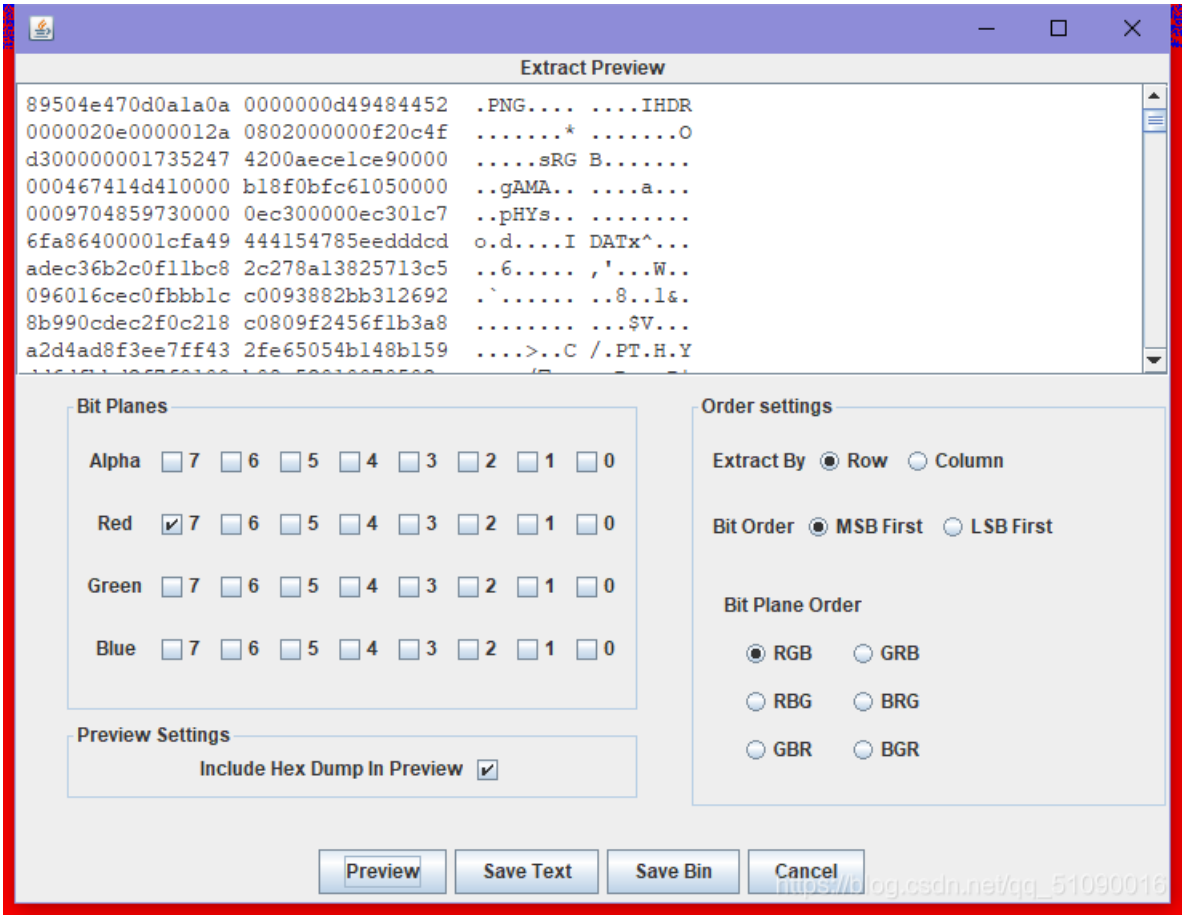
正常先把图片用010打开看看：



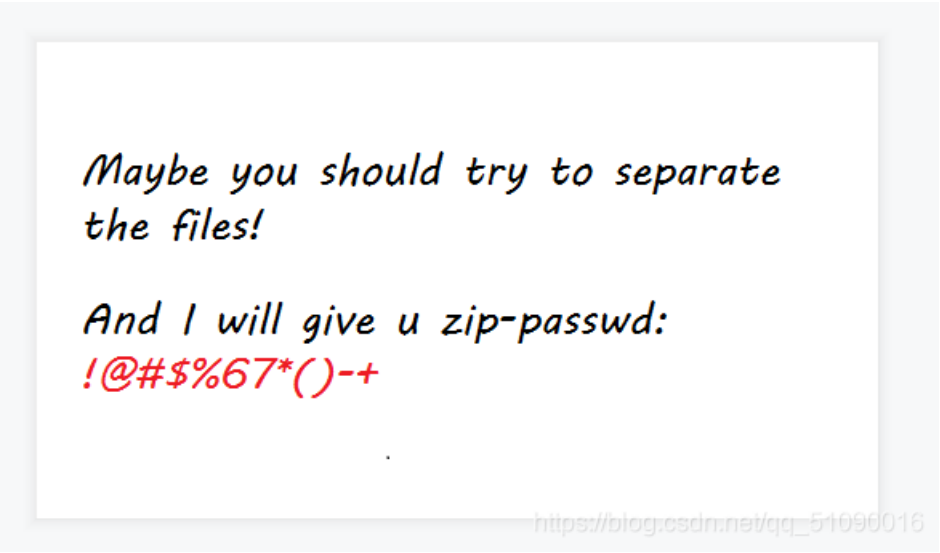
发现有压缩包的痕迹，改成zip看看



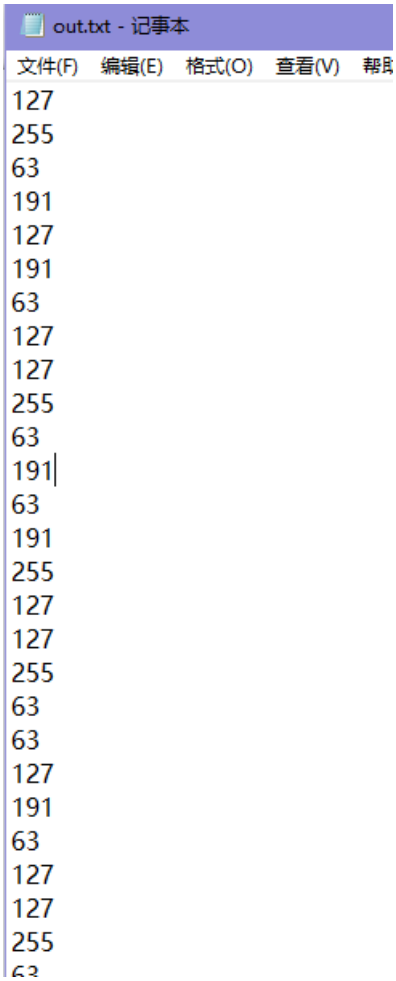
需要密码，从哪弄？应该是原本那张图片，那个颜色就让我怀疑是lsb隐写
这题和之前的不一样，在red通道发现图片（难怪图片是红色）



另存



得到解压密码，解压



几个重复的数字，之前好像碰到过，又有点忘了，刚好复习一下

用脚本转换二进制提取前两字节，转为ASCII

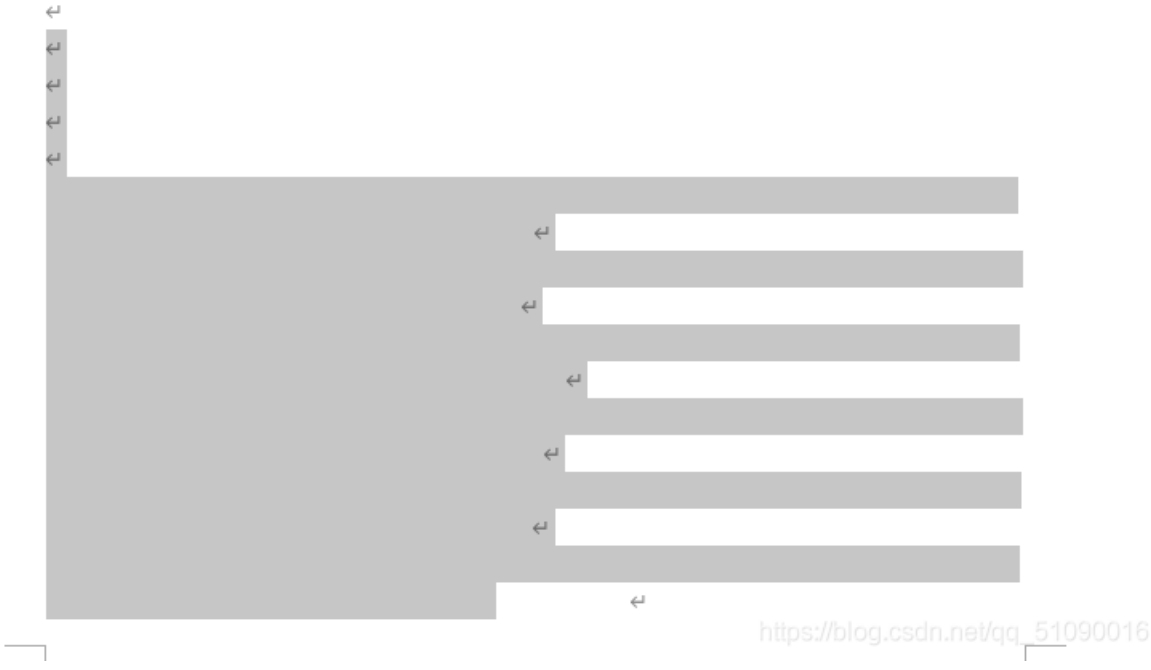
得到提示rar-passwd:0ac1fe6b77be5dbe

```
with open('out.txt') as a_file:
    content = [x.strip() for x in a_file.readlines()]
bins = []
for i in content:
    bins.append(bin(int(i))[2:].zfill(8)[:2])
stringBins = ''.join(bins)
num = 0
flag = ''
for i in range(int(len(stringBins)/8)):
    flag+=chr(int(stringBins[num:num+8],2))
    num+=8

print(flag)
```

```
rar-passwd:0ac1fe6b77be5dbe
```

得到zip，解压后猜测是docx

改后缀

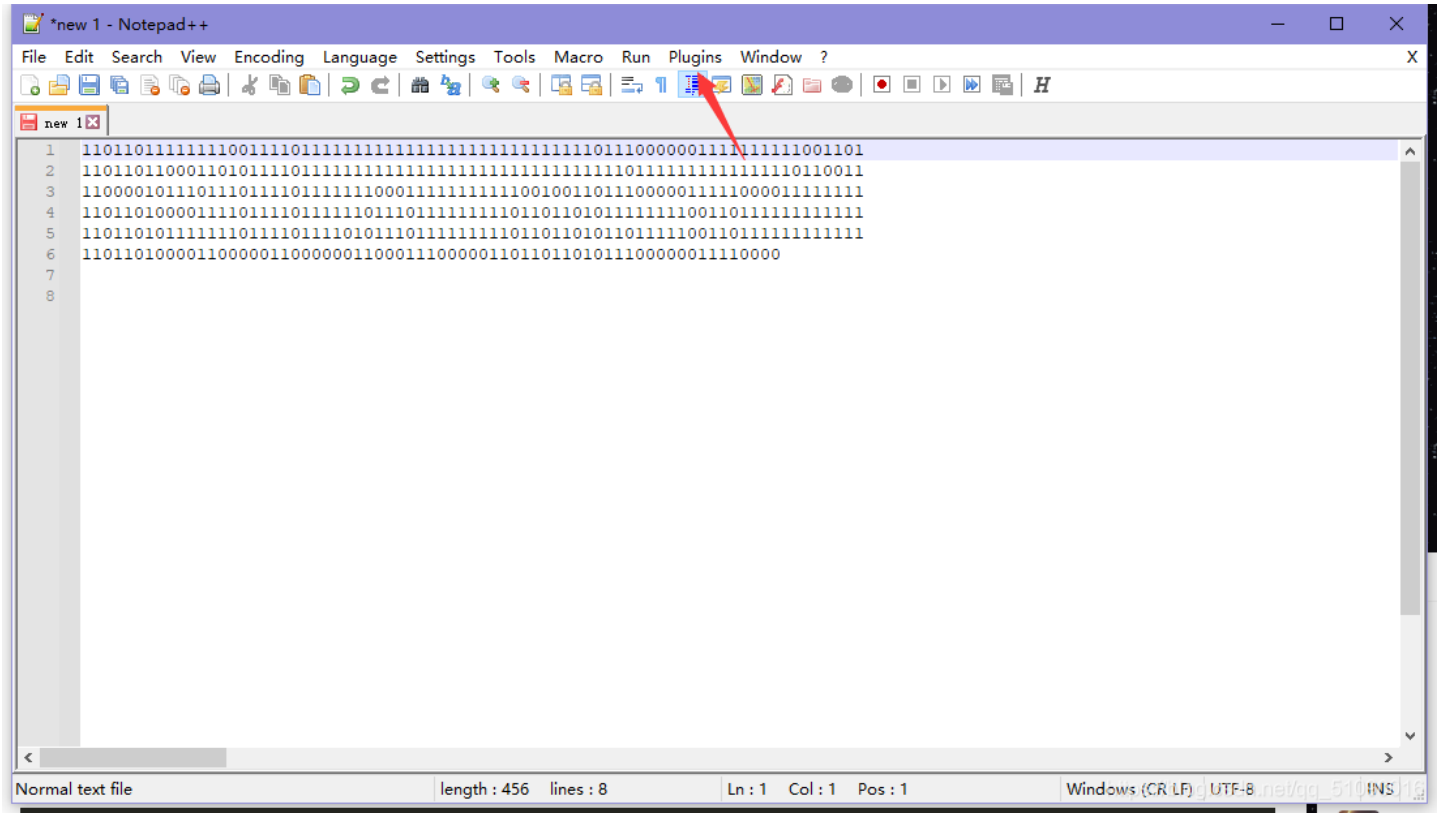


好像藏着什么，咋看不到，把字体改成黑色

MTEwMTEwMTExMTExMTEwMDExMTEwMTExMTExMTExMTExMTExMTExMTExMTExMTA
xMTEwMDAwMDAxMTExMTExMTExMDAxMTAx
MTEwMTEwMTEwMDAxMTAxMDExMTEwMTExMTExMTExMTExMTExMTExMTExMTExMTE
xMTAxMTExMTExMTExMTExMTEwMTEwMDEx
MTEwMDAwMTAxMTEwMTExMDExMTEwMTExMTExMTAwMDExMTExMTExMTExMDAxMDAxM

```
TAxMTEwMDAwMDExMTExMDAwMDExMTExMTEx
MTEwMTEwMTAwMDAxMTExMDExMTEwMTExMTExMDExMTAxMTExMTExMTEwMTEwMTEwM
TAxMTExMTExMTAwMTEwMTExMTExMTExMTEx
MTEwMTEwMTAxMTExMTExMDExMTEwMTExMTAxMDExMTAxMTExMTExMTEwMTEwMTEwMT
AxMTAxMTExMTAwMTEwMTExMTExMTExMTEx
MTEwMTEwMTAwMDAxMTAwMDAwMTEwMDAwMDAxMTAwMDExMTAwMDAwMTEwMTEwM
TEwMTAxMTEwMDAwMDAxMTExMDAwMD
```


