BMZCTF-MISC（一） WriteUp

原创

Crazy198410 于 2020-11-15 23:21:56 发布 441 收藏

分类专栏： BMZCTF BMZCTF-MISC 文章标签： 安全 python

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/Crazy198410/article/details/109612549

版权

BMZCTF 同时被 2 个专栏收录

2 篇文章 0 订阅

订阅专栏

BMZCTF-MISC

4 篇文章 0 订阅

订阅专栏

文章目录

MISC

http://www.bmzclub.cn/challenges
解密

0x01 签到题

关注公众号：白帽子社区，回复关键字：BMZCTF 获取flag

签到题

1

关注公众号：白帽子社区，回复关键字：BMZCTF 获取flag

https://blog.csdn.net/Crazy198410

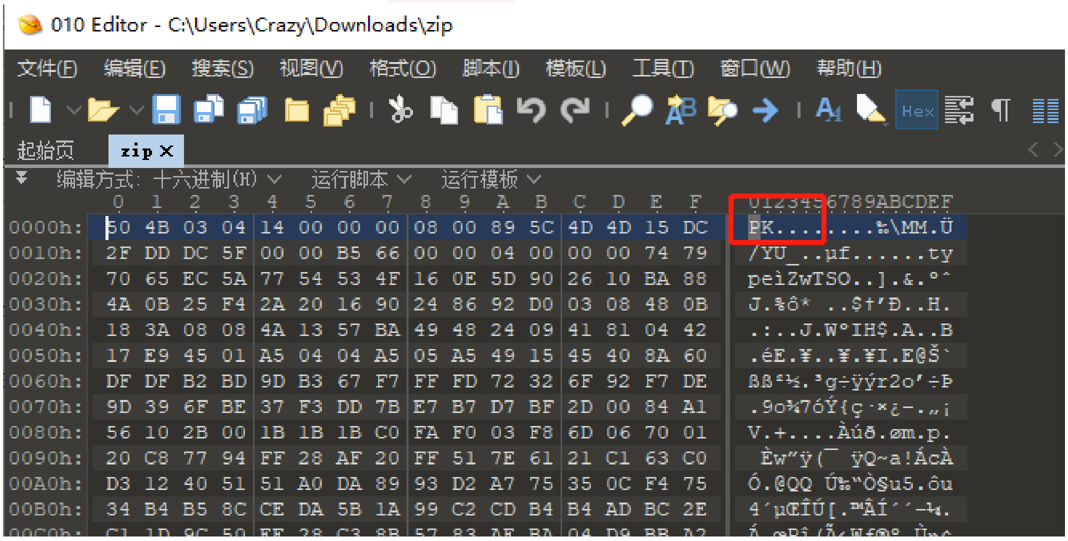
0x02 2018 HEBTUCTF 签到题

2018 HEBTUCTF 签到题

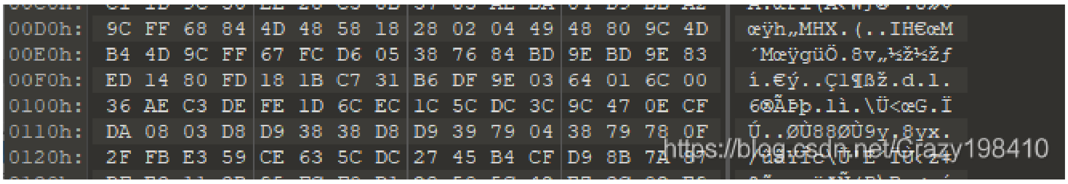
91

https://blog.csdn.net/Crazy198410

下载压缩包。是个名为zip的文件，没有后缀名。用`010editor`打开，看下文件头部：


010 Editor - C:\Users\Crazy\Downloads\zip
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 工具(T) 窗口(W) 帮助(H)
Hex
起始页
zip ×
编辑方式：十六进制(H)
运行脚本
运行模板

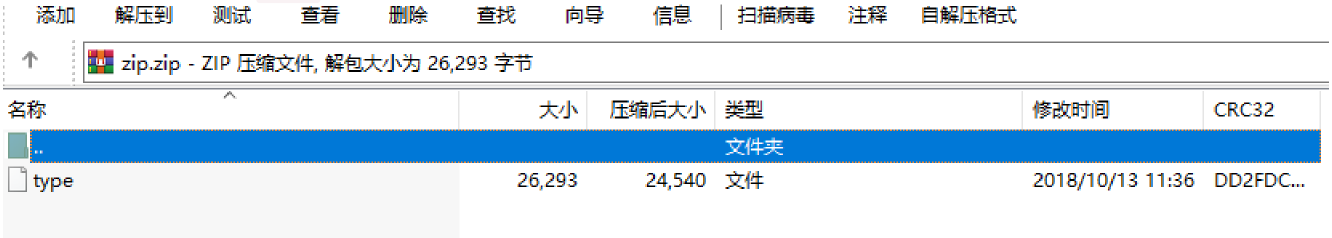
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 50 4B 03 04 14 00 00 00 08 00 89 5C 4D 4D 15 DC   PK........‰\MM.Ü
0010h: 2F DD DC 5F 00 00 B5 66 00 00 04 00 00 00 74 79   /ÝÜ_..µf......ty
0020h: 70 65 EC 5A 77 54 53 4F 16 0E 5D 90 26 10 BA 88   peìZwTSO..].&.º^
0030h: 4A 0B 25 F4 2A 20 16 90 24 86 92 D0 03 08 48 0B   J.%ô* ..$†'Ð..H.
0040h: 18 3A 08 08 4A 13 57 BA 49 48 24 09 41 81 04 42   .:..J.WºIH$.A..B
0050h: 17 E9 45 01 A5 04 04 A5 05 A5 49 15 45 40 8A 60   .éE.¥..¥.¥I.E@Š`
0060h: DF DF B2 BD 9D B3 67 F7 FF FD 72 32 6F 92 F7 DE   ßß²½.³g÷ÿýr2o'÷Þ
0070h: 9D 39 6F BE 37 F3 DD 7B E7 B7 D7 BF 2D 00 84 A1   .9o¾7óÝ{ç·×¿-.„¡
0080h: 56 10 2B 00 1B 1B 1B C0 FA F0 03 F8 6D 06 70 01   V.+....Àúð.øm.p.
0090h: 20 C8 77 94 FF 28 AF 20 FF 51 7E 61 21 C1 63 C0    Èw"ÿ(¯ ÿQ~a!ÁcÀ
00A0h: D3 12 40 51 51 A0 DA 89 93 D2 A7 75 35 0C F4 75   Ó.@QQ Ú‰"Ò§u5.ôu
00B0h: 34 B4 B5 8C CE DA 5B 1A 99 C2 CD B4 B4 AD BC 2E   4´µŒÎÚ[.™ÂÍ´´¼.
```



```
00D0h:  9C FF 68 84 4D 48 58 18 28 02 04 49 48 80 9C 4D   œÿh„MHX.(..IH€œM
00E0h:  B4 4D 9C FF 67 FC D6 05 38 76 84 BD 9E BD 9E 83   ´MœÿgüÖ.8v„½ž½žƒ
00F0h:  ED 14 80 FD 18 1B C7 31 B6 DF 9E 03 64 01 6C 00   í.€ý..Ç1¶ßž.d.l.
0100h:  36 AE C3 DE FE 1D 6C EC 1C 5C DC 3C 9C 47 0E CF   6®ÃÞþ.lì.\Ü<œG.Ï
0110h:  DA 08 03 D8 D9 38 38 D8 D9 39 79 04 38 79 78 0F   Ú..ØÙ88ØÙ9y.8yx.
0120h:  2F FB E3 59 CE 63 5C DC 27 45 B4 CF D9 8B 7A
```

https://blog.csdn.net/Crazy198410

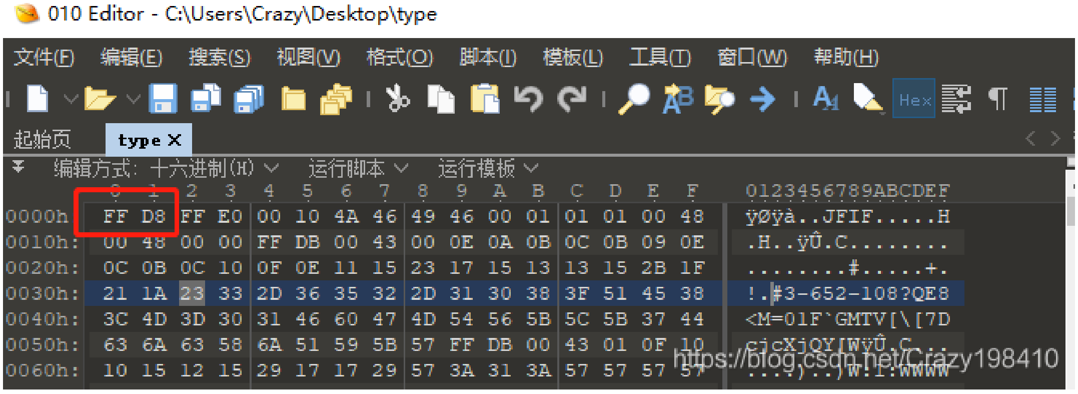
发现是个压缩包，添加后缀`.zip`，然后用winrar打开。



zip.zip - ZIP 压缩文件, 解包大小为 26,293 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
type	26,293	24,540	文件	2018/10/13 11:36	DD2FDC...

发现又是一个无后缀文件，解压出来，再用`010editor`打开

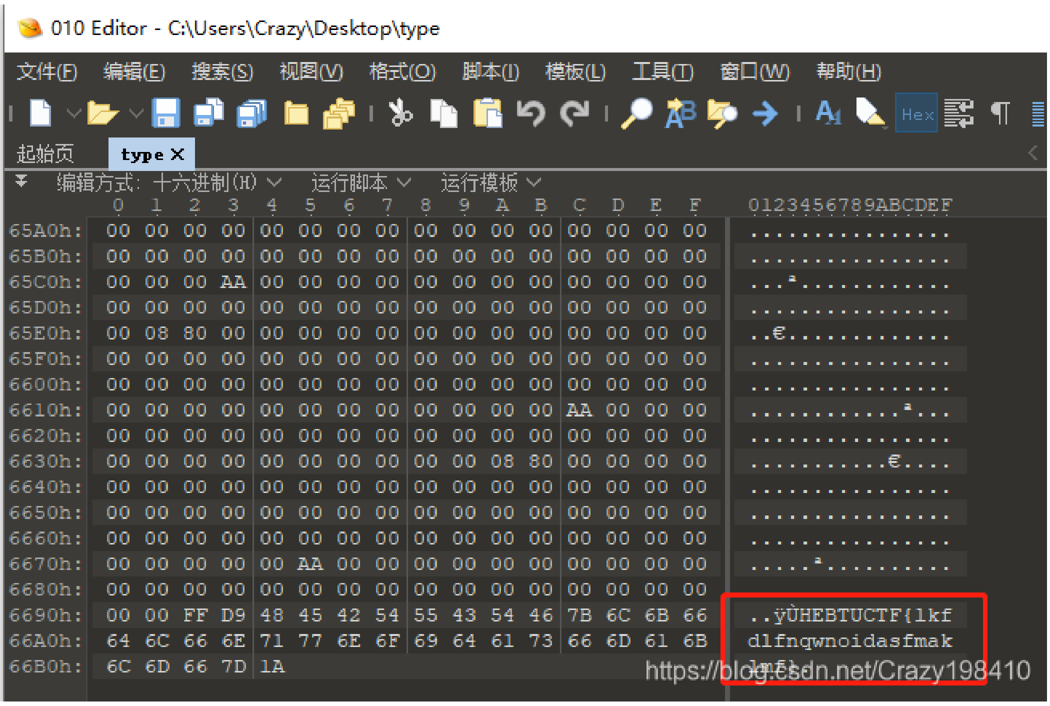


010 Editor - C:\Users\Crazy\Desktop\type

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0123456789ABCDEF
0000h:  FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 48   ÿØÿà..JFIF.....H
0010h:  00 48 00 00 FF DB 00 43 00 0E 0A 0B 0C 0B 09 0E   .H..ÿÛ.C........
0020h:  0C 0B 0C 10 0F 0E 11 15 23 17 15 13 13 15 2B 1F   ........#.....+.
0030h:  21 1A 23 33 2D 36 35 32 2D 31 30 38 3F 51 45 38   !.#3-652-108?QE8
0040h:  3C 4D 3D 30 31 46 60 47 4D 54 56 5B 5C 5B 37 44   <M=01F`GMTV[\[7D
0050h:  63 6A 63 58 6A 51 59 5B 57 FF DB 00 43 01 0F 10   cjcXjQY[WÿÛ.C...
0060h:  10 15 12 15 29 17 17 29 57 3A 31 3A 57 57 57 57   ....)..)W:1:WWWW
```

https://blog.csdn.net/Crazy198410

疑似是个jpg文件，再看下文件尾部：



010 Editor - C:\Users\Crazy\Desktop\type

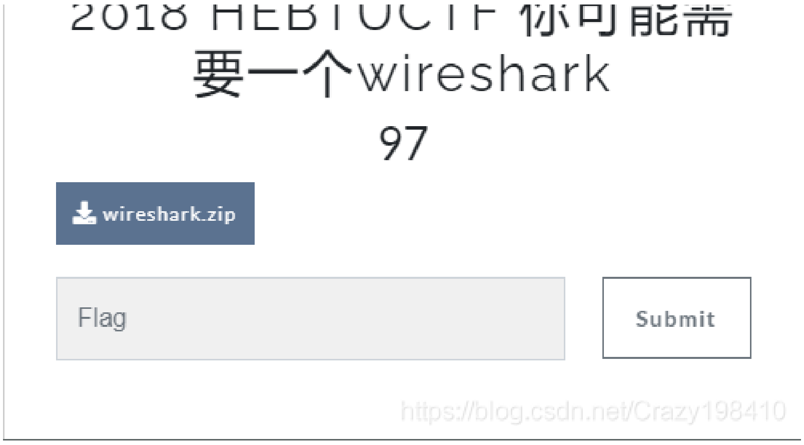
```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F    0123456789ABCDEF
65A0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
65B0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
65C0h:  00 00 00 AA 00 00 00 00 00 00 00 00 00 00 00 00   ...ª............
65D0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
65E0h:  00 08 80 00 00 00 00 00 00 00 00 00 00 00 00 00   ..€.............
65F0h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6600h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6610h:  00 00 00 00 00 00 00 00 00 00 00 00 AA 00 00 00   ............ª...
6620h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6630h:  00 00 00 00 00 00 00 00 00 00 08 80 00 00 00 00   ...........€....
6640h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6650h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6660h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
6670h:  00 00 00 00 00 AA 00 00 00 00 00 00 00 00 00 00   .....ª..........
6680h:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6690h:  00 00 FF D9 48 45 42 54 55 43 54 46 7B 6C 6B 66   ..ÿÙHEBTUCTF{lkf
66A0h:  64 6C 66 6E 71 77 6E 6F 69 64 61 73 66 6D 61 6B   dlfnqwnoidasfmak
66B0h:  6C 6D 66 7D 1A                                    lmf}.
```

https://blog.csdn.net/Crazy198410

可以得到flag

0x03 2018 HEBTUCTF 你可能需要一个wireshark

2018 HEBTUCTF 你可能需

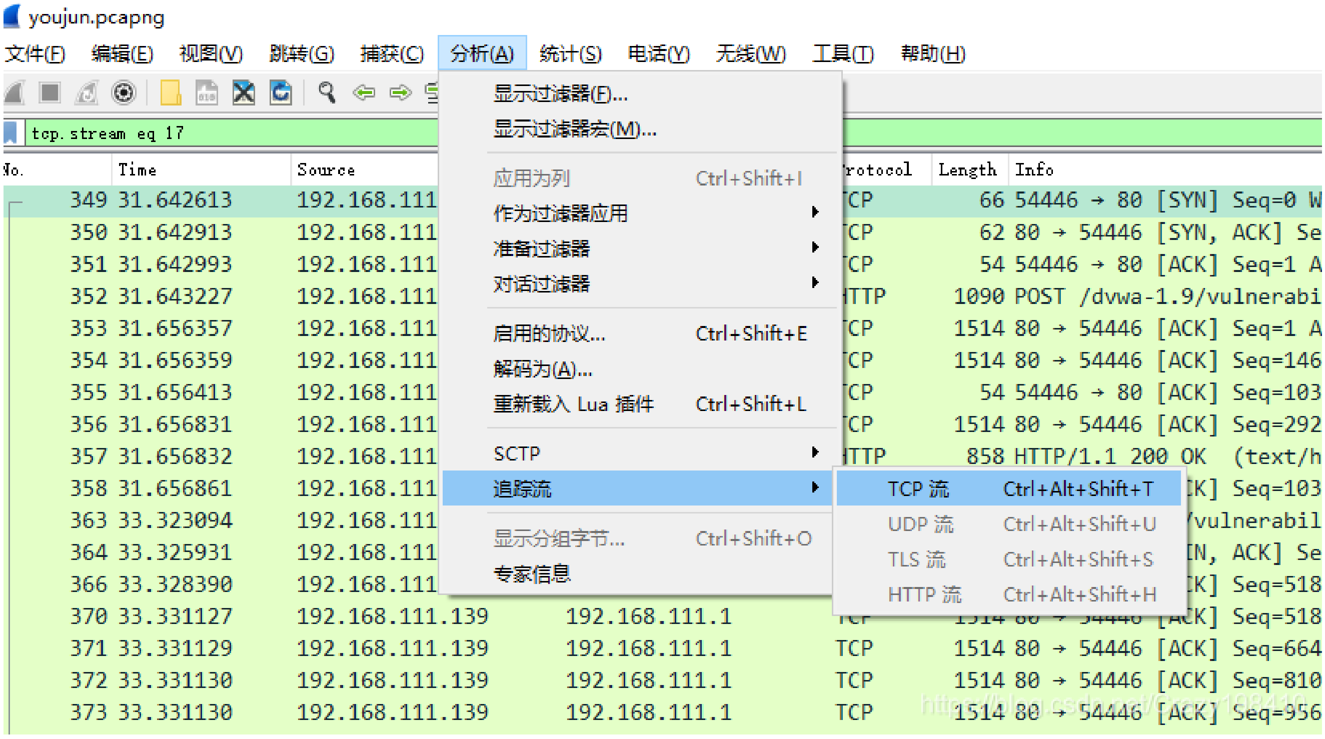


2018 HEBTUCTF 你可能需要一个wireshark

97

https://blog.csdn.net/Crazy198410

下载附件，是个流量包文件，用wireshark打开：

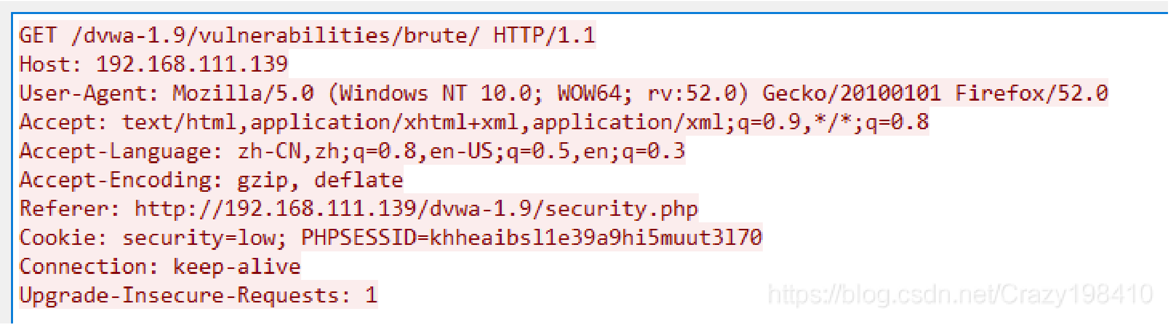


youjun.pcapng

```
tcp.stream eq 17

No.  Time       Source                                   rotocol  Length Info
 349 31.642613  192.168.111                              CP           66 54446 → 80 [SYN] Seq=0 W
 350 31.642913  192.168.111                              CP           62 80 → 54446 [SYN, ACK] Se
 351 31.642993  192.168.111                              CP           54 54446 → 80 [ACK] Seq=1 A
 352 31.643227  192.168.111                              TTP        1090 POST /dvwa-1.9/vulnerabi
 353 31.656357  192.168.111                              CP         1514 80 → 54446 [ACK] Seq=1 A
 354 31.656359  192.168.111                              CP         1514 80 → 54446 [ACK] Seq=146
 355 31.656413  192.168.111                              CP           54 54446 → 80 [ACK] Seq=103
 356 31.656831  192.168.111                              CP         1514 80 → 54446 [ACK] Seq=292
 357 31.656832  192.168.111                              TTP         858 HTTP/1.1 200 OK  (text/h
 358 31.656861  192.168.111                                              K] Seq=103
 363 33.323094  192.168.111                                              /vulnerabil
 364 33.325931  192.168.111                                              N, ACK] Se
 366 33.328390  192.168.111                                              K] Seq=518
 370 33.331127  192.168.111.139  192.168.111.1                           K] Seq=518
 371 33.331129  192.168.111.139  192.168.111.1           TCP        1514 80 → 54446 [ACK] Seq=664
 372 33.331130  192.168.111.139  192.168.111.1           TCP        1514 80 → 54446 [ACK] Seq=810
 373 33.331130  192.168.111.139  192.168.111.1           TCP        1514 80 → 54446 [ACK] Seq=956
```

https://blog.csdn.net/Crazy198410

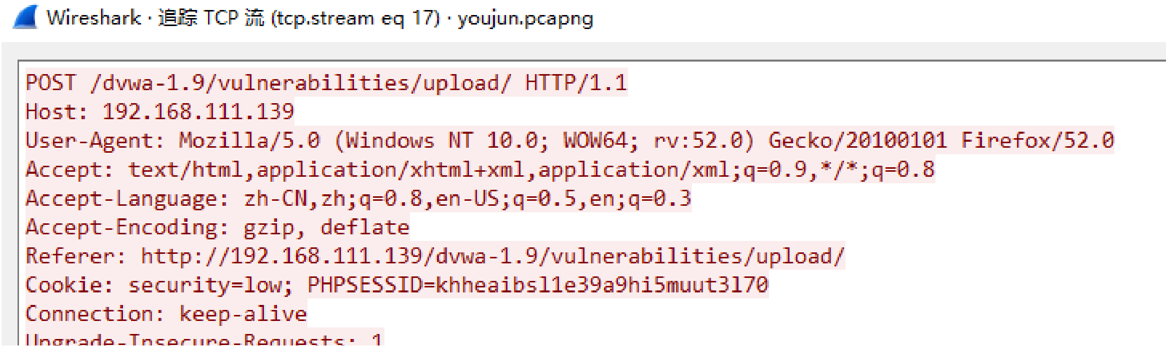
追踪TCP流，发现是个DVWA的练习流量包。



```
GET /dvwa-1.9/vulnerabilities/brute/ HTTP/1.1
Host: 192.168.111.139
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.111.139/dvwa-1.9/security.php
Cookie: security=low; PHPSESSID=khheaibsl1e39a9hi5muut3170
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

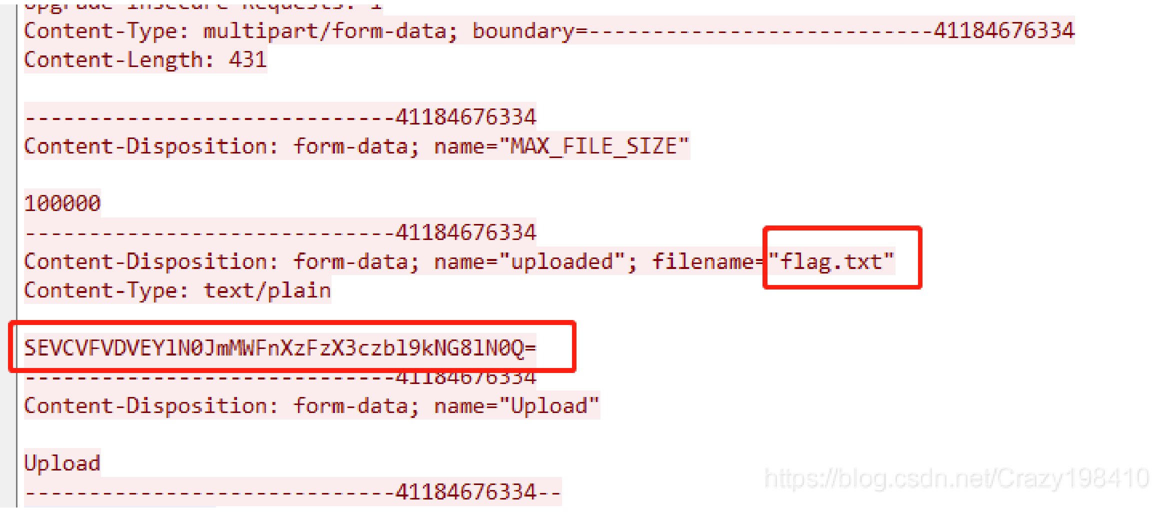
https://blog.csdn.net/Crazy198410

逐个流查看，在第17个流发现上传了一个flag.txt文件。文件内容是一串加密内容。



Wireshark · 追踪 TCP 流 (tcp.stream eq 17) · youjun.pcapng

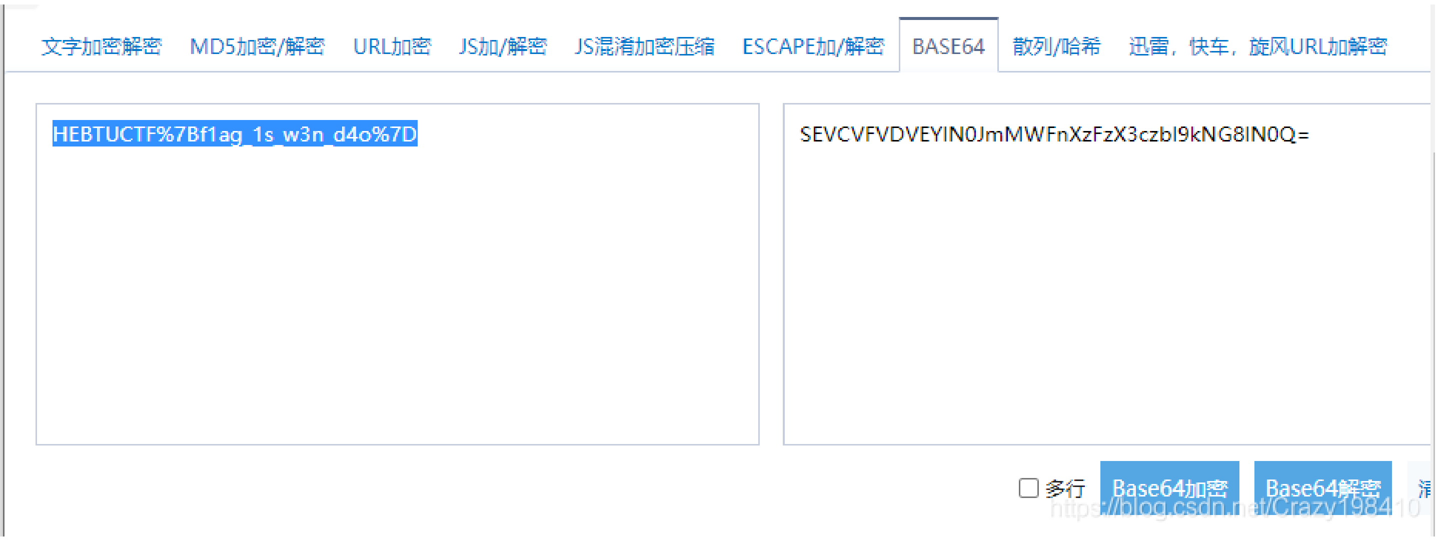
```
POST /dvwa-1.9/vulnerabilities/upload/ HTTP/1.1
Host: 192.168.111.139
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:52.0) Gecko/20100101 Firefox/52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://192.168.111.139/dvwa-1.9/vulnerabilities/upload/
Cookie: security=low; PHPSESSID=khheaibsl1e39a9hi5muut3170
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```



```
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=---------------------------41184676334
Content-Length: 431

-----------------------------41184676334
Content-Disposition: form-data; name="MAX_FILE_SIZE"

100000
-----------------------------41184676334
Content-Disposition: form-data; name="uploaded"; filename="flag.txt"
Content-Type: text/plain

SEVCVFVDVEYlN0JmMWFnXzFzX3czbl9kNG8lN0Q=
-----------------------------41184676334
Content-Disposition: form-data; name="Upload"

Upload
-----------------------------41184676334--
```