这又是什么？？果然做题全靠百度。。使用notepad解码每一行base64编码，如图



观察0的形状。。。脑洞太大了：He1Lo_mi5c~

[BSidesSF2019]zippy（密码解压）

pca文件，tcp追踪流

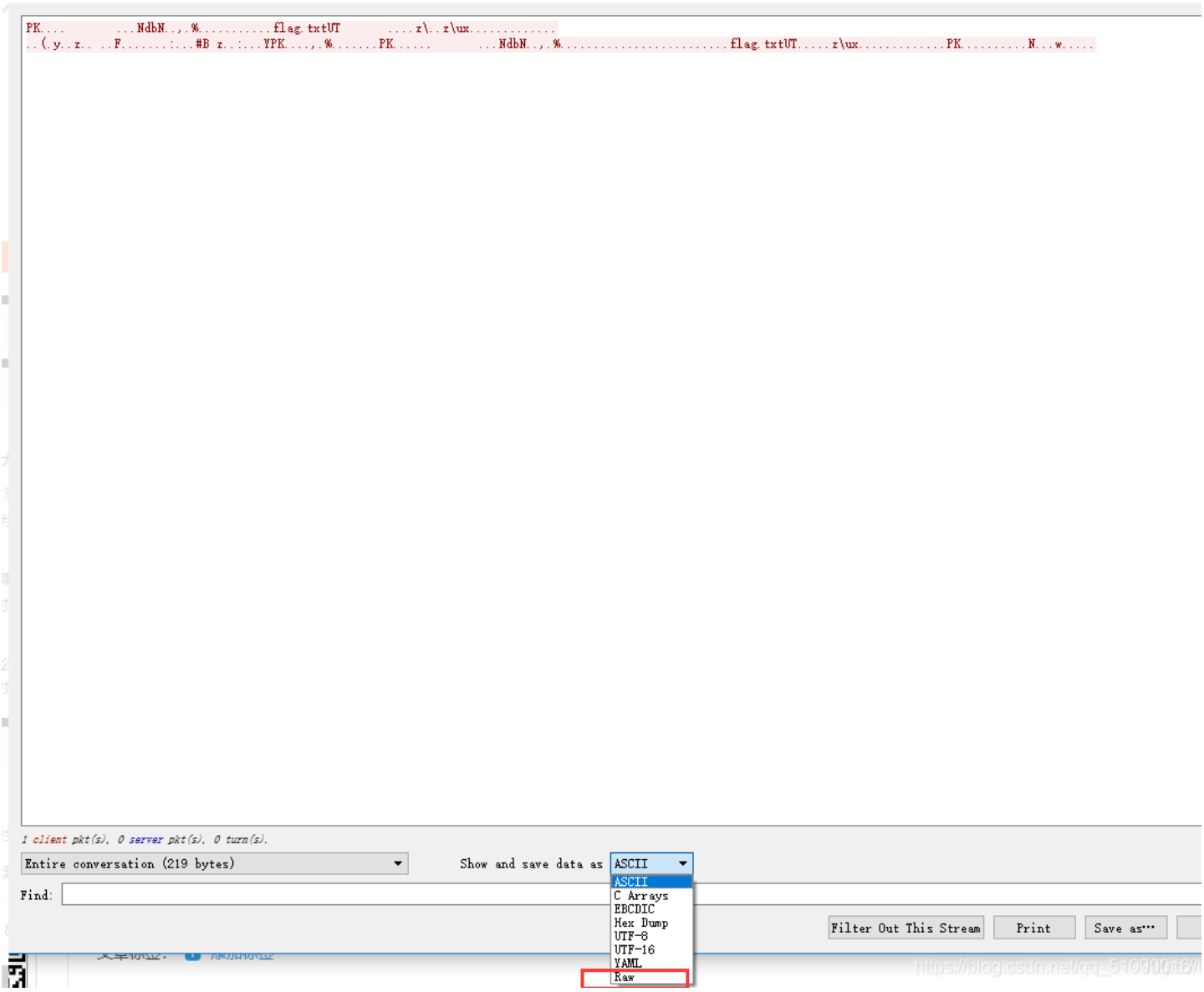
Wireshark · Follow TCP Stream (tcp.stream eq 0) · zippy.pcapng

```
nc -l -p 4445 > flag.zip
unzip -P supercomplexpassword flag.zip
Archive:  flag.zip
  inflating: flag.txt
```

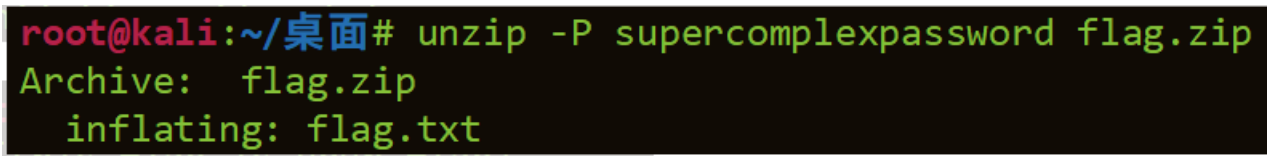
这是啥。。看wp吧

应该是正在发送一个加了密的zip文件，找到这个加密文件的数据流

以原始数据保存为zip文件



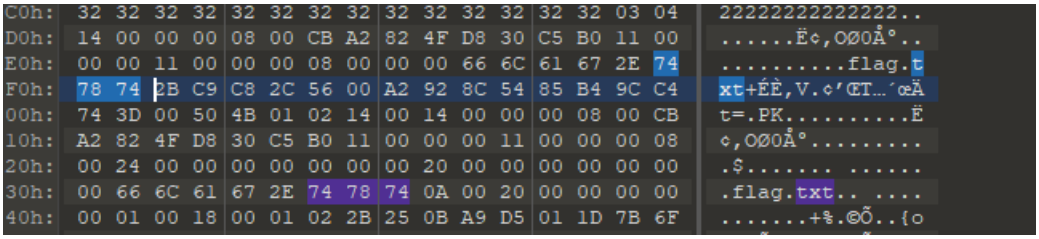
使用密码解压



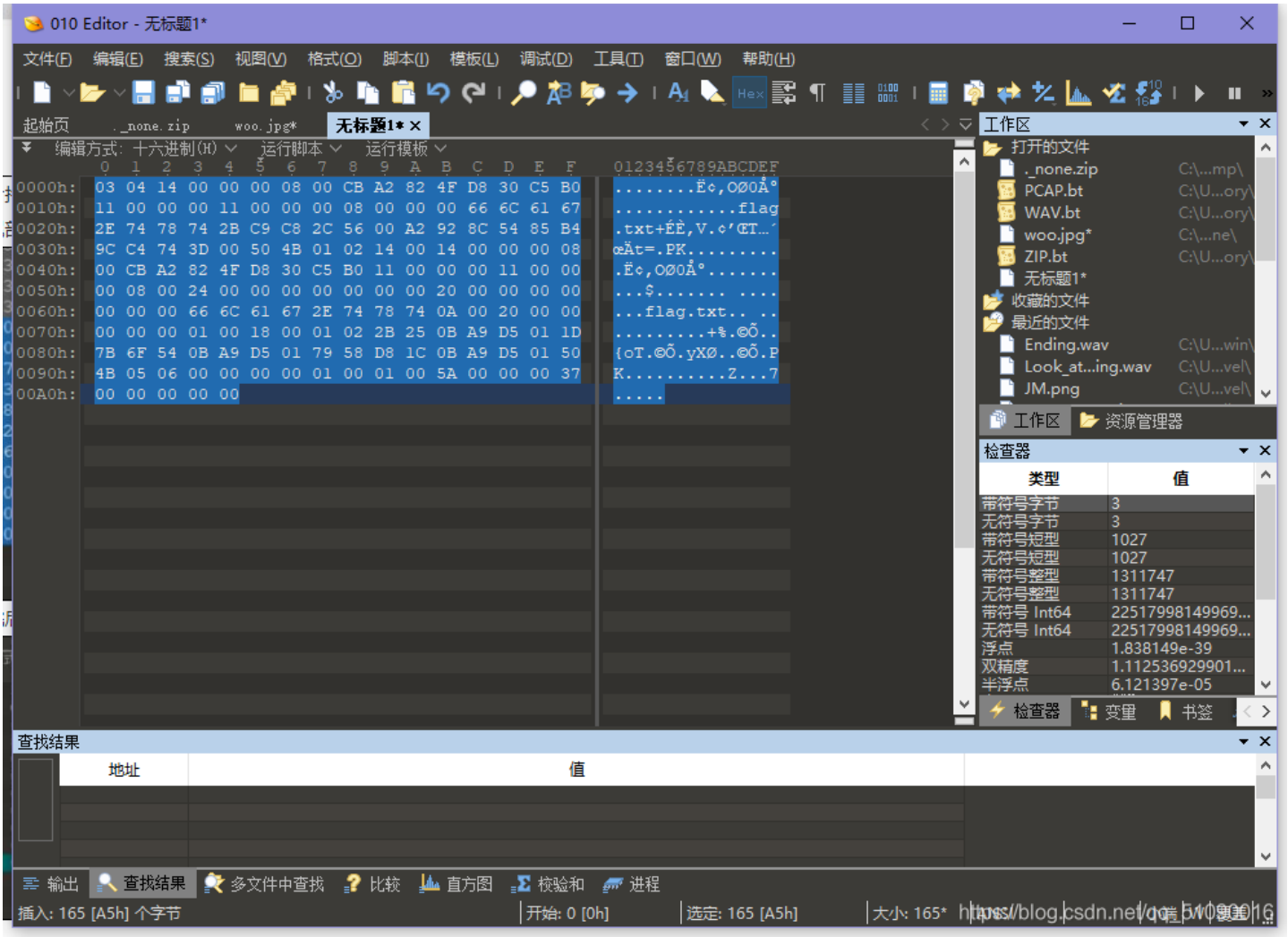
[ACTF新生赛2020]明文攻击(隐藏文件内容在010)