https://blog.csdn.net/Crazy198410

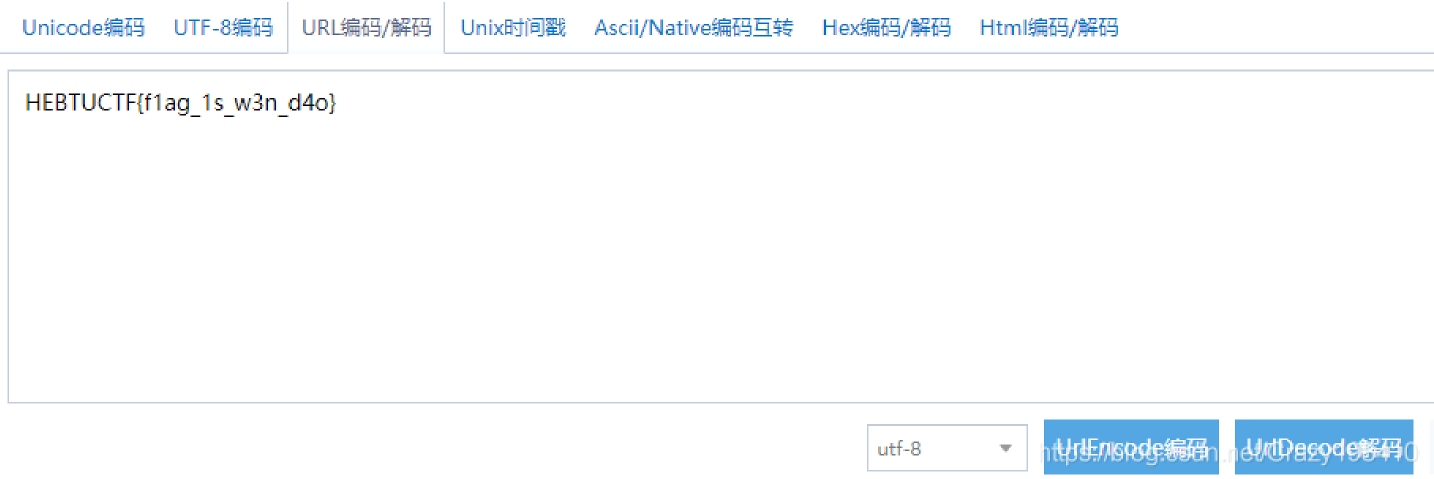
观察密文，看到有大小写字母，数字和“=”，初步判断为base64加密，使用在线解密尝试，再url解码，得到flag。



HEBTUCTF%7Bf1ag_1s_w3n_d4o%7D

SEVCVFVDVEYlN0JmMWFnXzFzX3czbl9kNG8lN0Q=

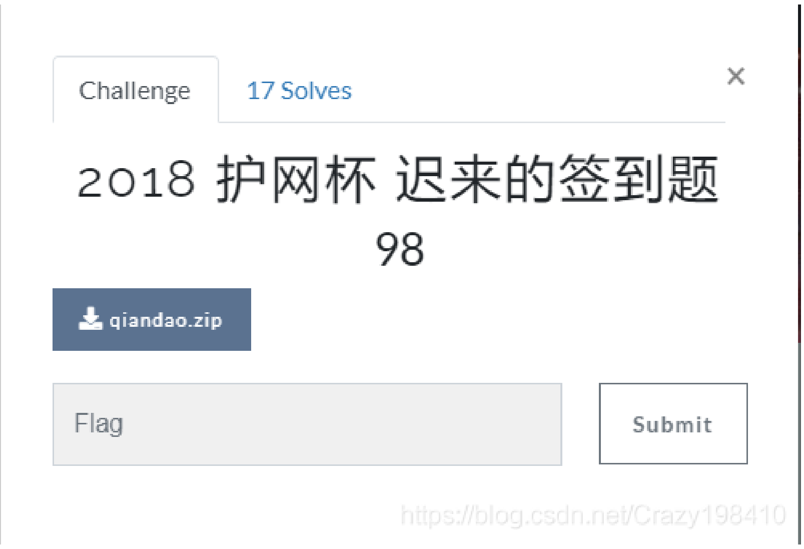
https://blog.csdn.net/Crazy198410



HEBTUCTF{f1ag_1s_w3n_d4o}

https://blog.csdn.net/Crazy198410

2018 护网杯 迟来的签到题

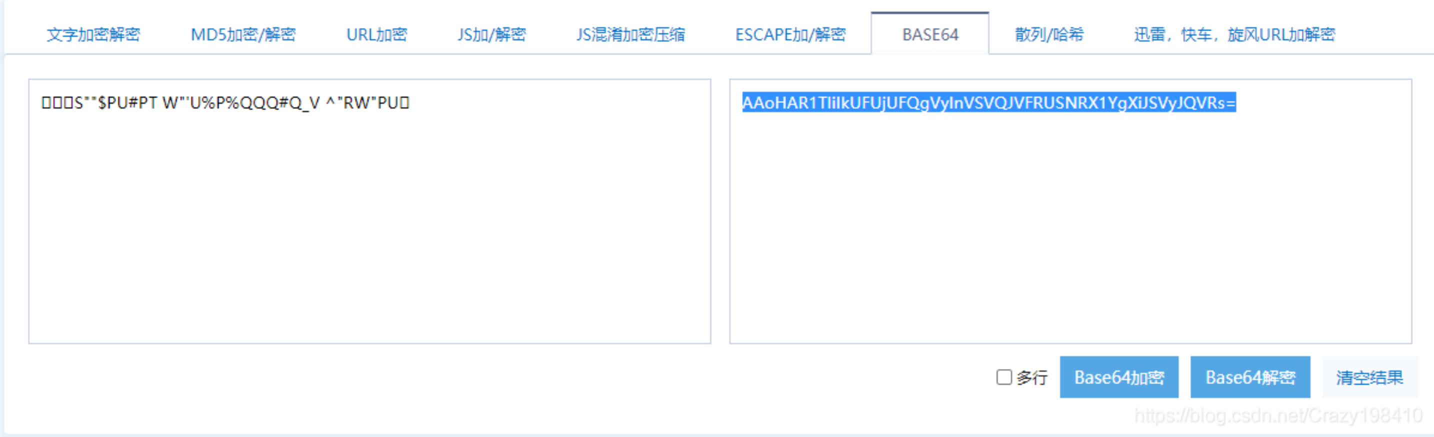


2018 护网杯 迟来的签到题

98

https://blog.csdn.net/Crazy198410

下载附件，解压后是一个文本，里面是一串密文：`AAoHAR1TIiIkUFUjUFQgVyInVSVQJVFRUSNRX1YgXiJSVyJQVRs=`，初步判断是base64密文，用base64解密后是一串乱码：



```
□□□S""$PU#PT W"'U%P%QQQ#Q_V ^"RW"PU□
```

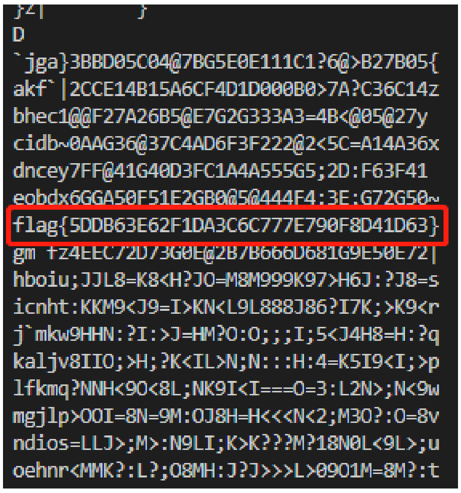
```
AAoHAR1TIiIkUFUjUFQgVyInVSVQJVFRUSNRX1YgXiJSVyJQVRs=
```

https://blog.csdn.net/Crazy198410

查看压缩包内题目提示：`easy xor???`
可以判断是要异或后才能得到flag。写python脚本

```
import base64
str1 = 'AAoHAR1TIiIkUFUjUFQgVyInVSVQJVFRUSNRX1YgXiJSVyJQVRs='

str2 = base64.b64decode(str1)
for i in range(200):
    tmp=''
    for j in str2:
        tmp += chr(j^i)
    print (tmp)
```

运行后，可以得到flag

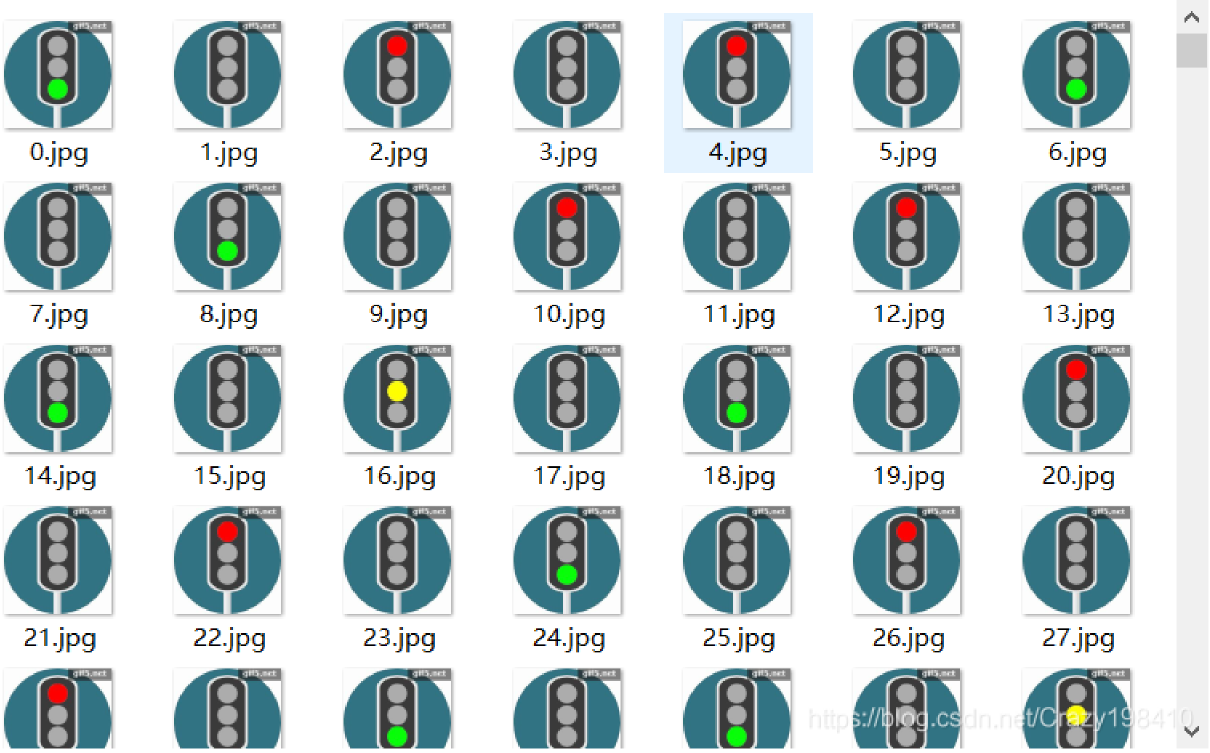


```
}z|         }
D
`jga}3BBD05C04@7BG5E0E111C1?6@>B27B05{
akf`|2CCE14B15A6CF4D1D000B0>7A?C36C14z
bhec1@@F27A26B5@E7G2G333A3=4B<@05@27y
cidb~0AAG36@37C4AD6F3F222@2<5C=A14A36x
dncey7FF@41G40D3FC1A4A555G5;2D:F63F41
eobdx6GGA50F51E2GB0@5@444F4:3E:G72G50~
flag{5DDB63E62F1DA3C6C777E790F8D41D63}
gm fz4EEC72D73G0E@2B7B666D681G9E50E72|
hboiu;JJL8=K8<H?JO=M8M999K97>H6J:?J8=s
icnht:KKM9<J9=I>KN<L9L888J86?I7K;>K9<r
j`mkw9HHN:?I:>J=HM?O:O;;;I;5<J4H8=H:?q
kaljv8IIO;>H;?K<IL>N;N:::H:4=K5I9<I;>p
lfkmq?NNH<9O<8L;NK9I<I===O=3:L2N>;N<9w
mgjlp>OOI=8N=9M:OJ8H=H<<<N<2;M3O?:O=8v
ndios=LLJ>;M>:N9LI;K>K???M?18N0L<9L>;u
oehnr<MMK?:L?;O8MH:J?J>>>L>0901M=8M?:t
```

Traffic_Light

下载是一张图片

对图片进行分解
得到1688张图片



观察图片，
1、发现第2的倍数的图片都是没有灯亮的。忽略不计。
2、绿灯和红灯总和为8或8的倍数时，下一张一定是黄色。
由此可以推断为二进制。绿为`1`，红为`0`。
写脚本进行编码：

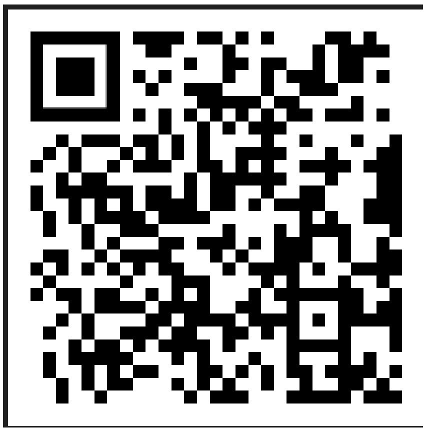
```
# -*-coding: utf-8 -*-
from PIL import Image

binstr = ""
flag = ""

def decode(s):
    return ''.join([chr(i) for i in [int(b, 2) for b in s.split(' ')]])

for i in range(1168):
    image=Image.open(r'./202011/'+str(i)+'.jpg')
    # print (image.getpixel((115,55)))#输出颜色值
    # print (image.getpixel((115,145)))
    tmp1 = image.getpixel((115,55))
    tmp2 = image.getpixel((115,150))
    # print (type(tmp1))
    if(tmp1[0] > 250):
        binstr += '1'
    elif(tmp2[1] > 250):
        binstr += '0'
    else:
        binstr += ''
print (binstr)

for i in range(len(binstr)):
    if i%8==0:
        flag +=decode(binstr[i:i+8])
print(flag)
```

得到flag

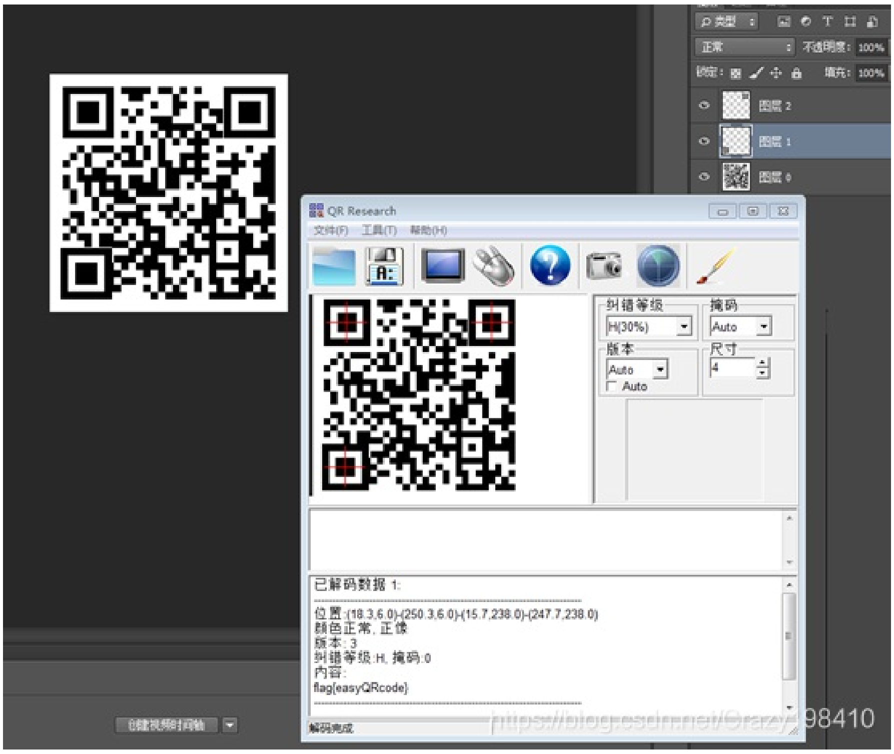
```
0110011001101100011000010110011101111011010100000110110000110011001101000111001100110011010111110111000000110100
0111100101011111001101000111010001110100001100110110111001110100001100010011000001101110010111110111010000110000
0101111101110100011100100011010001100110011001100011000101100011010111110111001100110100011001100011001101110100
0111100101011111011101110110100000110011011011100101111101111001001100000111010101011111001101000111001000110011
010111110011000001110101011101000111001100110001011001000011001101111101
flag{Pl34s3_p4y_4tt3nt10n_t0_tr4ff1c_s4f3ty_wh3n_y0u_4r3_0uts1d3}
```

Fix it

下载后是一张二维码图片，但只有一个黑框



使用photoshop进行修补，并识别：



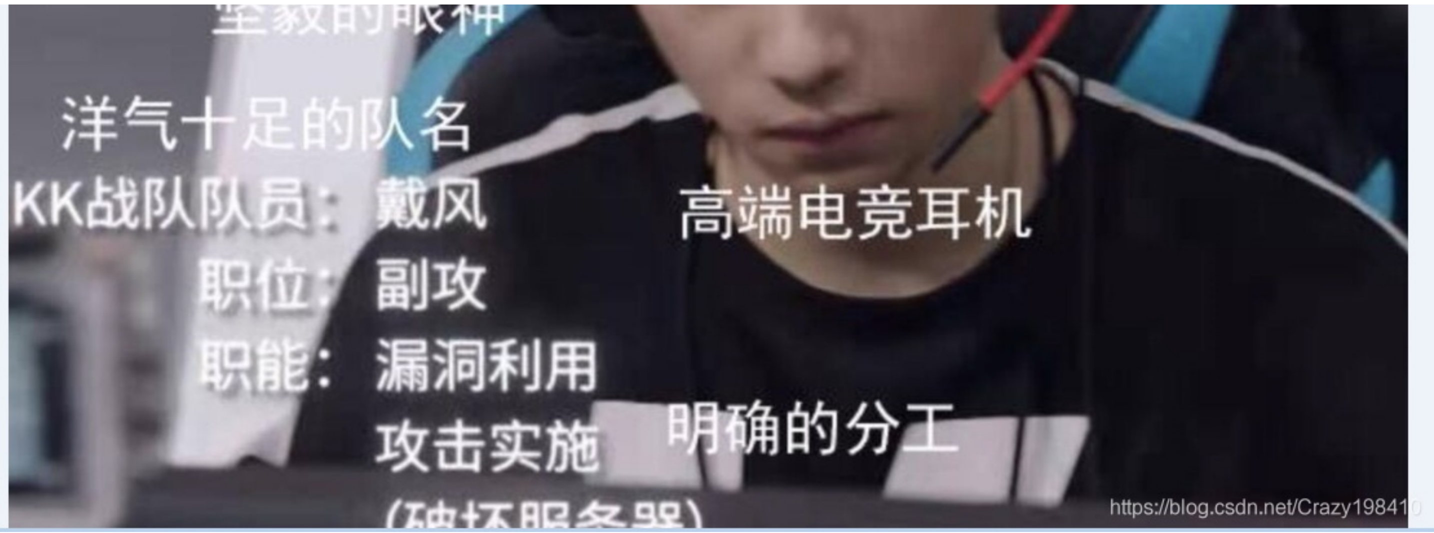
纠错等级 H(30%)
掩码 Auto
版本 Auto
□ Auto
尺寸 4

```
已解码数据 1:
------------------------------------------------------------
位置:(18.3,6.0)-(250.3,6.0)-(15.7,238.0)-(247.7,238.0)
颜色正常, 正像
版本: 3
纠错等级:H, 掩码:0
内容:
flag{easyQRcode}
------------------------------------------------------------
```

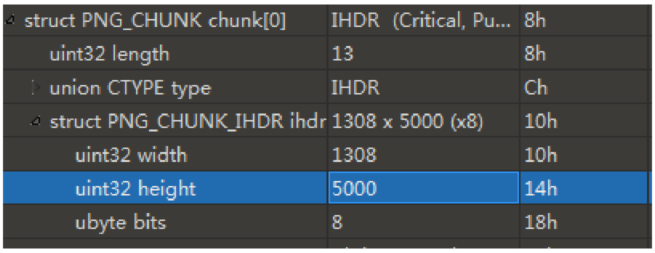
得到flag：flag{easyQRcode}

真正的CTFer

下载附件为一张图片



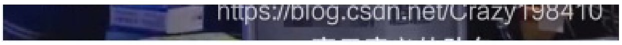
修改高度：



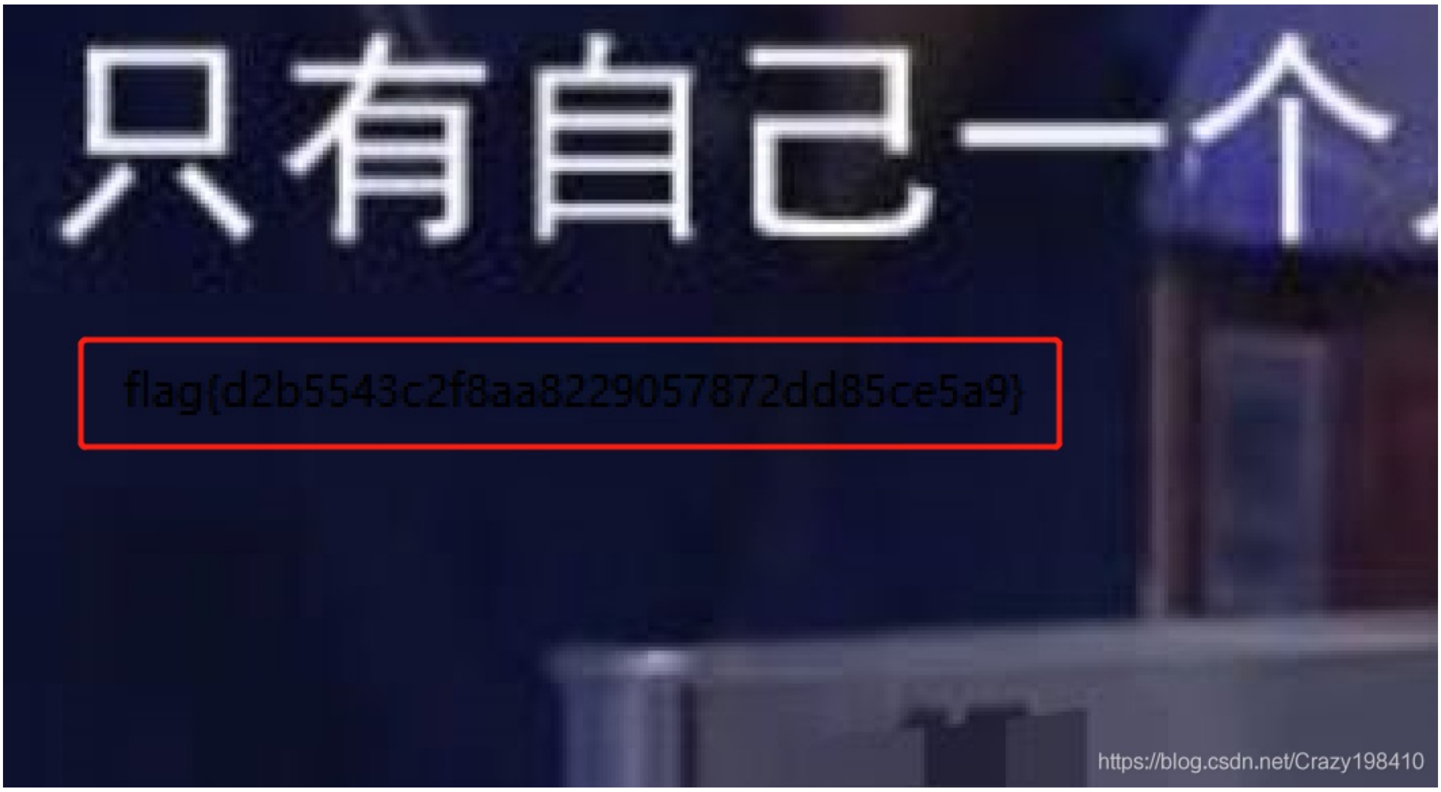
struct PNG_CHUNK chunk[0]	IHDR (Critical, Pu...	8h
uint32 length	13	8h
union CTYPE type	IHDR	Ch
struct PNG_CHUNK_IHDR ihdr	1308 x 5000 (x8)	10h
uint32 width	1308	10h
uint32 height	5000	14h
ubyte bits	8	18h

可以看到下面还有一张图片

真正的CTF选手
早起后精心打理的发型
傲风电竞椅
坚毅的眼神
洋气十足的队名
KK战队队员：戴风
职位：副攻
职能：漏洞利用
攻击实施
(破坏服务器)
高端电竞耳机
明确的分工
SP战队在决胜局中
性能超强的台式电脑
虚假的CTF选手
一早起来就没打理的懒散发型
迷茫的眼神
不知道什么牌子的椅子
只有自己一个人，分工不明确
性能不高的笔记本电脑
散乱的卫生纸



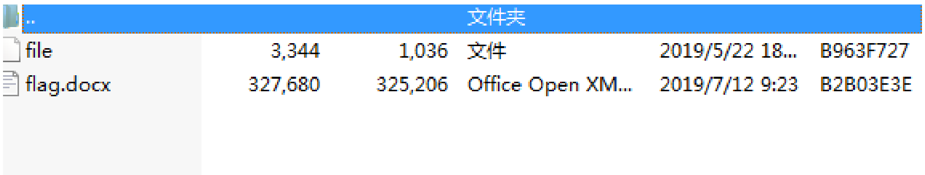
将图片放大可以看到flag：


只有自己一个
flag{d2b5543c2f8aa8229057872dd85ce5a9}
https://blog.csdn.net/Crazy198410

仔细看能看到flag
flag{d2b5543c2f8aa8229057872dd85ce5a9}

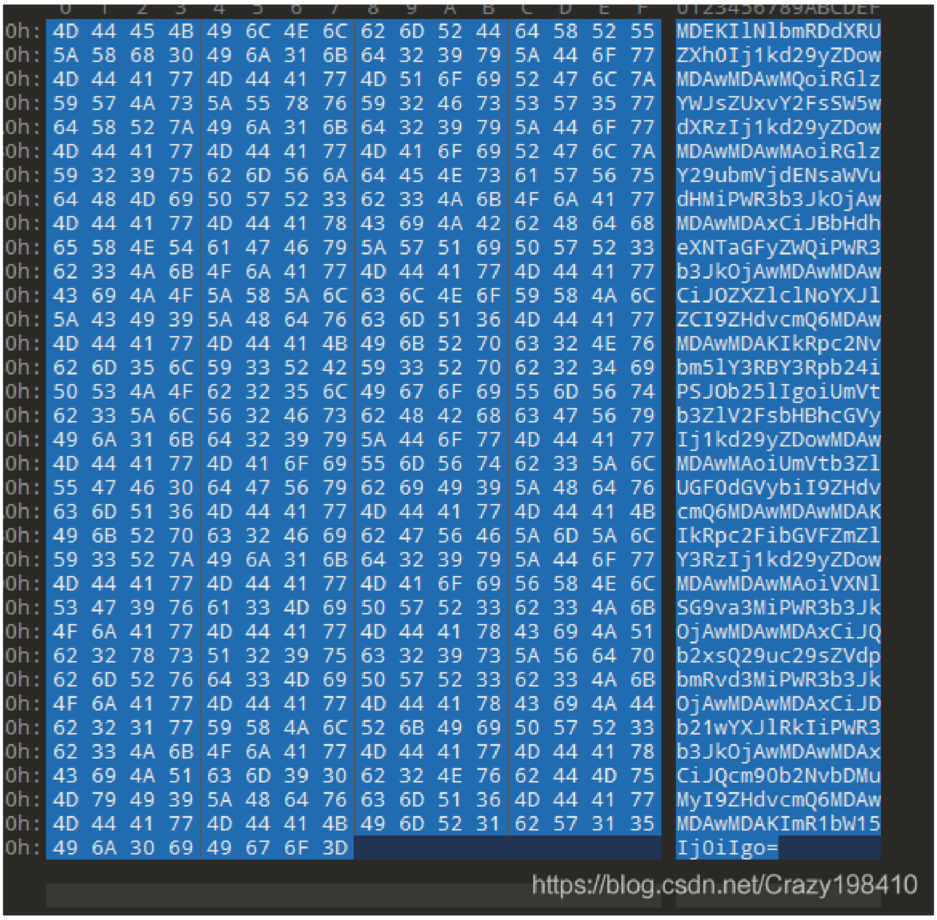
解不开的秘密

下载后是一个压缩包，里有一个file文件，和一个flag.docx文本



..			文件夹		
file	3,344	1,036	文件	2019/5/22 18...	B963F727
flag.docx	327,680	325,206	Office Open XM...	2019/7/12 9:23	B2B03E3E