文件里有个压缩包和图片，压缩包里只有txt文件，根据提示明文攻击，估计是要在图片里找到一个txt文件。

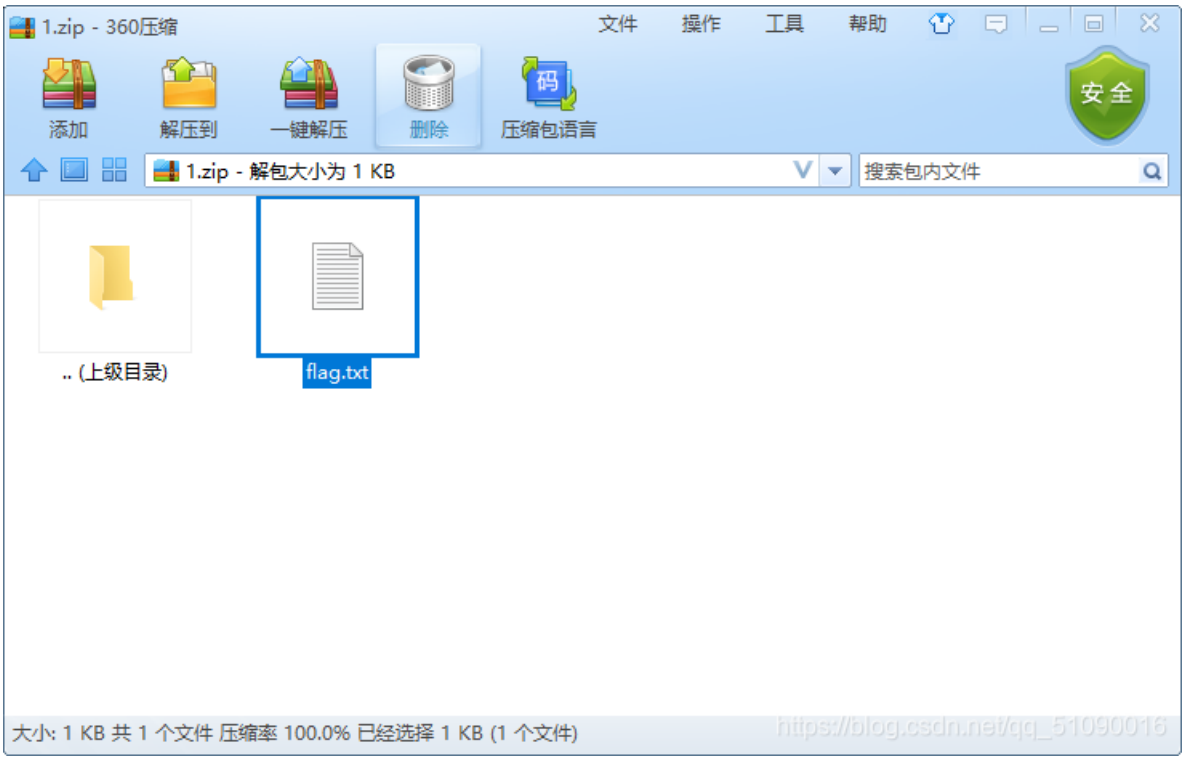
图片放到010发现有txt



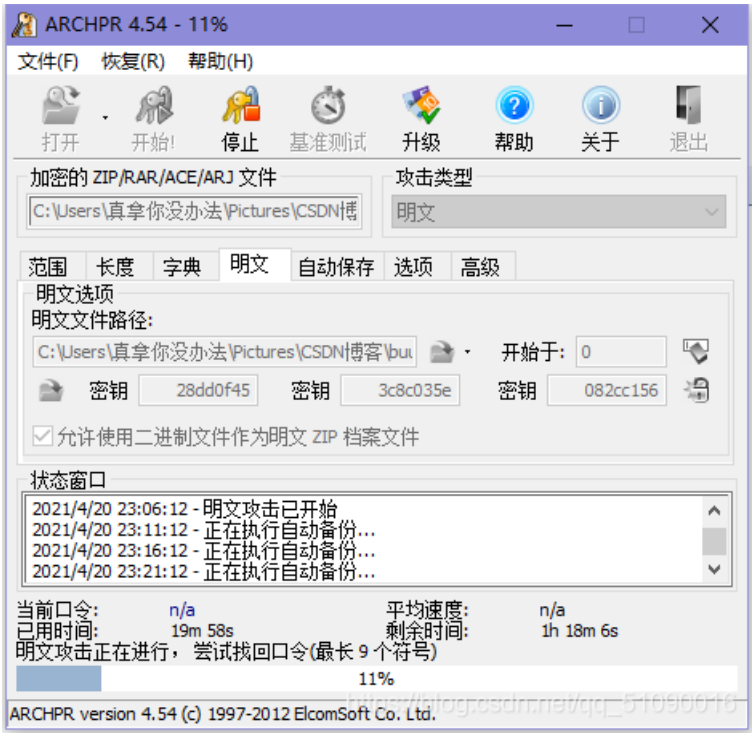
改成zip后缀失败。。那该咋办，看这后面的样子，像是一个压缩包，但是却没有文件头，试一下复制下来加个文件头504b：



果然得到了



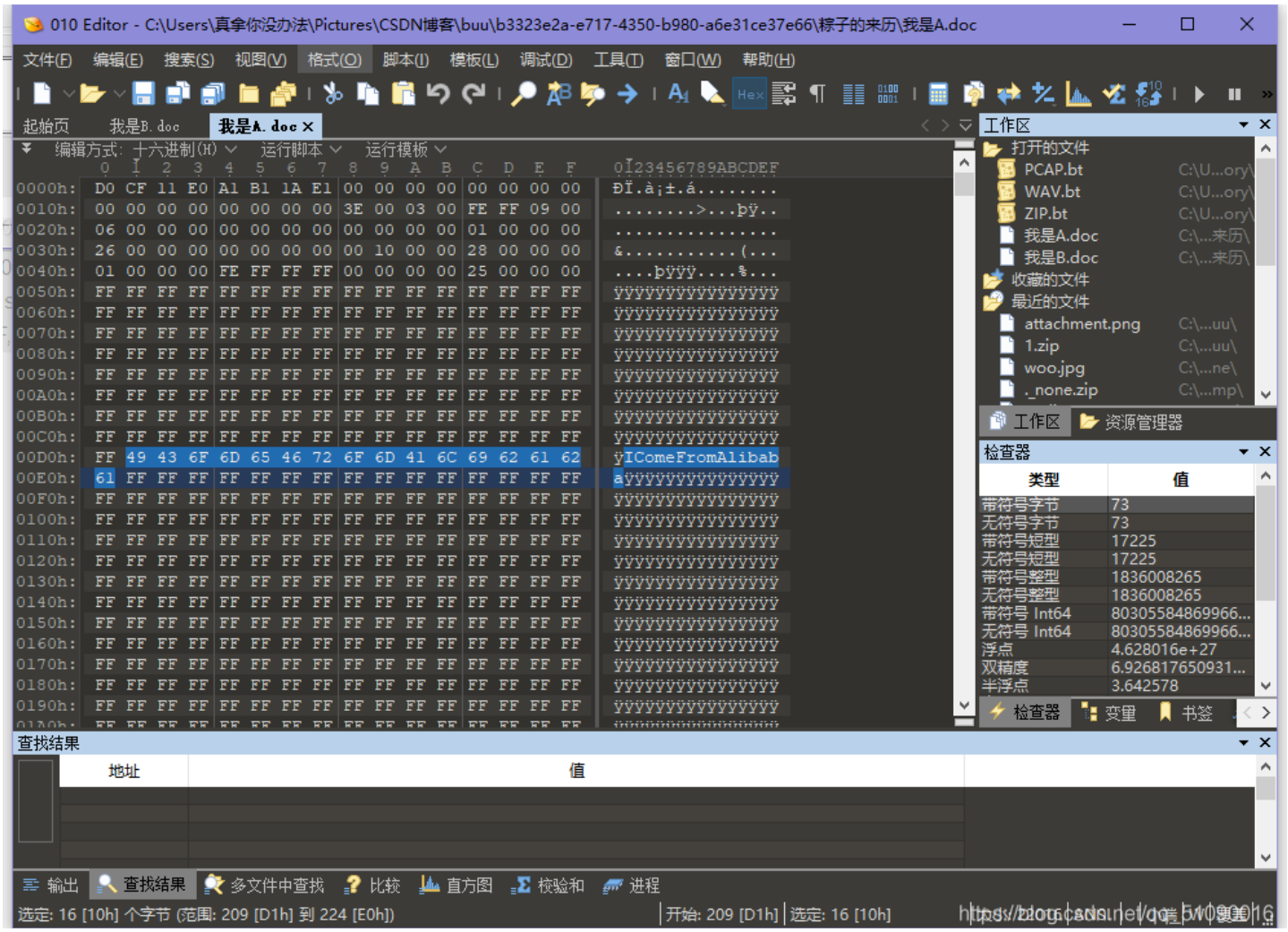
明文攻击



粽子的来历（脑洞太大 ff修复 行间距问题）

用word发现打不开，用010打开

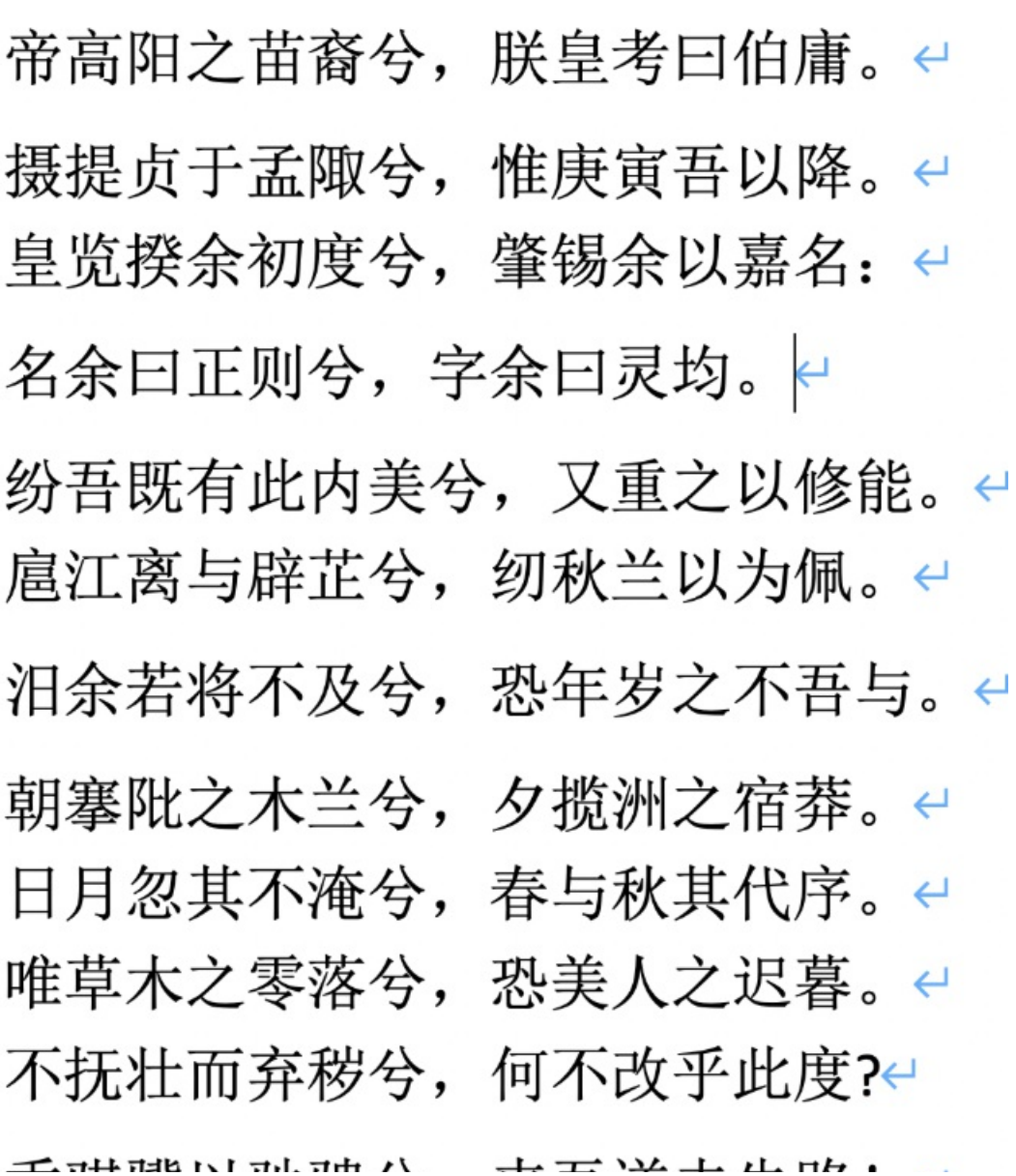
发现一段小段数字被F包围，尝试将其全部改为F，保存。



然后就能打开了

帝高阳之苗裔兮，朕皇考曰伯庸。↵
摄提贞于孟陬兮，惟庚寅吾以降。↵
皇览揆余初度兮，肇锡余以嘉名：↵
名余曰正则兮，字余曰灵均。↵
纷吾既有此内美兮，又重之以修能。↵
扈江离与辟芷兮，纫秋兰以为佩。↵
汩余若将不及兮，恐年岁之不吾与。↵
朝搴阰之木兰兮，夕揽洲之宿莽。↵
日月忽其不淹兮，春与秋其代序。↵
唯草木之零落兮，恐美人之迟暮。↵
不抚壮而弃秽兮，何不改乎此度?↵
乘骐骥以驰骋兮，来吾道夫先路！↵