用文本工具打开file文件，里面有许多数字和少许的英文。怀疑是十六进制
放入010editor中



```
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0h: 4D 44 45 4B 49 6C 4E 6C 62 6D 52 44 64 58 52 55  MDEKIlNlbmRDdXRU
0h: 5A 58 68 30 49 6A 31 6B 64 32 39 79 5A 44 6F 77  ZXh0Ij1kd29yZDow
0h: 4D 44 41 77 4D 44 41 77 4D 51 6F 69 52 47 6C 7A  MDAwMDAwMQoiRGlz
0h: 59 57 4A 73 5A 55 78 76 59 32 46 73 53 57 35 77  YWJsZUxvY2FsSW5w
0h: 64 58 52 7A 49 6A 31 6B 64 32 39 79 5A 44 6F 77  dXRzIj1kd29yZDow
0h: 4D 44 41 77 4D 44 41 77 4D 41 6F 69 52 47 6C 7A  MDAwMDAwMAoiRGlz
0h: 59 32 39 75 62 6D 56 6A 64 45 4E 73 61 57 56 75  Y29ubmVjdENsaWVu
0h: 64 48 4D 69 50 57 52 33 62 33 4A 6B 4F 6A 41 77  dHMiPWR3b3JkOjAw
0h: 4D 44 41 77 4D 44 41 78 43 69 4A 42 62 48 64 68  MDAwMDAxCiJBbHdh
0h: 65 58 4E 54 61 47 46 79 5A 57 51 69 50 57 52 33  eXNTaGFyZWQiPWR3
0h: 62 33 4A 6B 4F 6A 41 77 4D 44 41 77 4D 44 41 77  b3JkOjAwMDAwMDAw
0h: 43 69 4A 4F 5A 58 5A 6C 63 6C 4E 6F 59 58 4A 6C  CiJOZXZlclNoYXJl
0h: 5A 43 49 39 5A 48 64 76 63 6D 51 36 4D 44 41 77  ZCI9ZHdvcmQ6MDAw
0h: 4D 44 41 77 4D 44 41 4B 49 6B 52 70 63 32 4E 76  MDAwMDAKIkRpc2Nv
0h: 62 6D 35 6C 59 33 52 42 59 33 52 70 62 32 34 69  bm5lY3RBY3Rpb24i
0h: 50 53 4A 4F 62 32 35 6C 49 67 6F 69 55 6D 56 74  PSJOb25lIgoiUmVt
0h: 62 33 5A 6C 56 32 46 73 62 48 42 68 63 47 56 79  b3ZlV2FsbHBhcGVy
0h: 49 6A 31 6B 64 32 39 79 5A 44 6F 77 4D 44 41 77  Ij1kd29yZDowMDAw
0h: 4D 44 41 77 4D 41 6F 69 55 6D 56 74 62 33 5A 6C  MDAwMAoiUmVtb3Zl
0h: 55 47 46 30 64 47 56 79 62 69 49 39 5A 48 64 76  UGF0dGVybiI9ZHdv
0h: 63 6D 51 36 4D 44 41 77 4D 44 41 77 4D 44 41 4B  cmQ6MDAwMDAwMDAK
0h: 49 6B 52 70 63 32 46 69 62 47 56 46 5A 6D 5A 6C  IkRpc2FibGVFZmZl
0h: 59 33 52 7A 49 6A 31 6B 64 32 39 79 5A 44 6F 77  Y3RzIj1kd29yZDow
0h: 4D 44 41 77 4D 44 41 77 4D 41 6F 69 56 58 4E 6C  MDAwMDAwMAoiVXNl
0h: 53 47 39 76 61 33 4D 69 50 57 52 33 62 33 4A 6B  SG9va3MiPWR3b3Jk
0h: 4F 6A 41 77 4D 44 41 77 4D 44 41 78 43 69 4A 51  OjAwMDAwMDAxCiJQ
0h: 62 32 78 73 51 32 39 75 63 32 39 73 5A 56 64 70  b2xsQ29uc29sZVdp
0h: 62 6D 52 76 64 33 4D 69 50 57 52 33 62 33 4A 6B  bmRvd3MiPWR3b3Jk
0h: 4F 6A 41 77 4D 44 41 77 4D 44 41 78 43 69 4A 44  OjAwMDAwMDAxCiJD
0h: 62 32 31 77 59 58 4A 6C 52 6B 49 69 50 57 52 33  b21wYXJlRkIiPWR3
0h: 62 33 4A 6B 4F 6A 41 77 4D 44 41 77 4D 44 41 78  b3JkOjAwMDAwMDAx
0h: 43 69 4A 51 63 6D 39 30 62 32 4E 76 62 44 4D 75  CiJQcm90b2NvbDMu
0h: 4D 79 49 39 5A 48 64 76 63 6D 51 36 4D 44 41 77  MyI9ZHdvcmQ6MDAw
0h: 4D 44 41 77 4D 44 41 4B 49 6D 52 31 62 57 31 35  MDAwMDAKImR1bW15
0h: 49 6A 30 69 49 67 6F 3D                          IjOiIgo=
```

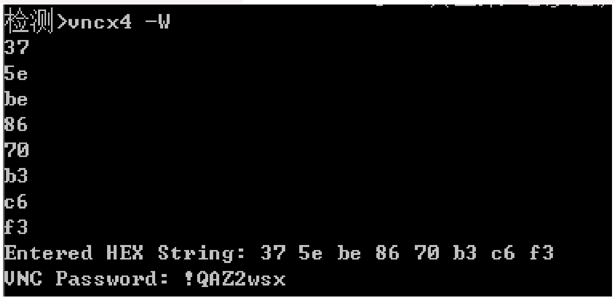
https://blog.csdn.net/Crazy198410

看到最后有个"="，怀疑是base64。用base64解码：

```
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Software\RealVNC]

[HKEY_CURRENT_USER\Software\RealVNC\vnclicensewiz]
"_AnlClientId"="8f5cc378-2e1d-4670-80e0-d2d81d882561"
"_AnlSelected"="0"
"_AnlInclRate"="0.0025"

[HKEY_CURRENT_USER\Software\RealVNC\vncserver]

[HKEY_CURRENT_USER\Software\RealVNC\VNCViewer4]
"dummy"=""

[HKEY_CURRENT_USER\Software\RealVNC\VNCViewer4\MRU]
"00"="127.0.0.1"
"Order"=hex:00,01
"01"="127.0.0.1:5900"

[HKEY_CURRENT_USER\Software\RealVNC\WinVNC4]
"Password"=hex:37,5e,be,86,70,b3,c6,f3
"SecurityTypes"="VncAuth"
"ReverseSecurityTypes"="None"
"QueryConnect"=dword:00000000
"PortNumber"=dword:0000170c
"LocalHost"=dword:00000000
"IdleTimeout"=dword:00000e10
"HTTPPortNumber"=dword:000016a8
"Hosts"="+,"
"AcceptKeyEvents"=dword:00000001
"AcceptPointerEvents"=dword:00000001
"AcceptCutText"=dword:00000001
"SendCutText"=dword:00000001
"DisableLocalInputs"=dword:00000000
"DisconnectClients"=dword:00000001
"AlwaysShared"=dword:00000000
"NeverShared"=dword:00000000
"DisconnectAction"="None"
"RemoveWallpaper"=dword:00000000
"RemovePattern"=dword:00000000
"DisableEffects"=dword:00000000
"UseHooks"=dword:00000001
"PollConsoleWindows"=dword:00000001
"CompareFB"=dword:00000001
"Protocol3.3"=dword:00000000
"dummy"=""
```

可以看到中间位置有`"Password"=hex:37,5e,be,86,70,b3,c6,f3`

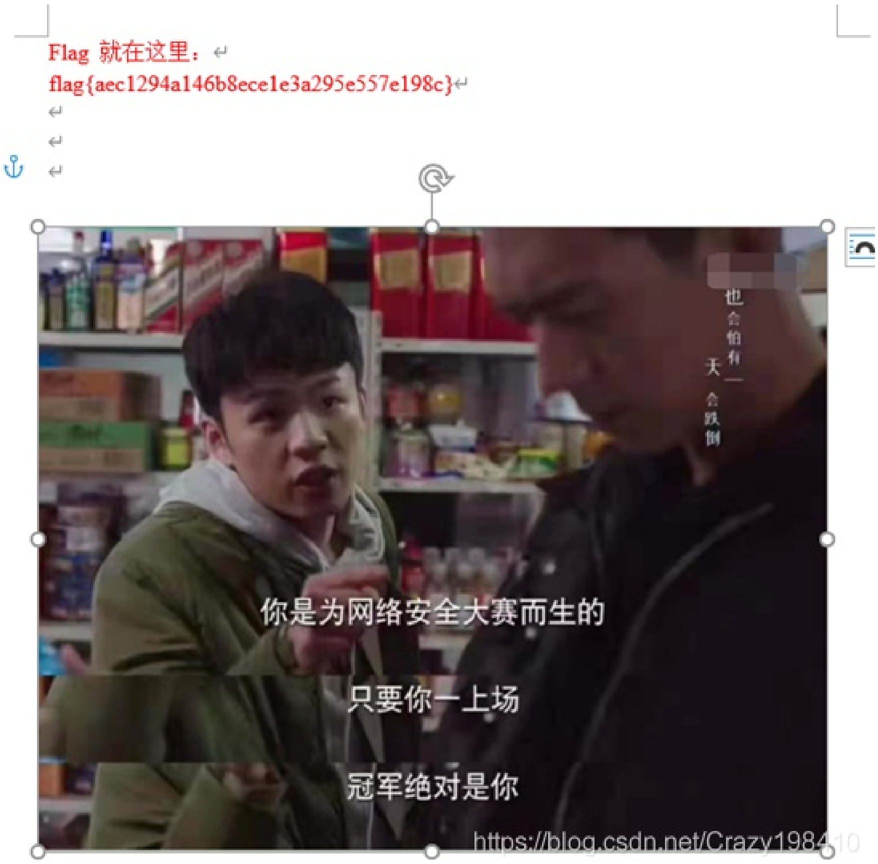
开头`[HKEY_CURRENT_USER\Software\RealVNC\vncserver]`中有RealVNC。于是用`Vccx4.exe`进行破解：



```
检测>vncx4 -W
37
5e
be
86
70
b3
c6
f3
Entered HEX String: 37 5e be 86 70 b3 c6 f3
VNC Password: !QAZ2wsx
```