仔细看，发现四个文档间距不同，想到MD5加密，设1.5倍为1,1倍为0，将每行的值连在一起，进行MD5解密（这也行？）
最后发现c是对的

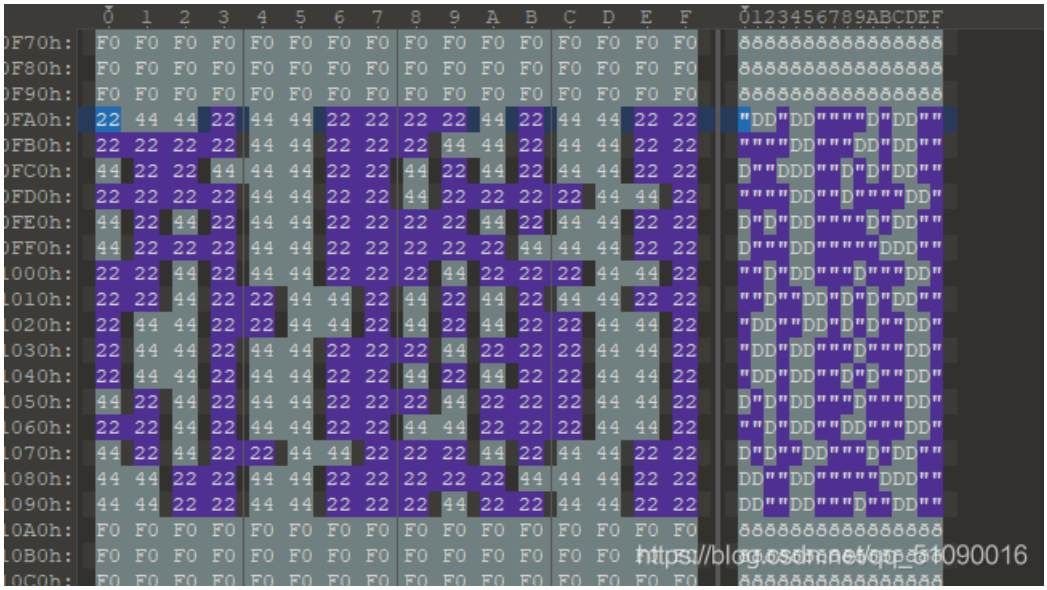


乘骐骥以驰骋兮，来吾道夫先路！

```
100100100001
```

md5 d473ee3def34bd022f8e5233036b3345为正确flag

派大星的烦恼（倒过来的二进制）



首先放到010里搜索字节22
根据提示
0x22，0x44 想到二进制

把右边那一段：

```
"DD"DD""""D"DD""""""DD""""DD"DD""D""DDD""D"D"DD""""""DD""D""""DD"D"DD""""D"DD""D""""DD""""DDD""""D"DD""""D""""DD"
""D""DD"D"D"DD""""DD""DD"D"D""DD""DD"DD""""D""""DD""DD"DD""D""DD"D"D"DD""""D""""DD""""D"DD""DD""""DD"D"D""DD""""D"DD""
DD""DD""""""DDD""DD""DD""""D""DD""
```

转换成二进制

01101100 00101100 00001100 01101100 10011100 10101100 00001100 10000110 10101100 00101100 10001100 00011100 00101100 01000110 00100110 10101100 01100110 10100110 01101100 01000110 01101100 10100110 10101100 01000110 00101100 11000110 10100110 00101100 11001100 00011100 11001100 01001100

尝试将其转化为字符
发现是一堆无意义的东西。。。

猜想是否 01 弄错 调换 01 尝试。。还是一堆无意义的东西

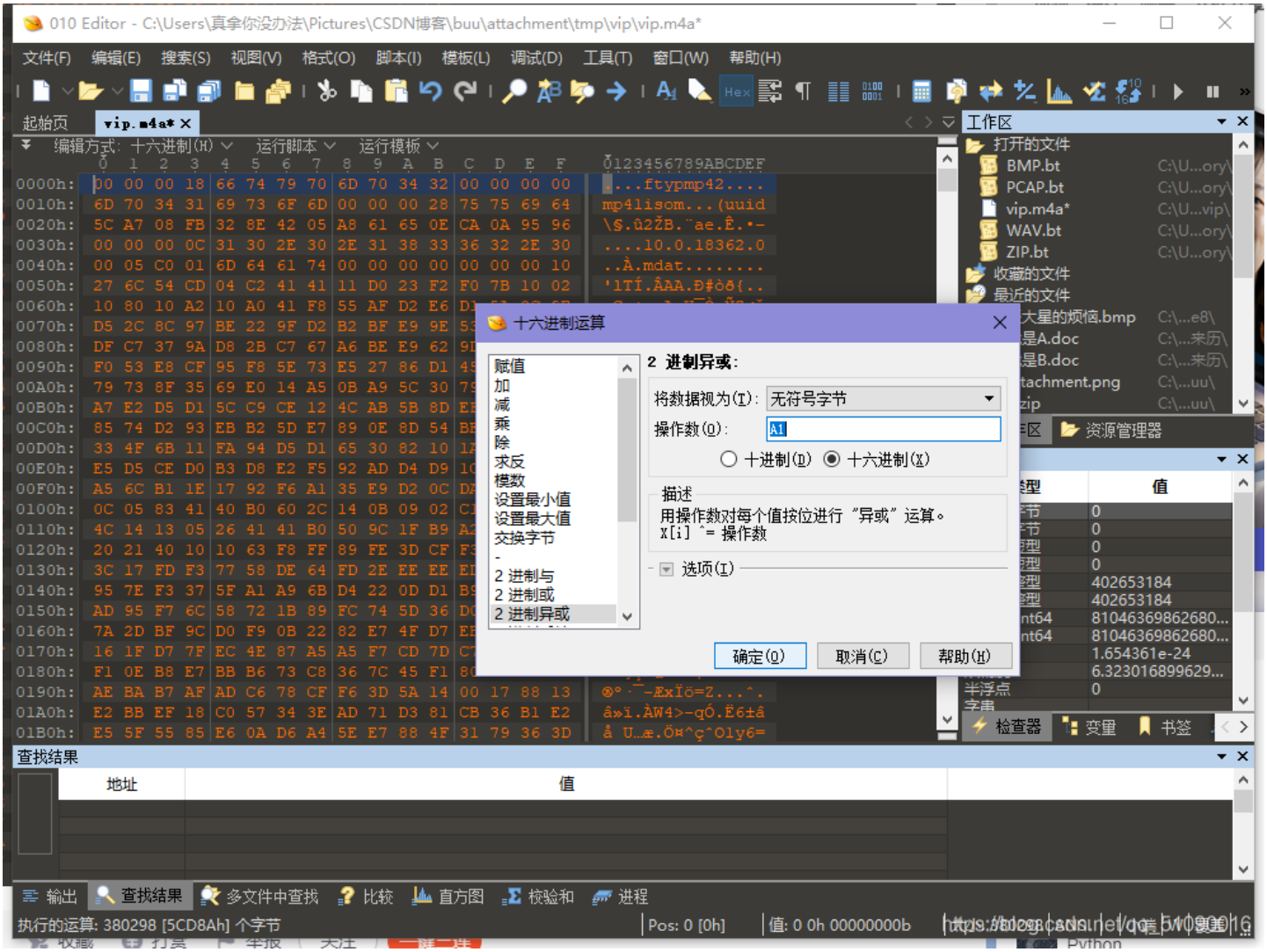
尝试倒序 将每八位倒转但整体位置不变

```
6406950a54184bd5fe6b6e5b4ce43832
```

[ACTF新生赛2020]music（对原文件进行异或）

先用audacit发现打不开，用010看看。。实在发现不了问题，看wp

猜测对整个原文件进行了异或，使用010 Editor在工具->十六进制运算->二进制异或对整个文件内容进行异或



保存后仔细听就得到了

[SCTF2019]电单车

放到audacity里面看看

细的为0宽的为1

0 01110100101010100110001000 01110100101010100110001000

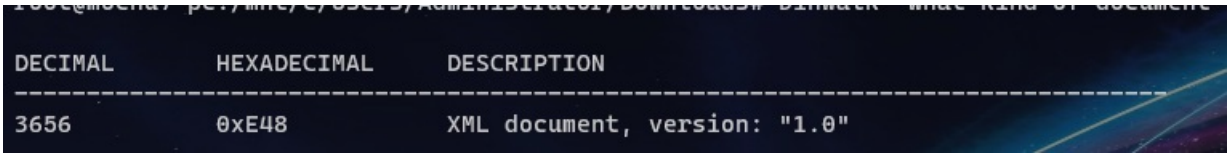
根据wp

钥匙信号(PT224X) = 同步引导码(8bit) + 地址位(20bit) + 数据位(4bit) + 停止码(1bit)

地址位长度为20bit，后4位为数据位即**01110100101010100110**就是flag（咱也不懂，咱也不敢问）

hashcat(爆破ppt文件)

010没看出啥，改成zip也没思路，binwalk一下



XML文件，添加文件后缀为ppt，有加密

使用Accent OFFICE Password Recovery对其爆破密码，老样子还是猜测密码为四位纯数字

得到密码9919

使用密码解开ppt文档，在倒数第二页有全选中标红即可（这个真想不到）

[UTCTF2020]zero（零宽度字符隐写）

又是没遇见过的类型。。。

打开看了看没思路，只好看wp

先看看什么是零宽度字符隐写

> 大意是，可以通过零宽度字符来隐藏一些信息，达到一些目的，比如隐写，或者水印。. 某种程度上来说，我们这里所说的零宽度字符可作为识别某些用户身份的“指纹”数据，也可非常方便地追溯到某些秘密数据的泄露源。. 然后分别插到载体消息中的每一个字符后面。. 若载体消息短于隐藏信息的二进制，那就把多余的都塞到载体消息的最后一个字符前面，这样在复制的时候不至于掉少一些隐藏字符.

那该怎么分辨呢。。

比如在下面这个例子中

```
"大佬早上好呀".length
34
```

我们能看到的只有六个字，但是显示的length有34，零宽度字符就会产生这个效果，他不影响阅读，但是真实存在，也会占长度。

虽然我们看到的样子是上面这个样子，但是实际上他是下面这个样子~

解密网址：https://330k.github.io/misc_tools/unicode_steganography.html

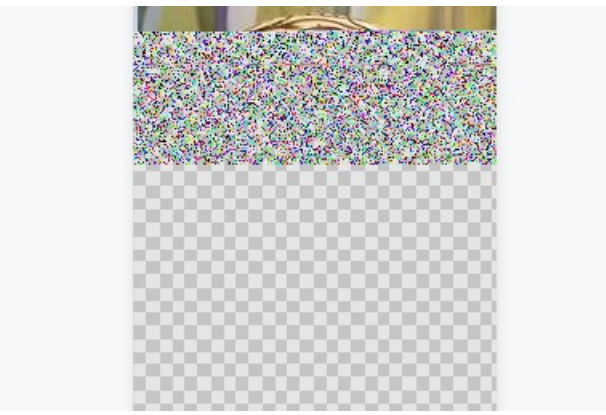
[湖南省赛2019]Findme

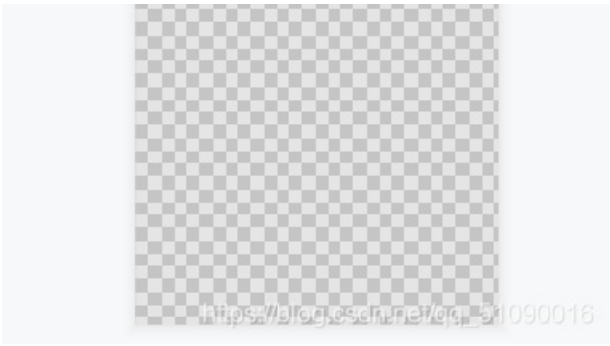
五张图片，一个个看

第一张应该是宽高不对

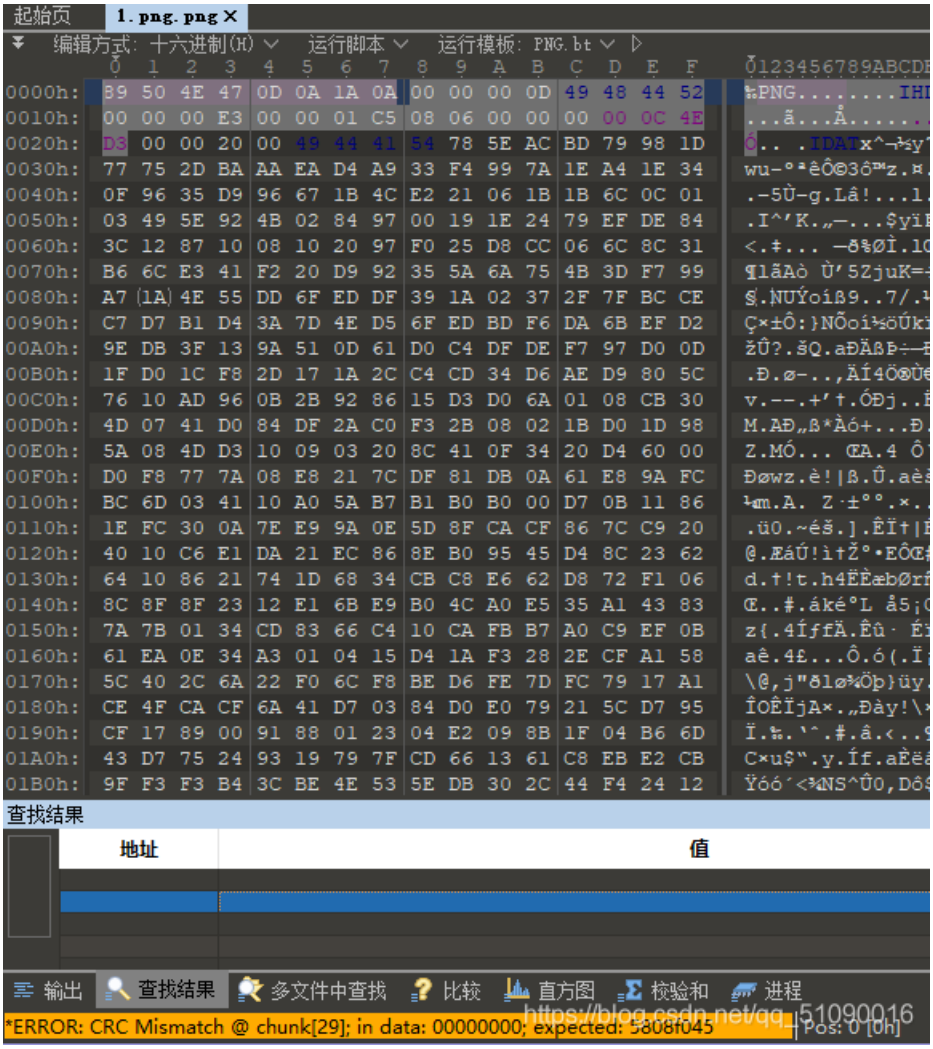
```
import zlib
import struct
file = '1.png'
fr = open(file,'rb').read()
data = bytearray(fr[12:29])
#crc32key = eval(str(fr[29:33]).replace('\\x','').replace("b'",'0x').replace("'",''))
crc32key = 0xC4ED3
#data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xF1\x08\x06\x00\x00\x00')
n = 4095
for w in range(n):
    width = bytearray(struct.pack('>i', w))
    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)
            print(data)
            newpic = bytearray(fr)
            for x in range(4):
                newpic[x+16] = width[x]
                newpic[x+20] = height[x]
            fw = open(file+'.png','wb')
            fw.write(newpic)
            fw.close
```

改完怎么是这样

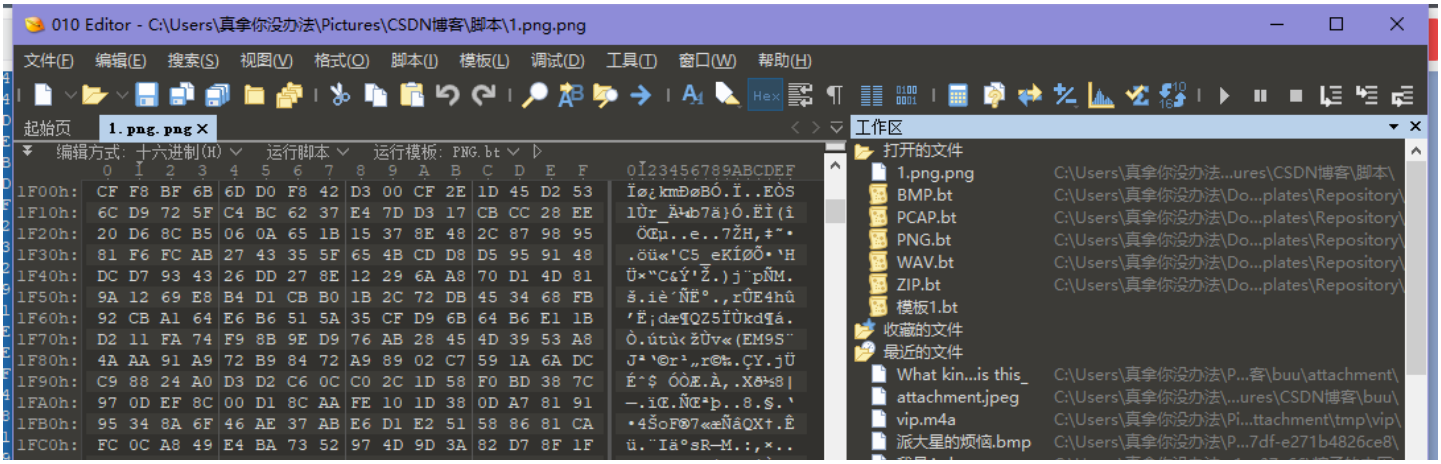


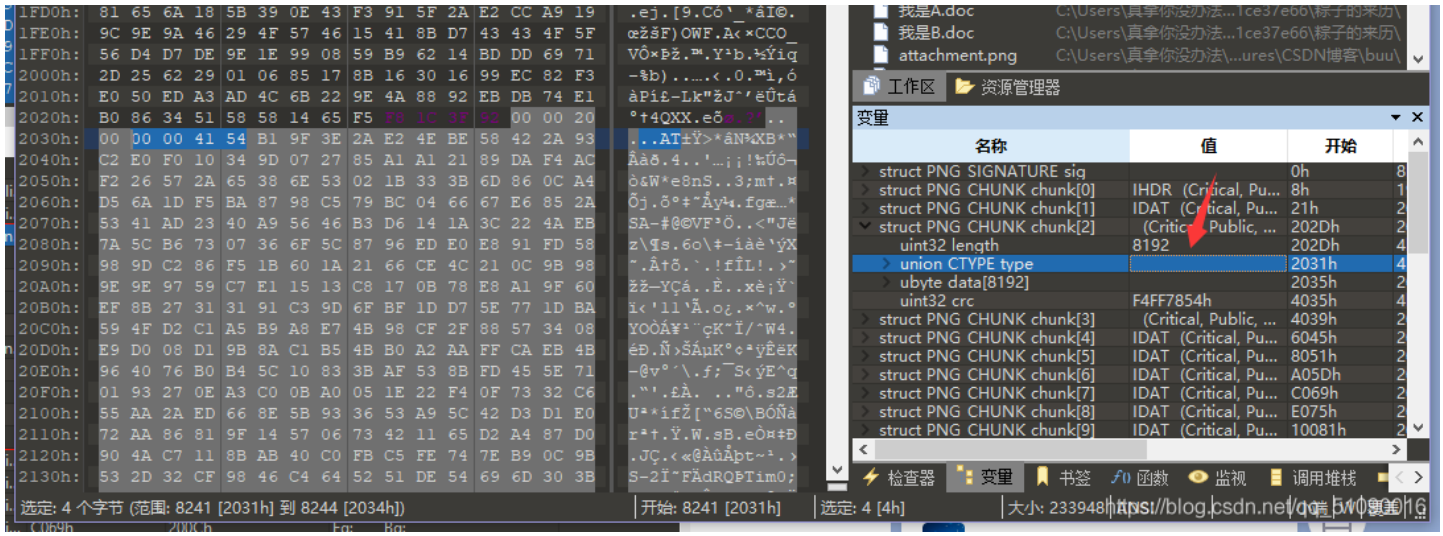


用010打开，发现错误

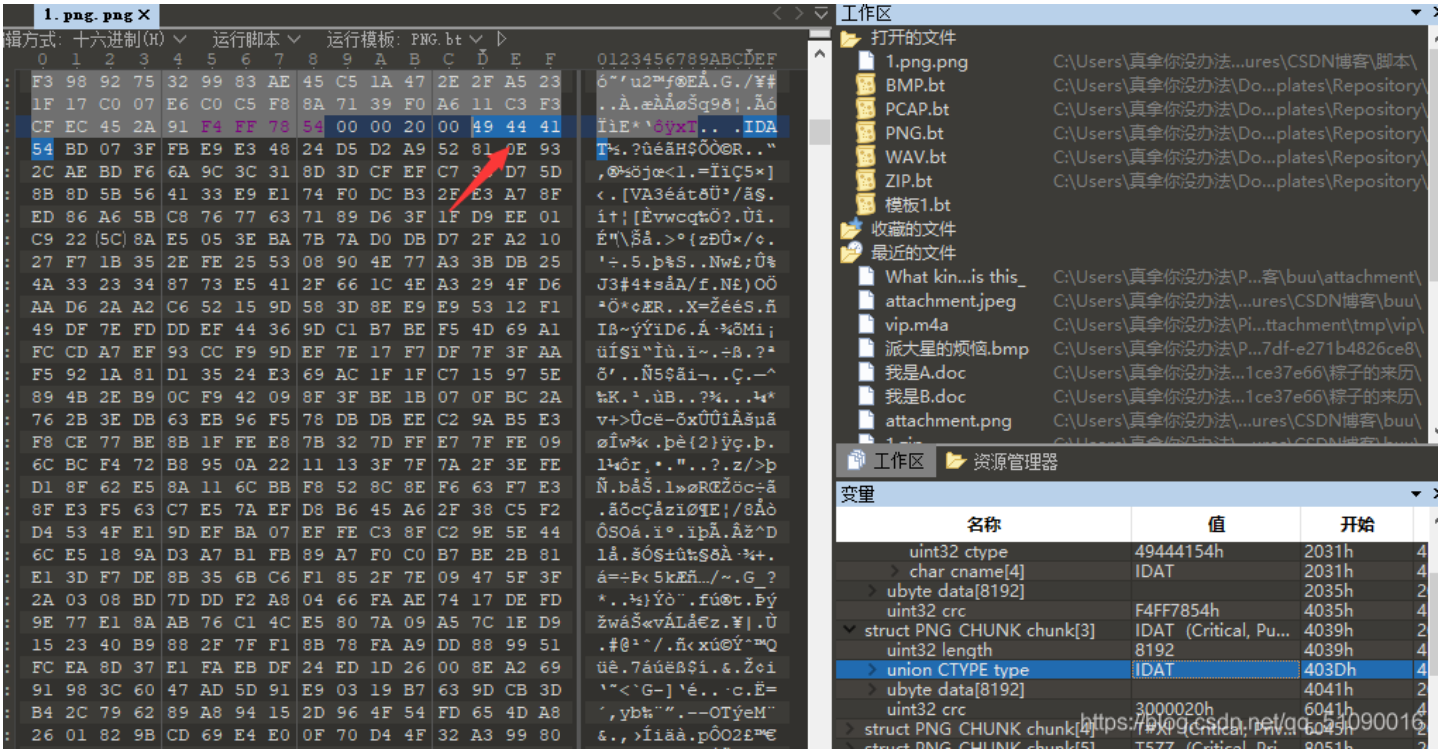


IDAT十六进制标识为：49 44 41 54，将两个chunk的IDAT在union CTYPE type的位置补上即可得到完整的图片




我是A.doc
我是B.doc
attachment.png
工作区
资源管理器
变量
名称
值
开始
struct PNG SIGNATURE sig
struct PNG CHUNK chunk[0]
IHDR (Critical, Pu...
struct PNG CHUNK chunk[1]
IDAT (Critical, Pu...
struct PNG CHUNK chunk[2]
uint32 length
8192
union CTYPE type
ubyte data[8192]
uint32 crc
F4FF7854h
struct PNG CHUNK chunk[3]
struct PNG CHUNK chunk[4]
struct PNG CHUNK chunk[5]
struct PNG CHUNK chunk[6]
struct PNG CHUNK chunk[7]
struct PNG CHUNK chunk[8]
struct PNG CHUNK chunk[9]
检查器
变量
书签
函数
监视
调用堆栈
选定: 4 个字节 (范围: 8241 [2031h] 到 8244 [2034h])
开始: 8241 [2031h]
选定: 4 [4h]
https://blog.csdn.net/qq_51000016

第二个chunk



两个改完就可以正常打开了

真的很杂

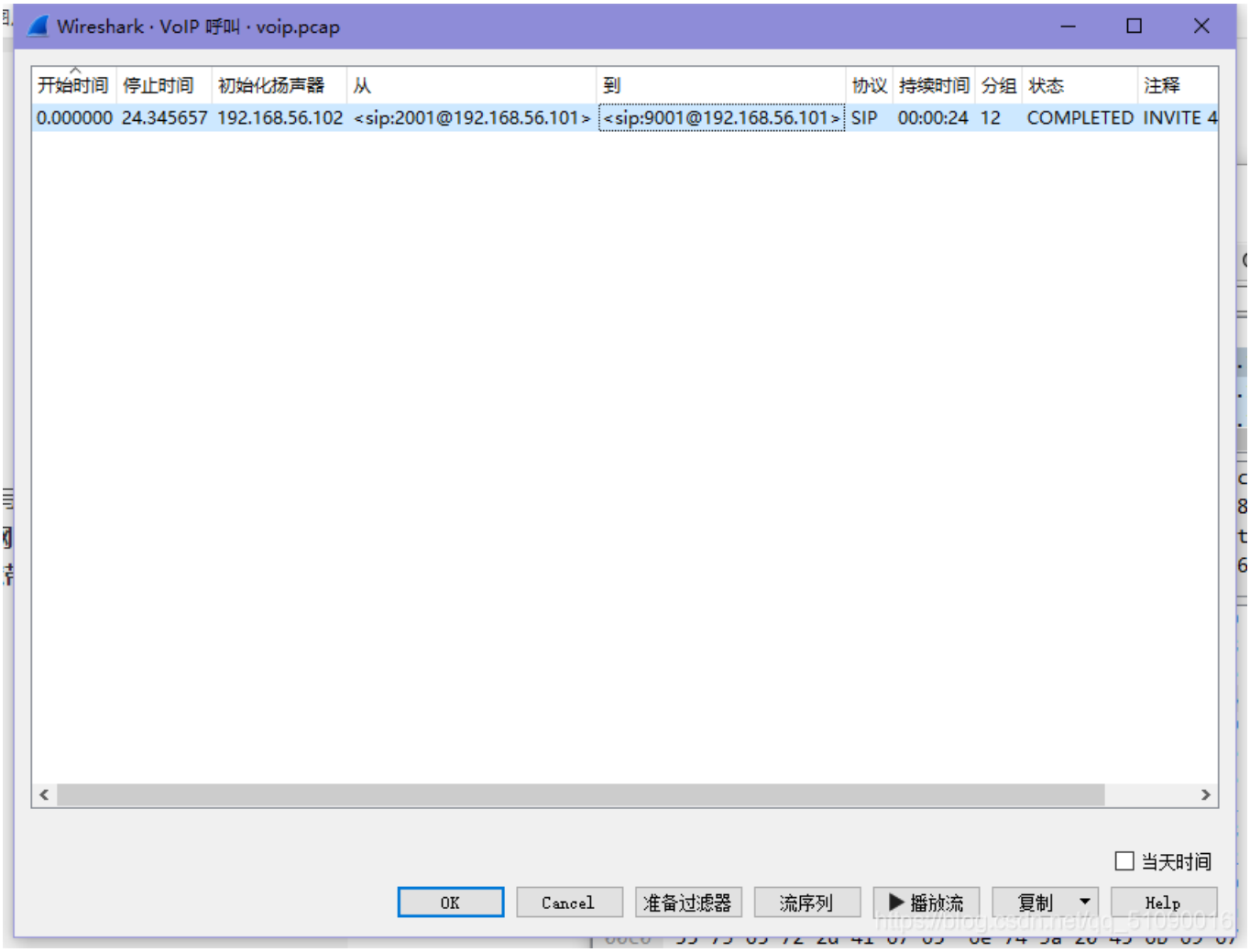
这题暂时先放着把。。

voip（Wireshark抓取RTP包）

啥是voip？

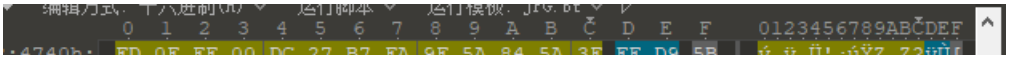
VoIP——基于IP的语音传输（英语：Voice over Internet Protocol，缩写为VoIP）是一种语音通话技术，经由网际协议（IP）来达成语音通话与多媒体会议，也就是经由互联网来进行通信。其他非正式的名称有IP电话（IP telephony）、互联网电话（Internet telephony）、宽带电话（broadband telephony）以及宽带电话服务（broadband phone service）