得到密码：`!QAZ2wsx`

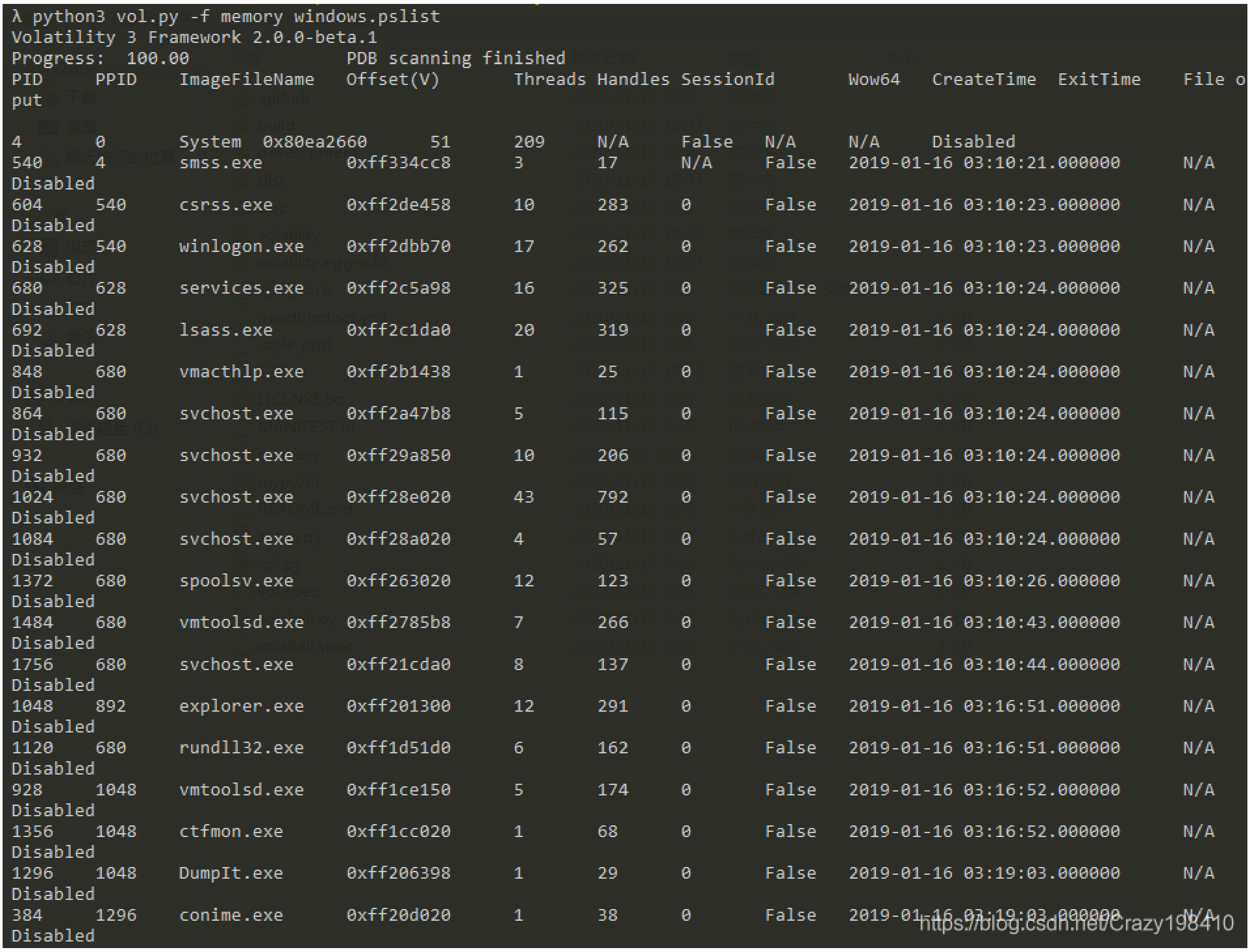
用密码打开flag.docx，将图片移开，并全选修改字体颜色，可以看到flag



Flag 就在这里：
flag{aec1294a146b8ece1e3a295e557e198c}

memory

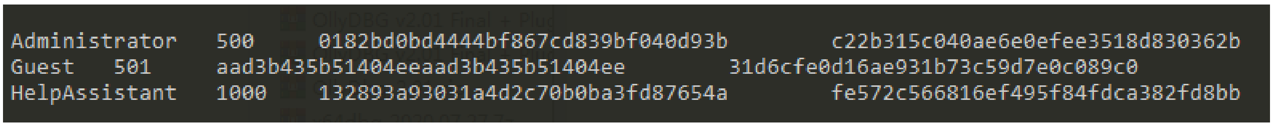
下载附件后，用`volatility`进行分析



```
λ python3 vol.py -f memory windows.pslist
Volatility 3 Framework 2.0.0-beta.1
Progress:  100.00               PDB scanning finished
PID     PPID    ImageFileName   Offset(V)       Threads Handles SessionId       Wow64   CreateTime      ExitTime        File o
put

4       0       System  0x80ea2660      51      209     N/A     False   N/A     N/A     Disabled
540     4       smss.exe        0xff334cc8      3       17      N/A     False   2019-01-16 03:10:21.000000      N/A
Disabled
604     540     csrss.exe       0xff2de458      10      283     0       False   2019-01-16 03:10:23.000000      N/A
Disabled
628     540     winlogon.exe    0xff2dbb70      17      262     0       False   2019-01-16 03:10:23.000000      N/A
Disabled
680     628     services.exe    0xff2c5a98      16      325     0       False   2019-01-16 03:10:24.000000      N/A
Disabled
692     628     lsass.exe       0xff2c1da0      20      319     0       False   2019-01-16 03:10:24.000000      N/A
Disabled
848     680     vmacthlp.exe    0xff2b1438      1       25      0       False   2019-01-16 03:10:24.000000      N/A
Disabled
864     680     svchost.exe     0xff2a47b8      5       115     0       False   2019-01-16 03:10:24.000000      N/A
Disabled
932     680     svchost.exe     0xff29a850      10      206     0       False   2019-01-16 03:10:24.000000      N/A
Disabled
1024    680     svchost.exe     0xff28e020      43      792     0       False   2019-01-16 03:10:24.000000      N/A
Disabled
1084    680     svchost.exe     0xff28a020      4       57      0       False   2019-01-16 03:10:24.000000      N/A
Disabled
1372    680     spoolsv.exe     0xff263020      12      123     0       False   2019-01-16 03:10:26.000000      N/A
Disabled
1484    680     vmtoolsd.exe    0xff2785b8      7       266     0       False   2019-01-16 03:10:43.000000      N/A
Disabled
1756    680     svchost.exe     0xff21cda0      8       137     0       False   2019-01-16 03:10:44.000000      N/A
Disabled
1048    892     explorer.exe    0xff201300      12      291     0       False   2019-01-16 03:16:51.000000      N/A
Disabled
1120    680     rundll32.exe    0xff1d51d0      6       162     0       False   2019-01-16 03:16:51.000000      N/A
Disabled
928     1048    vmtoolsd.exe    0xff1ce150      5       174     0       False   2019-01-16 03:16:52.000000      N/A
Disabled
1356    1048    ctfmon.exe      0xff1cc020      1       68      0       False   2019-01-16 03:16:52.000000      N/A
Disabled
1296    1048    DumpIt.exe      0xff206398      1       29      0       False   2019-01-16 03:19:03.000000      N/A
Disabled
384     1296    conime.exe      0xff20d020      1       38      0       False   2019-01-16 03:19:03.000000      N/A
Disabled
```

https://blog.csdn.net/Crazy198410

题目是“分析内存镜像,破解管理员的登录密码,flag为明文密码的MD5值”
我们分析镜像中的hash值：



```
Administrator   500     0182bd0bd4444bf867cd839bf040d93b        c22b315c040ae6e0efee3518d830362b
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
HelpAssistant   1000    132893a93031a4d2c70b0ba3fd87654a        fe572c566816ef495f84fdca382fd8bb
```

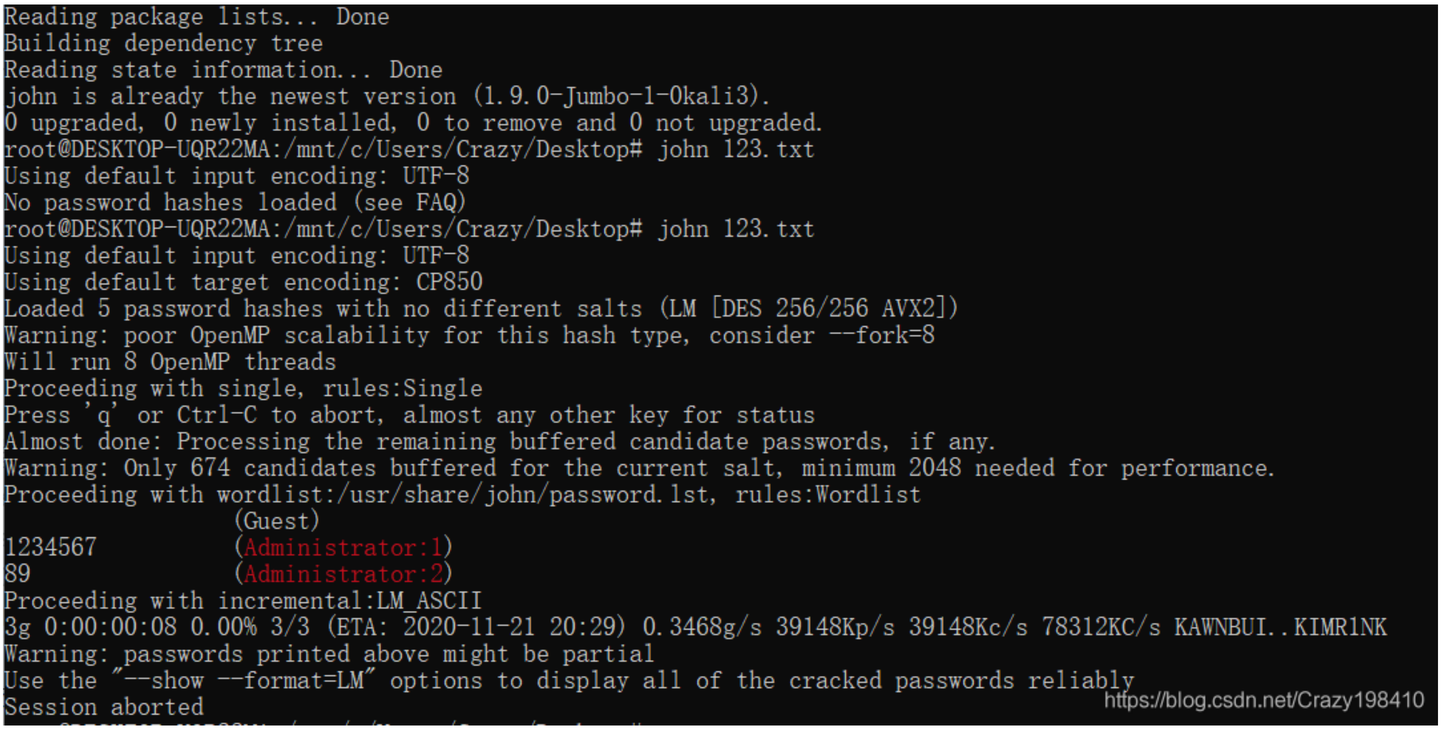
得到密文：

```
Administrator   500     0182bd0bd4444bf867cd839bf040d93b        c22b315c040ae6e0efee3518d830362b
Guest   501     aad3b435b51404eeaad3b435b51404ee        31d6cfe0d16ae931b73c59d7e0c089c0
HelpAssistant   1000    132893a93031a4d2c70b0ba3fd87654a        fe572c566816ef495f84fdca382fd8bb
```

进行修改将中间的" “改为”:"

```
Administrator:500:0182bd0bd4444bf867cd839bf040d93b:c22b315c040ae6e0efee3518d830362b
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0
HelpAssistant:1000:132893a93031a4d2c70b0ba3fd87654a        fe572c566816ef495f84fdca382fd8bb
```

存为文件，再用john进行爆破

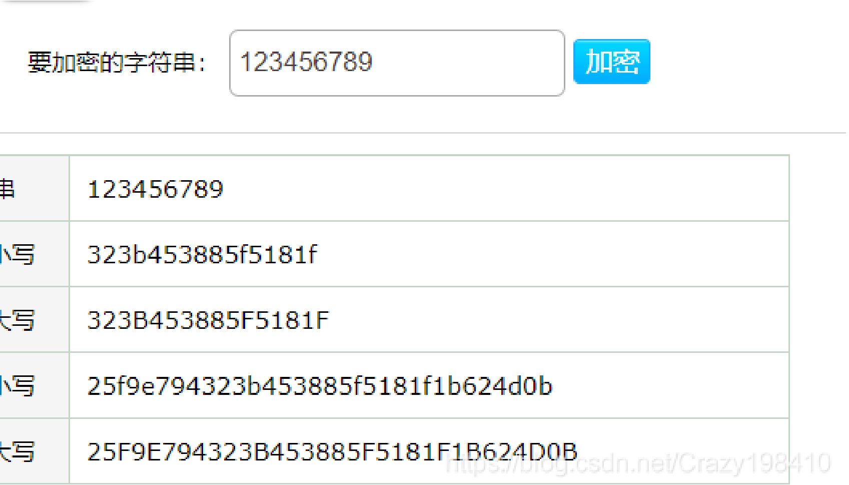


```
Reading package lists... Done
Building dependency tree
Reading state information... Done
john is already the newest version (1.9.0-Jumbo-1-0kali3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@DESKTOP-UQR22MA:/mnt/c/Users/Crazy/Desktop# john 123.txt
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
root@DESKTOP-UQR22MA:/mnt/c/Users/Crazy/Desktop# john 123.txt
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 5 password hashes with no different salts (LM [DES 256/256 AVX2])
Warning: poor OpenMP scalability for this hash type, consider --fork=8
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 674 candidates buffered for the current salt, minimum 2048 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
                 (Guest)
1234567          (Administrator:1)
89               (Administrator:2)
Proceeding with incremental:LM_ASCII
3g 0:00:00:08 0.00% 3/3 (ETA: 2020-11-21 20:29) 0.3468g/s 39148Kp/s 39148Kc/s 78312KC/s KAWNBUI..KIMR1NK
Warning: passwords printed above might be partial
Use the "--show --format=LM" options to display all of the cracked passwords reliably
Session aborted
```

https://blog.csdn.net/Crazy198410

可以得到密码 `123456789`

再md5加密，就是flag：



要加密的字符串： 123456789 加密

字符串	123456789
16位 小写	323b453885f5181f
16位 大写	323B453885F5181F
32位 小写	25f9e794323b453885f5181f1b624d0b
32位 大写	25F9E794323B453885F5181F1B624D0B

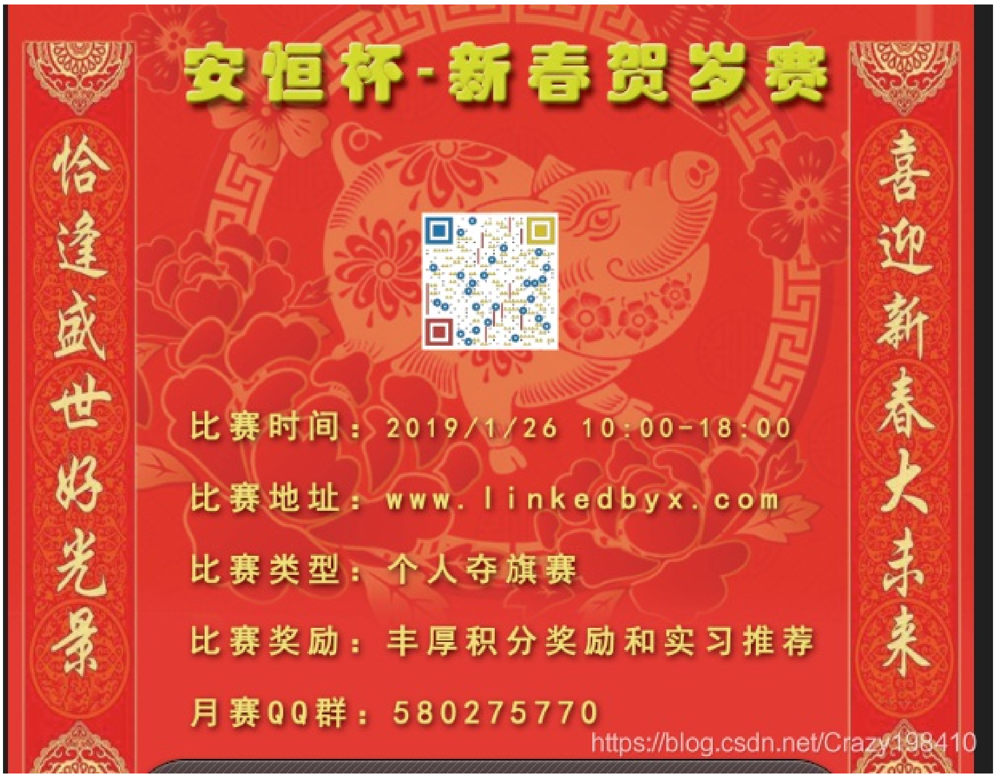
https://blog.csdn.net/Crazy198410

但提交错误，可能是题目的问题。

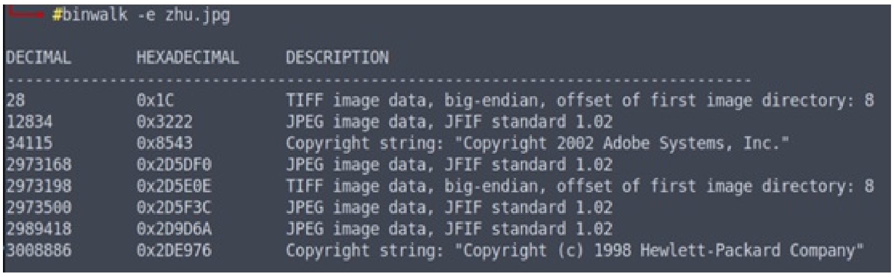
赢战2019

下载附件，为一张图片

赢战2019
大展宏图
福

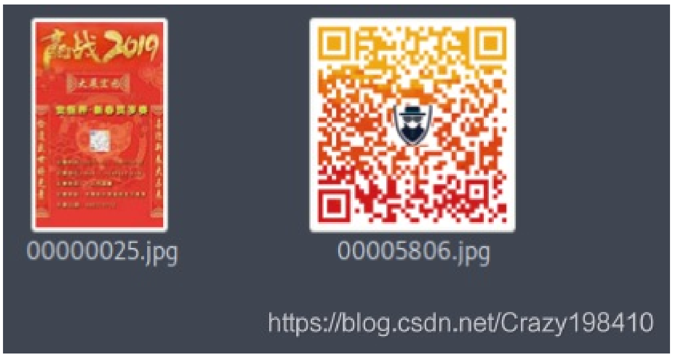

安恒杯-新春贺岁赛
恰逢盛世好光景
喜迎新春大未来
比赛时间：2019/1/26 10:00-18:00
比赛地址：www.linkedbyx.com
比赛类型：个人夺旗赛
比赛奖励：丰厚积分奖励和实习推荐
月赛QQ群：580275770
https://blog.csdn.net/Crazy198410