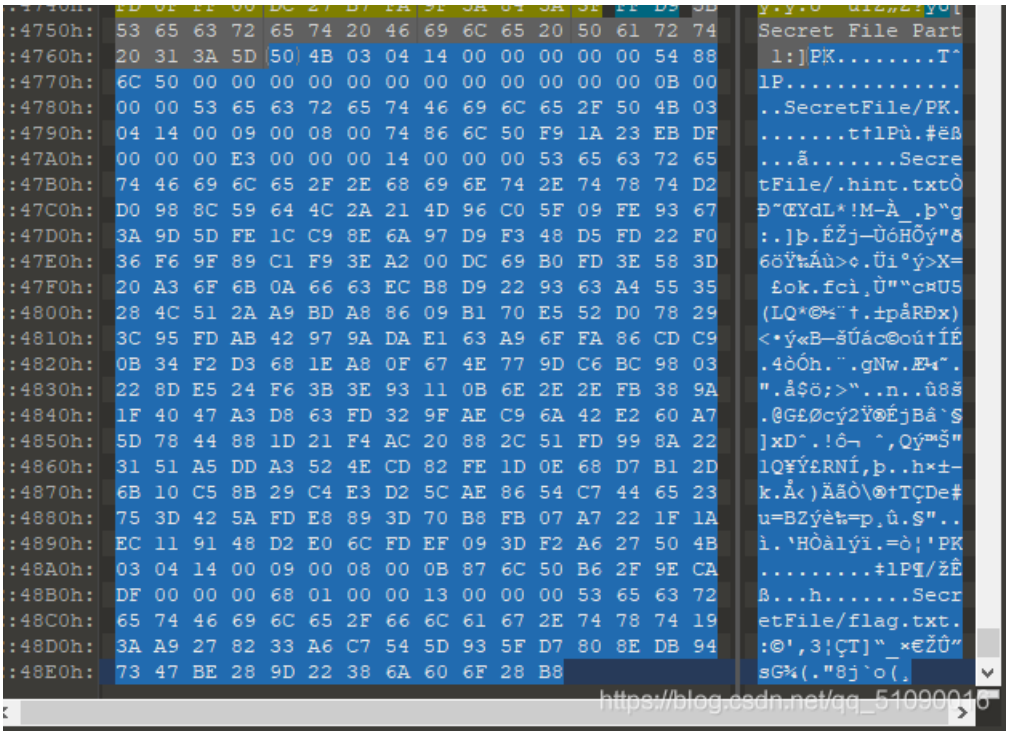
直接主菜单点击 电话——>VoIP电话 就可以听到声音



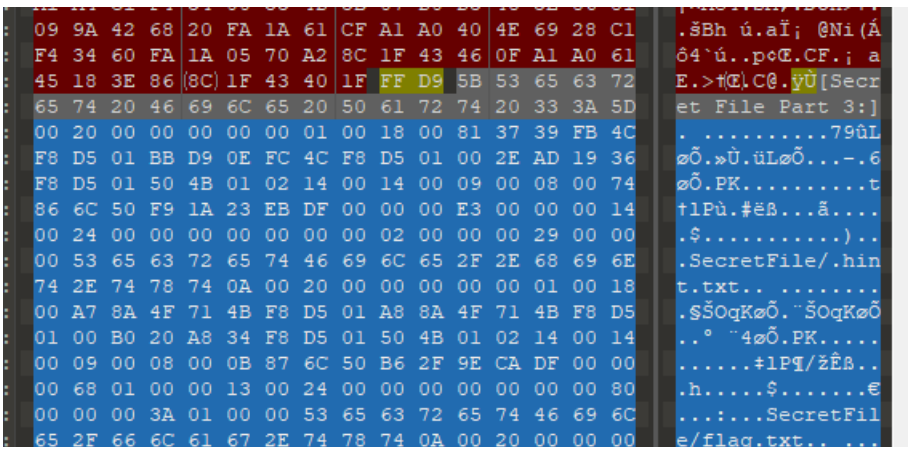
[MRCTF2020]pyFlag(拼接出来一个文件)

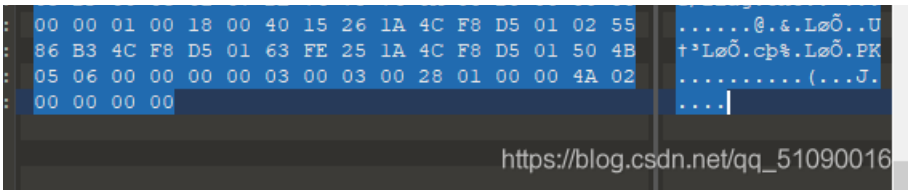
正常从图片入手，搜pk的时候在文件末尾发现了一个secret file，每个图片都有，提取出来组成一个zip



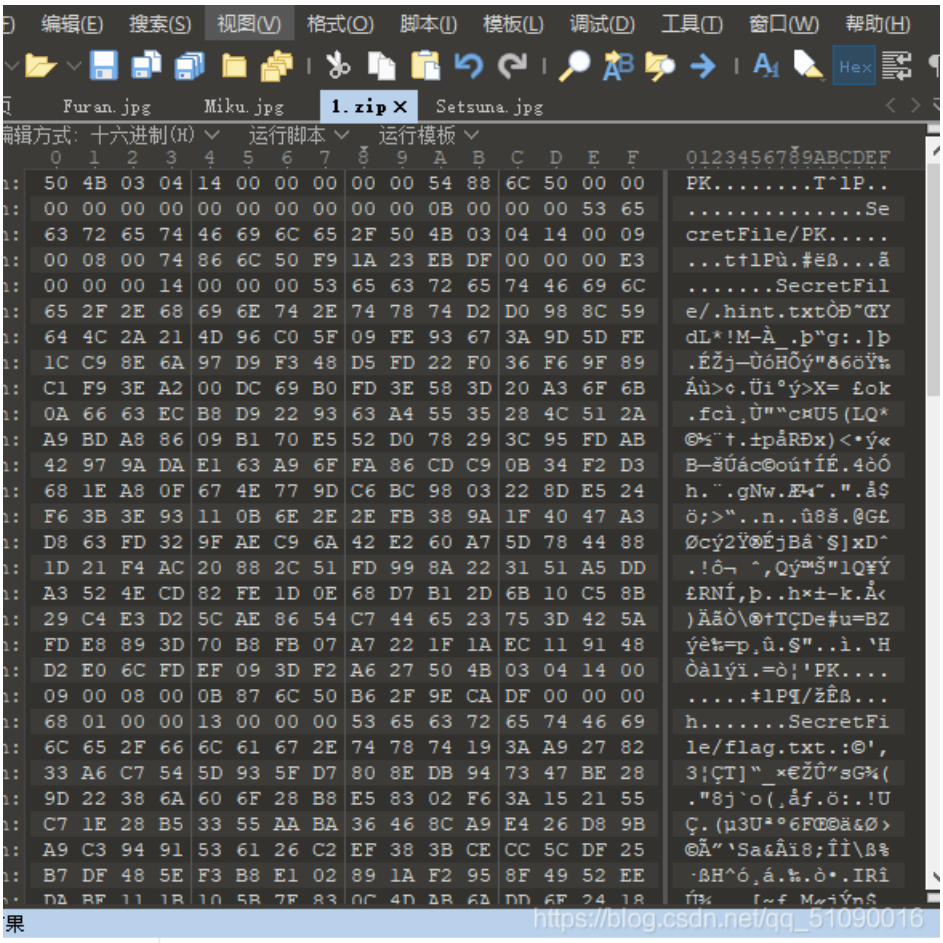

:4750h: 53 65 63 72 65 74 20 46 69 6C 65 20 50 61 72 74 Secret File Part
:4760h: 20 31 3A 5D 50 4B 03 04 14 00 00 00 00 00 54 88 1:]PK.....T^
:4770h: 6C 50 00 00 00 00 00 00 00 00 00 00 00 00 0B 00 lP..............
:4780h: 00 00 53 65 63 72 65 74 46 69 6C 65 2F 50 4B 03 ..SecretFile/PK.
:4790h: 04 14 00 09 00 08 00 74 86 6C 50 F9 1A 23 EB DF ....t†lPù.#ëß
:47A0h: 00 00 00 E3 00 00 00 14 00 00 00 53 65 63 72 65 ...ã.......Secre
:47B0h: 74 46 69 6C 65 2F 2E 68 69 6E 74 2E 74 78 74 D2 tFile/.hint.txtÒ
:47C0h: D0 98 8C 59 64 4C 2A 21 4D 96 C0 5F 09 FE 93 67 Ð˜ŒYdL*!M-À_.þ"g
:47D0h: 3A 9D 5D FE 1C C9 8E 6A 97 D9 F3 48 D5 FD 22 F0 :.]þ.ÉŽj—ÙóHÕý"ð
:47E0h: 36 F6 9F 89 C1 F9 3E A2 00 DC 69 B0 FD 3E 58 3D 6öŸ‰Áù>¢.Üi°ý>X=
:47F0h: 20 A3 6F 6B 0A 66 63 EC B8 D9 22 93 63 A4 55 35 £ok.fcì¸Ù"“c¤U5
:4800h: 28 4C 51 2A A9 BD A8 86 09 B1 70 E5 52 D0 78 29 (LQ*©½¨†.±påRÐx)
:4810h: 3C 95 FD AB 42 97 9A DA E1 63 A9 6F FA 86 CD C9 <•ý«B—šÚác©oú†ÍÉ
:4820h: 0B 34 F2 D3 68 1E A8 0F 67 4E 77 9D C6 BC 98 03 .4òÓh.¨.gNw.Æ¼˜.
:4830h: 22 8D E5 24 F6 3B 3E 93 11 0B 6E 2E 2E FB 38 9A ".å$ö;>“..n..û8š
:4840h: 1F 40 47 A3 D8 63 FD 32 9F AE C9 6A 42 E2 60 A7 .@G£Øcý2Ÿ®ÉjBâ`§
:4850h: 5D 78 44 88 1D 21 F4 AC 20 88 2C 51 FD 99 8A 22]xDˆ.!ô¬ ˆ,Q㙊"
:4860h: 31 51 A5 DD A3 52 4E CD 82 FE 1D 0E 68 D7 B1 2D 1Q¥Ý£RNÍ‚þ..h×±-
:4870h: 6B 10 C5 8B 29 C4 E3 D2 5C AE 86 54 C7 44 65 23 k.Å‹)ÄãÒ\®†TÇDe#
:4880h: 75 3D 42 5A FD E8 89 3D 70 B8 FB 07 A7 22 1F 1A u=BZýè‰=p¸û.§"..
:4890h: EC 11 91 48 D2 E0 6C FD EF 09 3D F2 A6 27 50 4B ì.‘HÒàlýï.=ò¦'PK
:48A0h: 03 04 14 00 09 00 08 00 0B 87 6C 50 B6 2F 9E CA ....‡lP¶/žÊ
:48B0h: DF 00 00 00 68 01 00 00 13 00 00 00 53 65 63 72 ß...h.......Secr
:48C0h: 65 74 46 69 6C 65 2F 66 6C 61 67 2E 74 78 74 19 etFile/flag.txt.
:48D0h: 3A A9 27 82 33 A6 C7 54 5D 93 5F D7 80 8E DB 94 :©'‚3¦ÇT]“_×€ŽÛ”
:48E0h: 73 47 BE 28 9D 22 38 6A 60 6F 28 B8 sG¾(."8j`o(¸