binwalk进行分析：

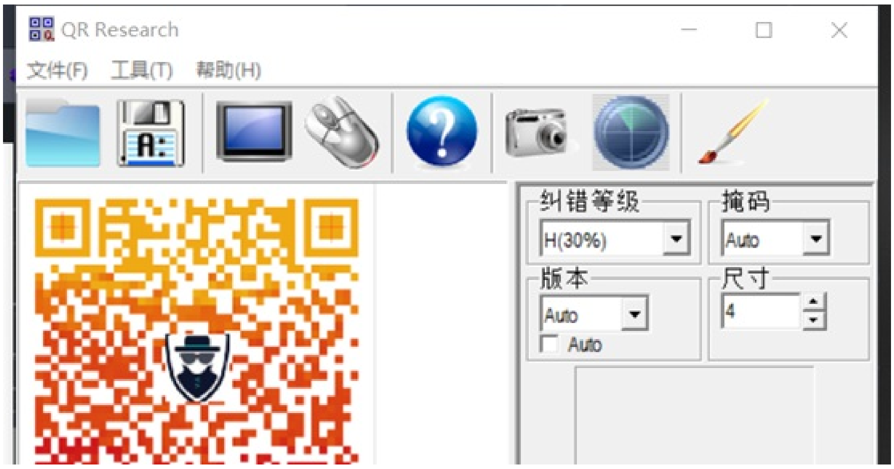


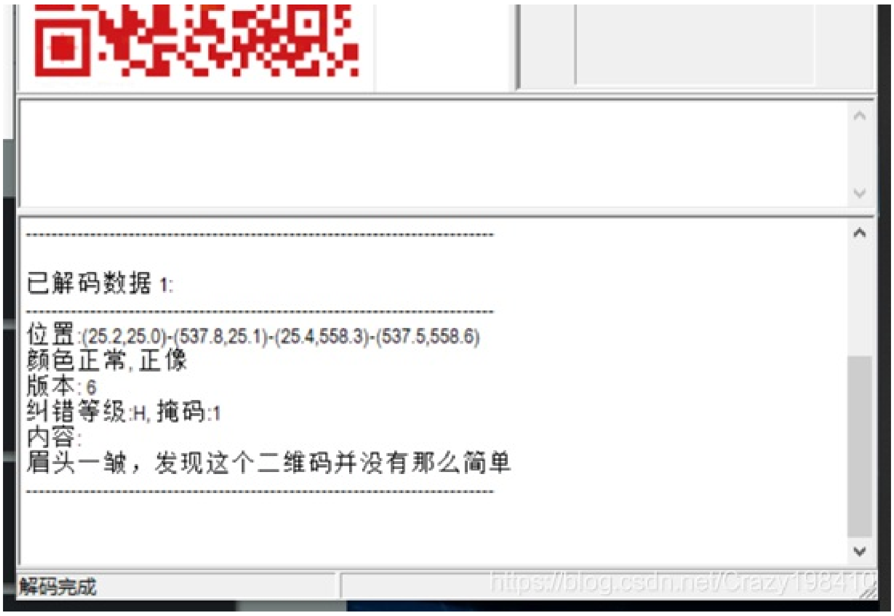
```
└── #binwalk -e zhu.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
28            0x1C            TIFF image data, big-endian, offset of first image directory: 8
12834         0x3222          JPEG image data, JFIF standard 1.02
34115         0x8543          Copyright string: "Copyright 2002 Adobe Systems, Inc."
2973168       0x2D5DF0        JPEG image data, JFIF standard 1.02
2973198       0x2D5E0E        TIFF image data, big-endian, offset of first image directory: 8
2973500       0x2D5F3C        JPEG image data, JFIF standard 1.02
2989418       0x2D9D6A        JPEG image data, JFIF standard 1.02
3008886       0x2DE976        Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
```

分解出两张图片



对二维码扫描：

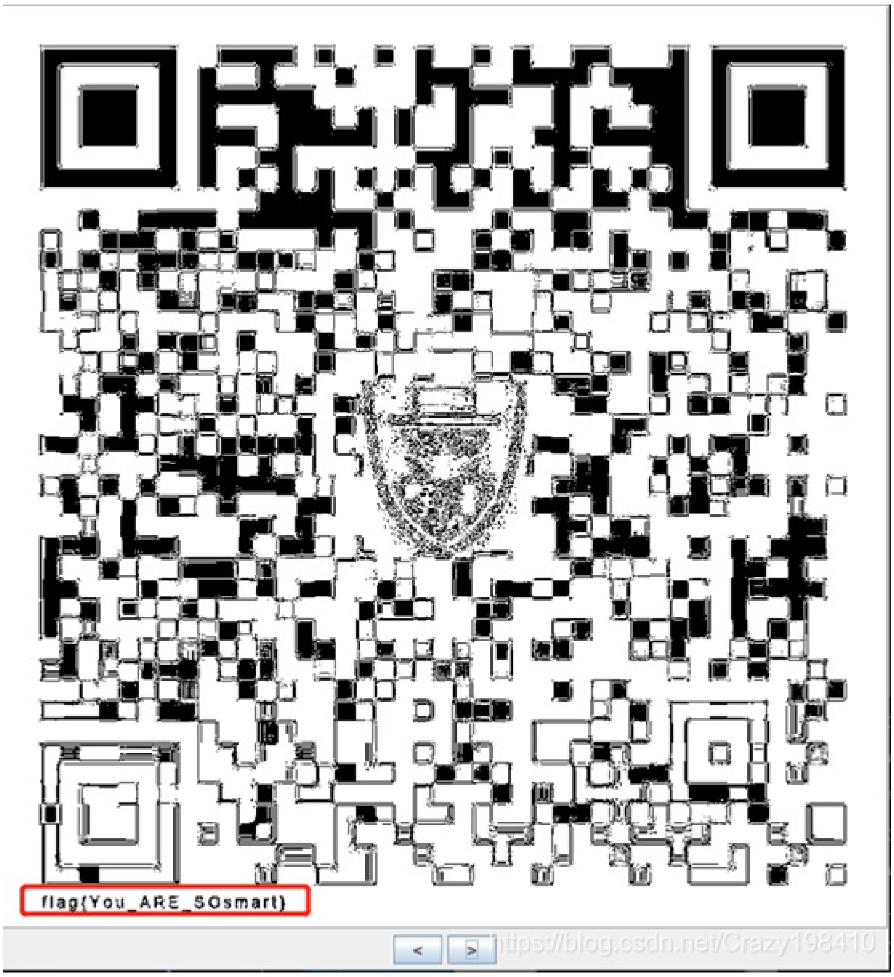




```
--------------------------------------------------------------------------------

已解码数据 1:
--------------------------------------------------------------------------------
位置:(25.2,25.0)-(537.8,25.1)-(25.4,558.3)-(537.5,558.6)
颜色正常, 正像
版本: 6
纠错等级:H, 掩码:1
内容:
眉头一皱，发现这个二维码并没有那么简单
--------------------------------------------------------------------------------
```

https://blog.csdn.net/Crazy198410

无有用信息。
再对二维码图片进行分析：

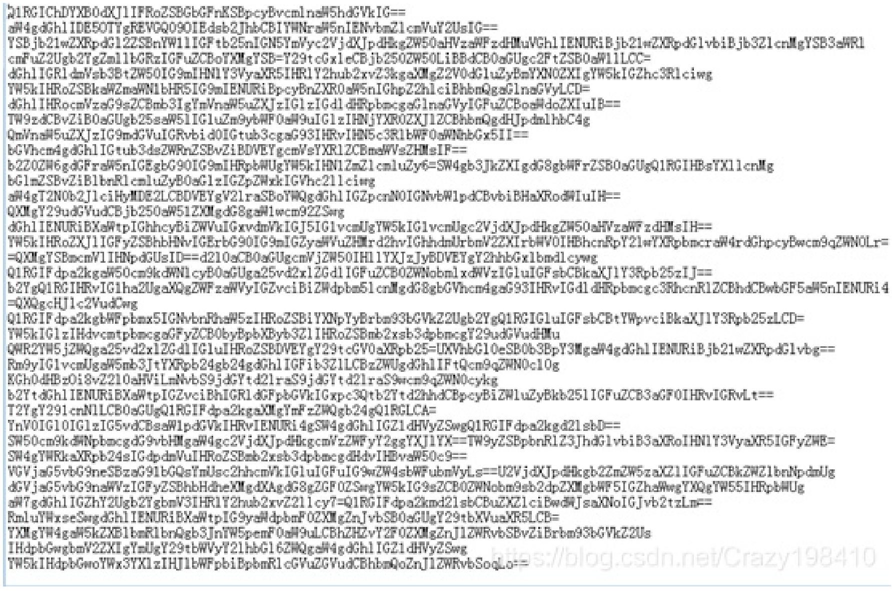


https://blog.csdn.net/Crazy198410

可以在左下角看到flag
flag{You_ARE_SOsmart}
要写全。不是{}内的。。。。

2020sdnisc-CTF的起源

下载附件为一个文本



```
Q1RGIChDYXB0dXJlIFRoZSBGbGFnKSBpcyBvcmlnaW5hdGVkIG==
aW4gdGhlIDE5OTYgREVGQ09OIEdsb2JhbCBIYWNraW5nIENvbmZlcmVuY2UsIG==
YSBjb21wZXRpdGl2ZSBnYW1lIGFtb25nIGN5YmVyc2VjdXJpdHkgZW50aHVzaWFzdHMuVGhlIENURiBjb21wZXRpdGlvbiBjb3ZlcnMgYSB3aWRl
cmFuZ2Ugb2YgZmllbGRzIGFuZCBoYXMgYSB=Y29tcGxleCBjb250ZW50LiBBdCB0aGUgc2FtZSB0aW1lLCC=
dGhlIGRldmVsb3BtZW50IG9mIHN1Y3VyaXR5IHRlY2hub2xvZ3kgaXMgZ2V0dGluZyBmYXN0ZXIgYW5kIGZhc3Rlciwg
YW5kIHRoZSBkaWZmaWN1bHR5IG9mIENURiBpcyBnZXR0aW5nIGhpZ2hlciBhbmQgaGlnaGVyLCD=
dGhlIHRocmVzaG9sZCBmb3IgYmVnaW5uZXJzIGlzIGdldHRpbmcgaGlnaGVyIGFuZCBoaWdoZXIuIB==
TW9zdCBvZiB0aGUgb25saW5lIGluZm9ybWF0aW9uIGlzIHNjYXR0ZXJlZCBhbmQgdHJpdmlhbC4g
QmVnaW5uZXJzIG9mdGVuIGRvbid0IGtub3cgaG93IHRvIHN5c3RlbWF0aWNhbGx5II==
bGVhcm4gdGhlIGtub3dsZWRnZSBvZiBDVEYgcmVsYXRlZCBmaWVsZHMsIF==
b2Z0ZW6gdGFraW5nIGEgbG90IG9mIHRpbWUgYW5kIHN1ZmZlcmluZy6=SW4gb3JkZXIgdG8gbWFrZSB0aGUgQ1RGIHBsYXllcnMg
bGlmZSBvZiBlbnRlcmluZyB0aGlzIGZpZWxkIGVhc2llciwg
aW4gT2N0b2JlciHyMDE2LCBDVEYgV2lraSBoYWQgdGhlIGZpcnN0IGNvbW1pdCBvbiBHaXRodWIuIH==
QXMgY29udGVudCBjb250aW51ZXMgdG8gaW1wcm92ZSwg
dGhlIENURiBXaWtpIGhhcyBiZWVuIGxvdmVkIGJ5IG1vcmUgYW5kIG1vcmUgc2VjdXJpdHkgZW50aHVzaWFzdHMsIH==
YW5kIHRoZXJlIGFyZSBhbHMvIGErbG90IG9mIGZyaWVuZHMrd2hvIGhhdmUrbmV2ZXIrbWV0IHBhcnRpY2lwYXRpbmcraW4rdGhpcyBwcm9qZWN0Lr=
=QXMgYSBmcmVlIHNpdGUsID==d2l0aCB0aGUgcmVjZW50IHl1YXJzJyBDVEYgY2hhbGxlbmdlcywg
Q1RGIFdpa2kgaW50cm9kdWNlcyB0aGUga25vd2xlZGdlIGFuZCB0ZWNobmlxdWVzIGluIGFsbCBkaXJlY3Rpb25zIIJ==
b2YgQ1RGIHRvIG1ha2UgaXQgZWFzaWVyIGZvciBiZWdpbm5lcnMgdG8gbGVhcm4gaG93IHRvIGdldHRpbmcgc3RhcnRlZCBhdCBwbGF5aW5nIENURi4
=QXQgcHJlc2VudCwg
Q1RGIFdpa2kgbWFpbmx5IGNvbnRhaW5zIHRoZSBiYXNpYyBrbm93bGVkZ2Ugb2YgQ1RGIGluIGFsbCBtYWpvciBkaXJlY3Rpb25zLCD=
YW5kIG1zIHdvcmtpbmcgaGFyZCB0byBpbXByb3ZlIHRoZSBmb2xsb3dpbmcgY29udGVudHMu
QWR2YW5jZWQga25vd2xlZGdlIGluIHRoZSBDVEYgY29tcGV0aXRpb25=UXVhbGl0eSB0b3BpY3MgaW4gdGhlIENURiBjb21wZXRpdGlvbg==
Rm9yIG1vcmUgaW5mb3JtYXRpb24gb24gdGhlIGFib3ZlLCBzZWUgdGhlIFtQcm9qZWN0c10g
KGh0dHBzOi8vZ2l0aHViLmNvbS9jdGYtd2lraS9jdGYtd2lraS9wcm9qZWN0cykg
b2YtdGhlIENURiBXaWtpIGZvciBhIGRldGFpbGVkIGxpc3Qtb2Ytd2hhdCBpcyBiZWluZyBkb25lIGFuZCB3aGF0IHRvIGRvLt==
T2YgY29lcnNlLCB0aGUgQ1RGIFdpa2kgaXMgYmFzZWQgb24gQ1RGLCA=
YnV0IG10IGlzIG5vdCBsaW1pdGVkIHRvIENURi4gSW4gdGhlIGZ1dHVyZSwgQ1RGIFdpa2kgd2lsbD==
SW50cm9kdWNpbmcgdG9vbHMgaW4gc2VjdXJpdHkgcmVzZWFyY2ggYXJlYX==TW9yZSBpbnRlZ3JhdGlvbiB3aXRoIHN1Y3VyaXR5IGFyZWE=
SW4gYWRkaXRpb24sIGdpdmVuIHRoZSBmb2xsb3dpbmcgdHdvIHBvaW50c9==
VGVjaG5vbG9neSBzaG91bGQsYmUsc2hhcmVkIGluIGFuIG9wZW4sbWFubmVyLs==U2VjdXJpdHkgb2ZmZW5zaXZlIGFuZCBkZWZlbnNpdmUg
dGVjaG5vbG9naWVzIGFyZSBhbHdheXMgdXAgdG8gZGF0ZSwgYW5kIG9sZCB0ZWNobm9sb2dpZXMgbWF5IGZhaWwgYXQgYW55IHRpbWUg
aW7gdGhlIGZhY2Ugb2YgbmV3IHRlY2hub2xvZ2llcy7=Q1RGIFdpa2kmd2lsbCBuZXZlciBwdWJsaXNoIGJvb2tzLm==
RmluYWxseSwgdGhlIENURiBXaWtpIG9yaWdpbmF0ZXMgZnJvbSB0aGUgY29tbXVuaXR5LCB=
YXMgYW4gaW5kZXBlbmRlbnQgb3JnYW5pemF0aW9uLCBhZHZvY2F0ZXMgZnJlZWRvbSBvZiBrbm93bGVkZ2Us
IHdpbGwgbmV2ZXIgYmUgY29tbWVyY2lhbGl6ZWQgaW4gdGhlIGZ1dHVyZSwg
YW5kIHdpbGwoYWx3YXlzIHJlbWFpbiBpbmRlcGVuZGVudCBhbmQoZnJlZWRvbSoqLo==
```