起始页
Furan.jpg
Miku.jpg
1.zip
Setsuna.jpg
编辑方式: 十六进制(H)
运行脚本
运行模板: JPG.bt
0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
:6B60h: 60 6A F8 5F FE 46 3D 24 76 3A 95 B7 FE 8C 14 8A `jø_þF=$v:•·þŒ.Š
:6B70h: 87 C4 BD 4D 5F 19 92 FE 30 D7 CB 73 FF 00 13 19 ‡Ä½M_.’þ0×Ësÿ...
:6B80h: FF 00 F4 2A 45 3D 5B 38 E9 00 0E 40 F5 A6 64 F7 ÿ.ô*E=[8é..@õ¦d÷
:6B90h: 24 80 07 95 55 B9 14 3D 87 1D CB 00 05 24 0F 53 $€.•U¹.=‡.Ë..$.S
:6BA0h: 40 CB 56 9D 5A 93 2A 24 7A 9B B0 48 90 1F 95 DB @ËV.Z“*$z›°H..•Û
:6BB0h: E6 F7 A2 22 A8 F4 44 40 00 30 29 88 FF D9 5B 53 æ÷¢"¨ôD@.0)ˆÿÙ[S
:6BC0h: 65 63 72 65 74 20 46 69 6C 65 20 50 61 72 74 20 ecret File Part
:6BD0h: 32 3A 5D E5 83 02 F6 3A 15 21 55 C7 1E 28 B5 33 2:]åƒ.ö:.!UÇ.(µ3
:6BE0h: 55 AA BA 36 46 8C A9 E4 26 D8 9B A9 C3 94 91 53 Uª°6FŒ©ä&Ø›©Ã”‘S
:6BF0h: 61 26 C2 EF 38 3B CE CC 5C DF 25 B7 DF 48 5E F3 a&Âï8;ÎÌ\ß%·ßH^ó
:6C00h: B8 E1 02 89 1A F2 95 8F 49 52 EE DA BE 11 1B 10 ¸á.‰.ò•.IRîÚ¾...
:6C10h: 5B 7E 83 0C 4D AB 6A DD 6E 24 18 D5 CE C1 E4 1B [~ƒ.M«jÝn$.ÕÎÁä.
:6C20h: 6B 97 36 66 5E 6A 13 41 37 FC CA E0 F6 A5 A7 4A k—6f^j.A7üÊàö¥§J
:6C30h: C5 89 AD 12 B2 0A AF BD CB AC 23 04 8C E9 44 3E Å‰-.².¯½Ë¬#.ŒéD>
:6C40h: C8 FE 46 36 2F 8C 36 8C 81 F9 A6 61 9C B2 0E C1 ÈþF6/Œ6Œ.ù¦aœ².Á
:6C50h: 9E B5 93 A6 10 F9 00 DB 24 E5 F3 30 BC 19 EB 67 žµ“¦.ù.Û$åó0¼.ëg
:6C60h: 1C C6 46 83 01 24 25 17 CC E3 84 C7 69 0D 09 04 .ÆFƒ.$%.Ìã„Çi...
:6C70h: 73 E3 A9 AC 84 57 1B B2 61 32 B0 AE 03 26 C4 A3 sã©¬„W.²a2°®.&Ä£
:6C80h: 6F A3 4C 3F 48 87 FB F0 D1 EC E3 10 F7 35 D8 27 o£L?H‡ûðÑìã.÷5Ø'
:6C90h: 22 DD 56 99 90 50 4B 01 02 14 00 14 00 00 00 00 "ÝV™.PK.........
:6CA0h: 00 54 88 6C 50 00 00 00 00 00 00 00 00 00 00 00 .Tˆ lP..........
:6CB0h: 00 0B 00 24 00 00 00 00 00 00 00 10 00 00 00 00 ...$............
:6CC0h: 00 00 00 53 65 63 72 65 74 46 69 6C 65 2F 0A ...SecretFile/.

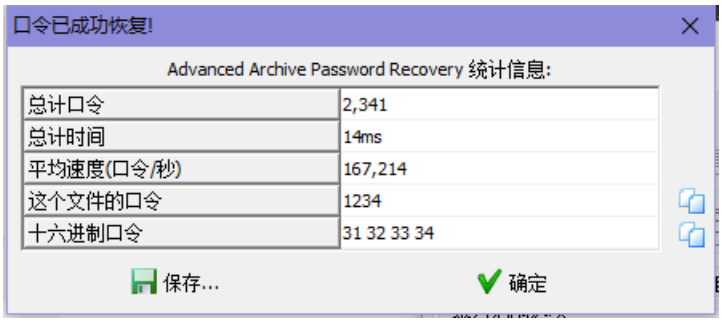

09 9A 42 68 20 FA 1A 61 CF A1 A0 40 4E 69 28 C1 .šBh ú.aÏ¡ @Ni(Á
F4 34 60 FA 1A 05 70 A2 8C 1F 43 46 0F A1 A0 61 ô4`ú..p¢Œ.CF.¡ a
45 18 3E 86 8C 1F 43 40 1F FF D9 5B 53 65 63 72 E.>†Œ.C@.ÿÙ[Secr
65 74 20 46 69 6C 65 20 50 61 72 74 20 33 3A 5D et File Part 3:]
00 20 00 00 00 00 00 00 01 00 18 00 81 37 39 FB 4C 79ûL
F8 D5 01 BB D9 0E FC 4C F8 D5 01 00 2E AD 19 36 øÕ.»Ù.üLøÕ...-.6
F8 D5 01 50 4B 01 02 14 00 14 00 09 00 08 00 74 øÕ.PK.........t
86 6C 50 F9 1A 23 EB DF 00 00 00 E3 00 00 00 14 †lPù.#ëß...ã....
00 24 00 00 00 00 00 00 00 02 00 00 00 29 00 00 .$...........)..
00 53 65 63 72 65 74 46 69 6C 65 2F 2E 68 69 6E .SecretFile/.hin
74 2E 74 78 74 0A 00 20 00 00 00 00 00 01 00 18 t.txt.. ....
00 A7 8A 4F 71 4B F8 D5 01 A8 8A 4F 71 4B F8 D5 .§ŠOqKøÕ.¨ŠOqKøÕ
01 00 B0 20 A8 34 F8 D5 01 50 4B 01 02 14 00 14 ..° ¨4øÕ.PK.....
00 09 00 08 00 0B 87 6C 50 B6 2F 9E CA DF 00 00 ......‡lP¶/žÊß..
00 68 01 00 00 13 00 24 00 00 00 00 00 00 00 80 .h.....$.......€
00 00 00 3A 01 00 00 53 65 63 72 65 74 46 69 6C ...:...SecretFil
65 2F 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 e/flag.txt.. ...



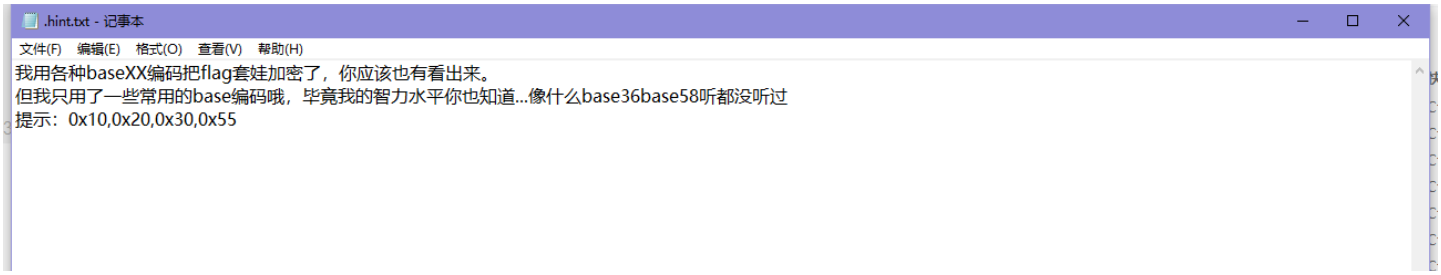
复制出来得到：

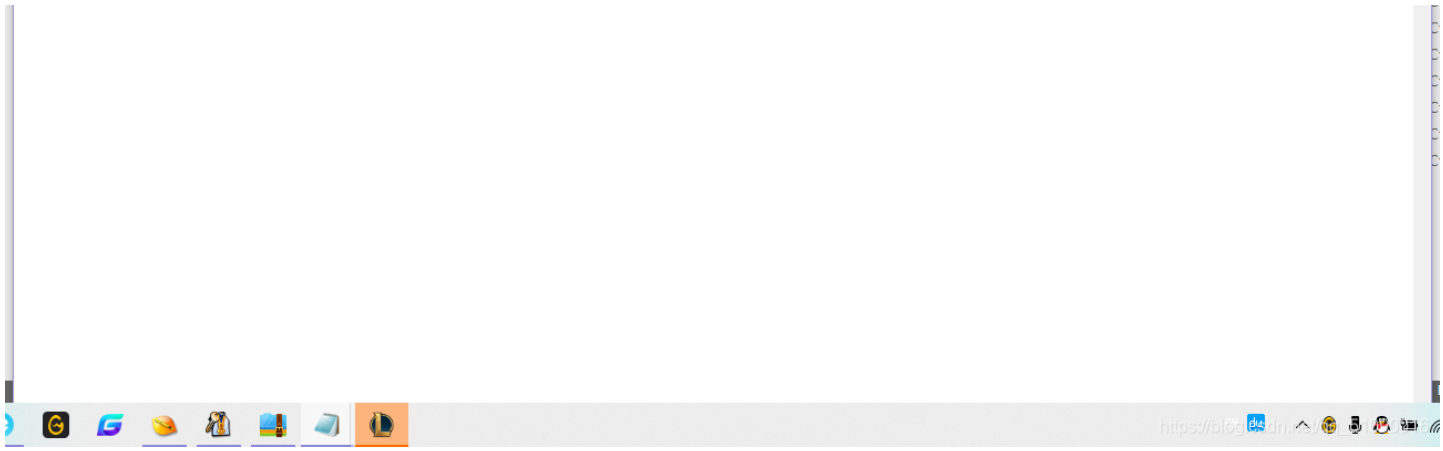


解压需要密码，爆破

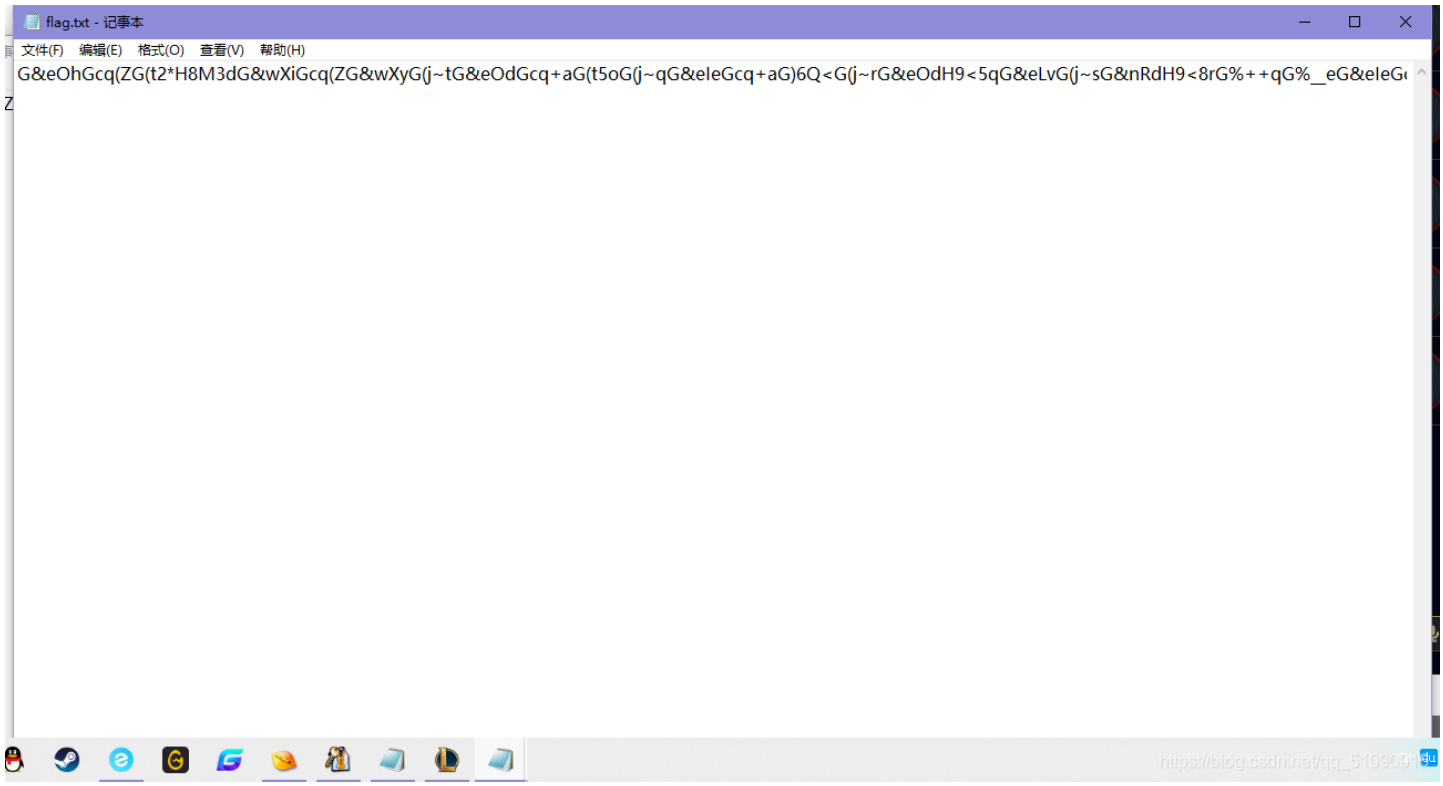


得到两个txt文件，看hint





再看flag.txt



根据提示，会有的base解码：16，32，48，85（16进制转10进制）

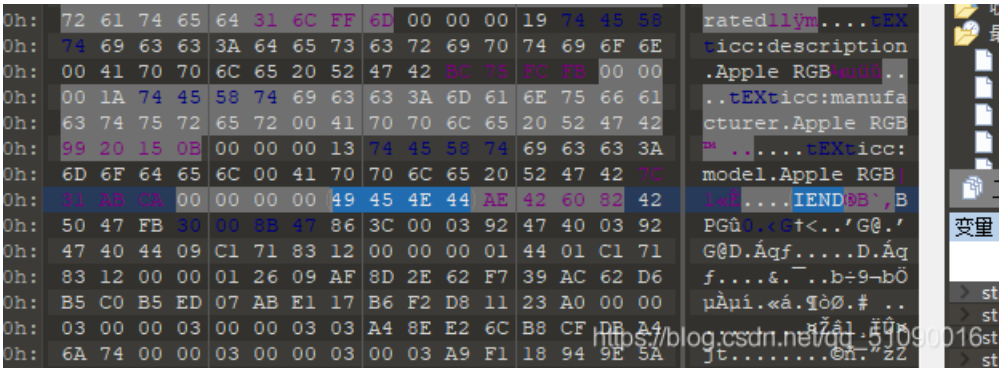
用脚本：

```python
#!/usr/bin/env python

import base64
import re

def baseDec(text,type):
    if type == 1:
        return base64.b16decode(text)
    elif type == 2:
        return base64.b32decode(text)
    elif type == 3:
        return base64.b64decode(text)
    elif type == 4:
        return base64.b85decode(text)
    else:
        pass

def detect(text):
    try:
        if re.match("^[0-9A-F=]+$",text.decode()) is not None:
            return 1
    except:
        pass

    try:
        if re.match("^[A-Z2-7=]+$",text.decode()) is not None:
            return 2
    except:
        pass

    try:
        if re.match("^[A-Za-z0-9+/=]+$",text.decode()) is not None:
            return 3
    except:
        pass

    return 4

def autoDec(text):
    while True:
        if b"MRCTF{" in text:
            print("\n"+text.decode())
            break

        code = detect(text)
        text = baseDec(text,code)

with open("flag.txt",'rb') as f:
    flag = f.read()

autoDec(flag)
```

就行了

Business Planning Group（010搜索iend bpg图像格式）

010打开后搜索iend（PNG图像的图像结束数据（IEND））



发现bpg

BPG (新的图像格式)

编辑 讨论 上传视频

本词条缺少概述图，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

BPG（Better Portable Graphics，更好的可移植图形）是一种新的图像格式。它的目的是在质量或文件大小有问题时替换jpeg图像格式。

中文名	更好的可移植图形	缩　写	BPG
外文名	Better Portable Graphics	同类项	JPEG等图片格式

BPG（更好的可移植图形）是一种新的图像格式。它的目的是在质量或文件大小有问题时替换jpeg图像格式。其主要优点是：

高压缩比。对于类似的质量，文件比jpeg小得多。

大多数Web浏览器都支持小型的javascript解码器（gzipped大小：56kb）。

基于HEVC开放视频压缩标准的子集。

支持与jpeg相同的色度格式（灰度，ycbcr 4:2:0，4:2:2，4:4:4），以减少转换过程中的损失。支持alpha通道。还支持RGB、YCGCO和CMYK颜色空间。

本机支持每个通道8到14位，以获得更高的动态范围。

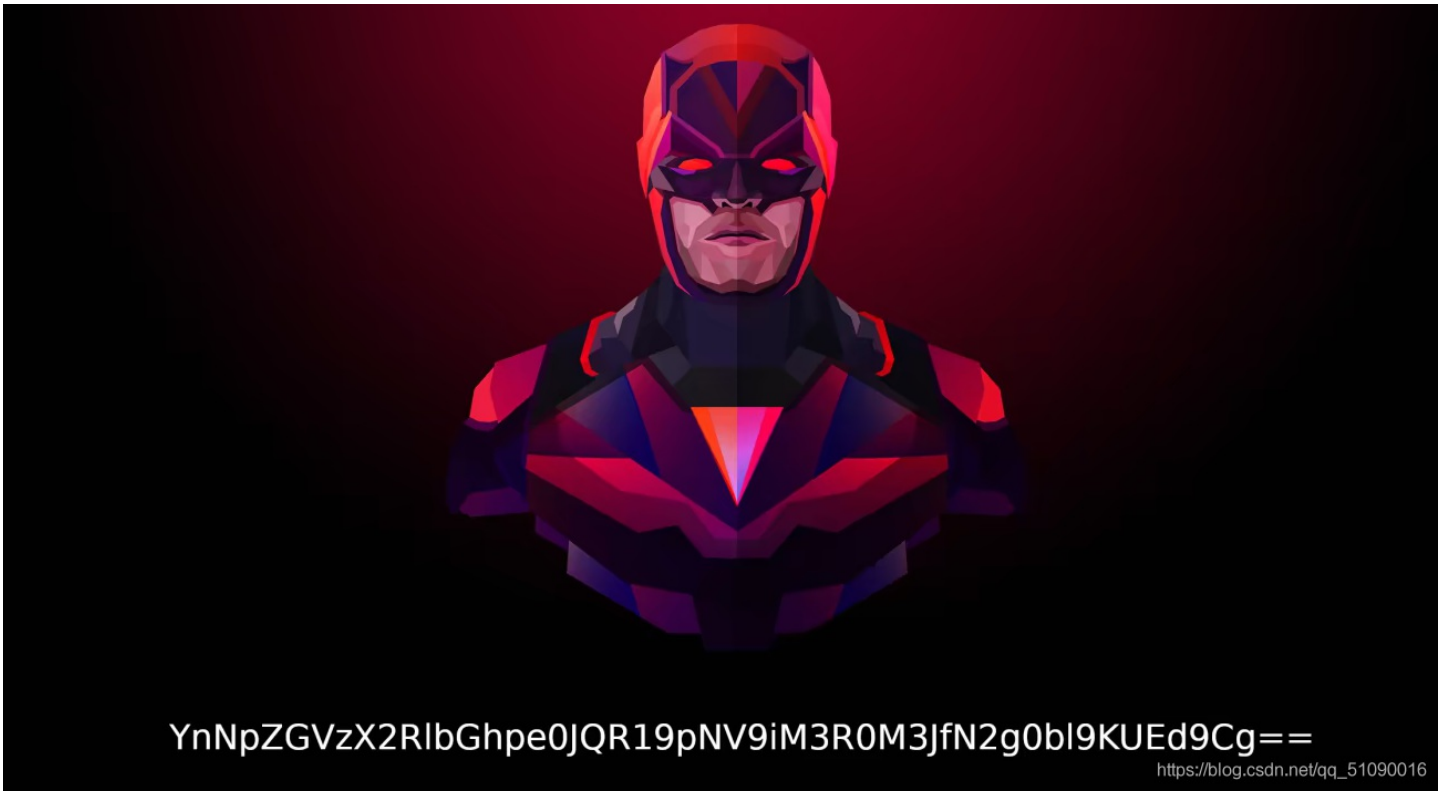
支持无损压缩。

可以包含各种元数据（如exif、icc profile、xmp）。

动画支持。 [1]

将这些数据另存出来为bpg文件，发现Windows的图像软件不能直接打开，网上能打开的工具

https://bellard.org/bpg/



base64解密即可

[GWCTF2019]huyao（频域盲水印隐写）

看到两张一样的图片，想到盲水印，但这里用BlindWaterMark还不行，看wp知道是频域盲水印隐写，用脚本

```python
# coding=utf-8
import cv2
import numpy as np
import random
import os
from argparse import ArgumentParser
ALPHA = 5


def build_parser():
    parser = ArgumentParser()
    parser.add_argument('--original', dest='ori', required=True)
    parser.add_argument('--image', dest='img', required=True)
    parser.add_argument('--result', dest='res', required=True)
    parser.add_argument('--alpha', dest='alpha', default=ALPHA)
    return parser


def main():
    parser = build_parser()
    options = parser.parse_args()
    ori = options.ori
    img = options.img
    res = options.res
    alpha = options.alpha
    if not os.path.isfile(ori):
        parser.error("original image %s does not exist." % ori)
    if not os.path.isfile(img):
        parser.error("image %s does not exist." % img)
    decode(ori, img, res, alpha)


def decode(ori_path, img_path, res_path, alpha):
    ori = cv2.imread(ori_path)
    img = cv2.imread(img_path)
    ori_f = np.fft.fft2(ori)
    img_f = np.fft.fft2(img)
    height, width = ori.shape[0], ori.shape[1]
    watermark = (ori_f - img_f) / alpha
    watermark = np.real(watermark)
    res = np.zeros(watermark.shape)
    random.seed(height + width)
    x = range(height / 2)
    y = range(width)
    random.shuffle(x)
    random.shuffle(y)
    for i in range(height / 2):
        for j in range(width):
            res[x[i]][y[j]] = watermark[i][j]
    cv2.imwrite(res_path, res, [int(cv2.IMWRITE_JPEG_QUALITY), 100])


if __name__ == '__main__':
    main()
```

得到

[UTCTF2020]File Carving

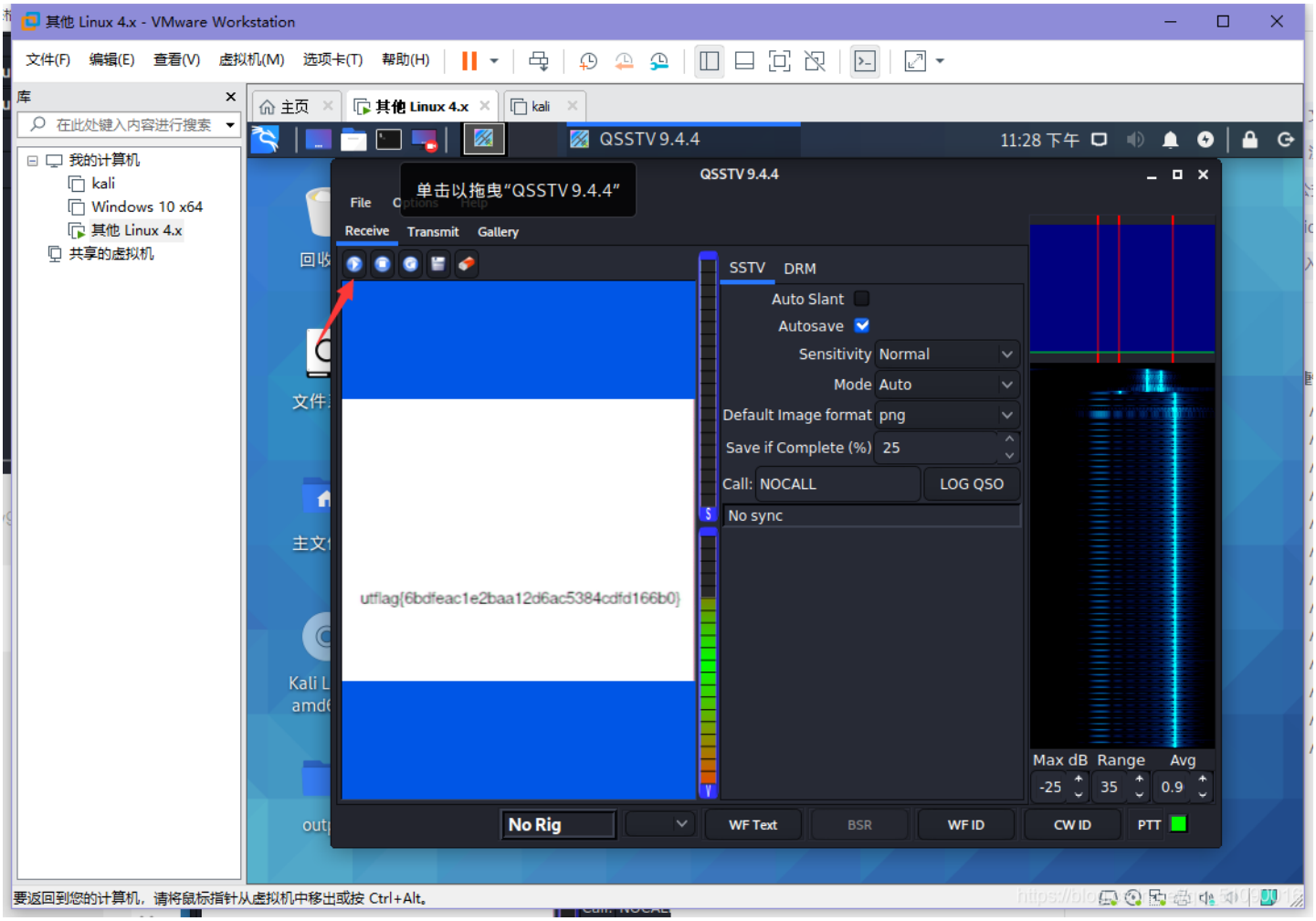
[GUET-CTF2019]soul sipse（音频隐写）

[UTCTF2020]spectogram（奇怪的操作）

[UTCTF2020]sstv(qsstv)

kali打开qsstv
Options->Configuration->Sound勾选From file

点这个



我爱Linux(Python Picke序列化内容)