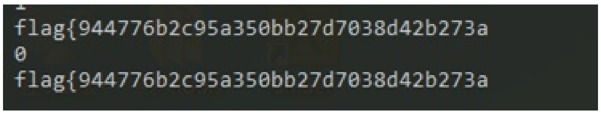
https://blog.csdn.net/Crazy198410

看到每一行后面基本都有==

怀疑是base64隐写。用脚本解密：

```
def get_base64_diff_value(s1, s2):
    base64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
    res = 0
    for i in xrange(len(s2)):
        if s1[i] != s2[i]:
            return abs(base64chars.index(s1[i]) - base64chars.index(s2[i]))
    return res


def solve_stego():
    with open('flag.txt', 'rb') as f:
        file_lines = f.readlines()
        bin_str = ''
        for line in file_lines:
            steg_line = line.replace('\n', '')
            norm_line = line.replace('\n', '').decode('base64').encode('base64').replace('\n', '')
            diff = get_base64_diff_value(steg_line, norm_line)
            print diff
            pads_num = steg_line.count('=')
            if diff:
                bin_str += bin(diff)[2:].zfill(pads_num * 2)
            else:
                bin_str += '0' * pads_num * 2
            print goflag(bin_str)


def goflag(bin_str):
    res_str = ''
    for i in xrange(0, len(bin_str), 8):
        res_str += chr(int(bin_str[i:i + 8], 2))
    return res_str


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

可以得到flag

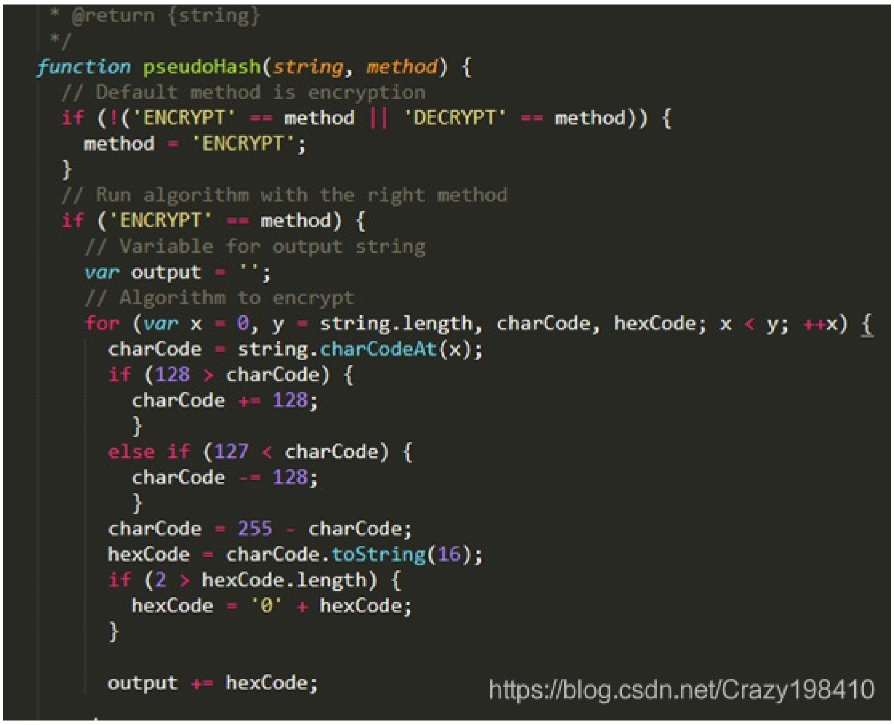


```
flag{944776b2c95a350bb27d7038d42b273a
0
flag{944776b2c95a350bb27d7038d42b273a
```

补全大括号即可

2020sdnisc-简单的js

下载附件是一个js文件
打开：



```
 * @return {string}
 */
function pseudoHash(string, method) {
  // Default method is encryption
  if (!('ENCRYPT' == method || 'DECRYPT' == method)) {
    method = 'ENCRYPT';
  }
  // Run algorithm with the right method
  if ('ENCRYPT' == method) {
    // Variable for output string
    var output = '';
    // Algorithm to encrypt
    for (var x = 0, y = string.length, charCode, hexCode; x < y; ++x) {
      charCode = string.charCodeAt(x);
      if (128 > charCode) {
        charCode += 128;
        }
      else if (127 < charCode) {
        charCode -= 128;
        }
      charCode = 255 - charCode;
      hexCode = charCode.toString(16);
      if (2 > hexCode.length) {
        hexCode = '0' + hexCode;
      }

      output += hexCode;
```

https://blog.csdn.net/Crazy198410

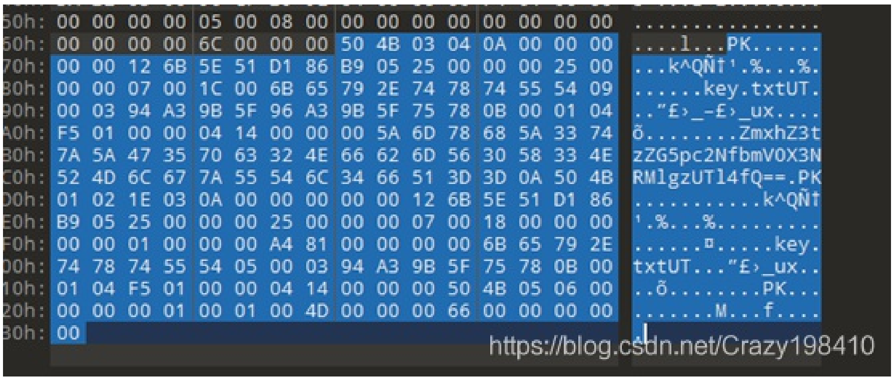
是一段代码。给了算法过程和结果。只要逆运算就可以。
写脚本：

```
s='19131e18041b1d4c47191d19194f1949481a481a1d4c1c461b4d484b191b4e474f1e4b1d4c02'
flag=''
for i in range(0,len(s),2):
    tmp = int(s[i:i+2],16)

    #print (tmp)
    flag+=chr((255-128)-tmp)
print (flag)
```

得到flag：flag{db38fbff0f67e7eb3c9d274fd180a4b3}

2020sdnisc-损坏的流量包

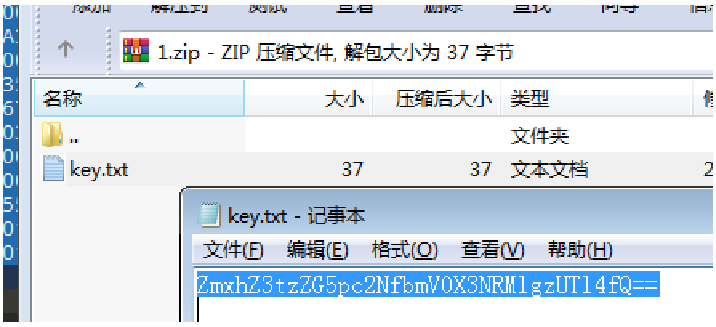
下载附件，是一个流量包，但wireshark打不开。
用010editor打开。



```
50h: 00 00 00 00 05 00 08 00 00 00 00 00 00 00 00 00  ................
60h: 00 00 00 00 6C 00 00 00 50 4B 03 04 0A 00 00 00  ....l...PK......
70h: 00 00 12 6B 5E 51 D1 86 B9 05 25 00 00 00 25 00  ...k^QÑ†¹.%...%.
80h: 00 00 07 00 1C 00 6B 65 79 2E 74 78 74 55 54 09  ......key.txtUT.
90h: 00 03 94 A3 9B 5F 96 A3 9B 5F 75 78 0B 00 01 04  ..”£›_–£›_ux....
A0h: F5 01 00 00 04 14 00 00 00 5A 6D 78 68 5A 33 74  õ........ZmxhZ3t
B0h: 7A 5A 47 35 70 63 32 4E 66 62 6D 56 30 58 33 4E  zZG5pc2NfbmV0X3N
C0h: 52 4D 6C 67 7A 55 54 6C 34 66 51 3D 3D 0A 50 4B  RMlgzUTl4fQ==.PK
D0h: 01 02 1E 03 0A 00 00 00 00 00 12 6B 5E 51 D1 86  ..........k^QÑ†
E0h: B9 05 25 00 00 00 25 00 00 00 07 00 18 00 00 00  ¹.%...%.........
F0h: 00 00 01 00 00 00 A4 81 00 00 00 00 6B 65 79 2E  ......¤.....key.
00h: 74 78 74 55 54 05 00 03 94 A3 9B 5F 75 78 0B 00  txtUT...”£›_ux..
10h: 01 04 F5 01 00 00 04 14 00 00 00 50 4B 05 06 00  ..õ........PK...
20h: 00 00 00 01 00 01 00 4D 00 00 00 66 00 00 00 00  .......M...f....
30h: 00
```

https://blog.csdn.net/Crazy198410

在文件的结尾发现压缩包，提取出来，并保存为zip文件。打开



1.zip - ZIP 压缩文件, 解包大小为 37 字节

名称	大小	压缩后大小	类型
..			文件夹
key.txt	37	37	文本文档

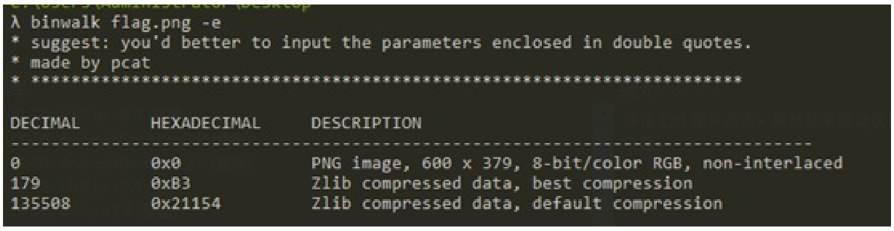
key.txt - 记事本

```
ZmxhZ3tzZG5pc2NfbmV0X3NRMlgzUTl4fQ==
```

是一个base64加密，解密后，就是flag：flag{sdnisc_net_sQ2X3Q9x}

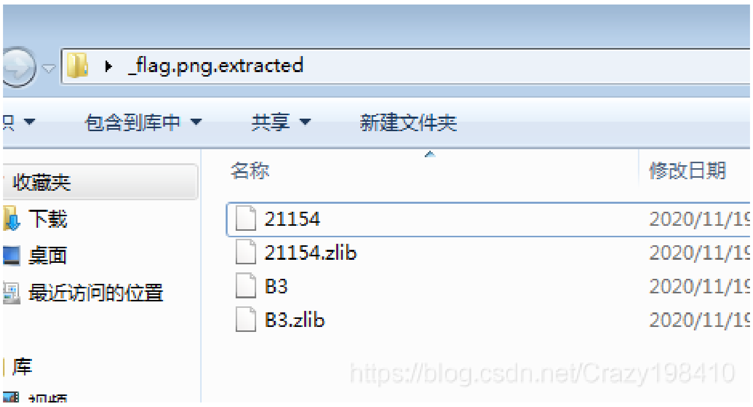
2020sdnisc-过去和现在

下载附件，是一张图片，用binwalk分析：



```
λ binwalk flag.png -e
* suggest: you'd better to input the parameters enclosed in double quotes.
* made by pcat
* ***********************************************************************

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 600 x 379, 8-bit/color RGB, non-interlaced
179           0xB3            Zlib compressed data, best compression
135508        0x21154         Zlib compressed data, default compression
```

得到若干文件，逐一打开。

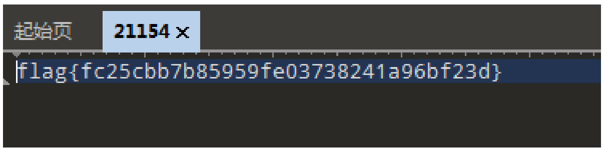


_flag.png.extracted

名称	修改日期
21154	2020/11/1
21154.zlib	2020/11/1
B3	2020/11/1
B3.zlib	2020/11/1

https://blog.csdn.net/Crazy198410

在21154中发现flag



```
flag{fc25cbb7b85959fe03738241a96bf23d}
```

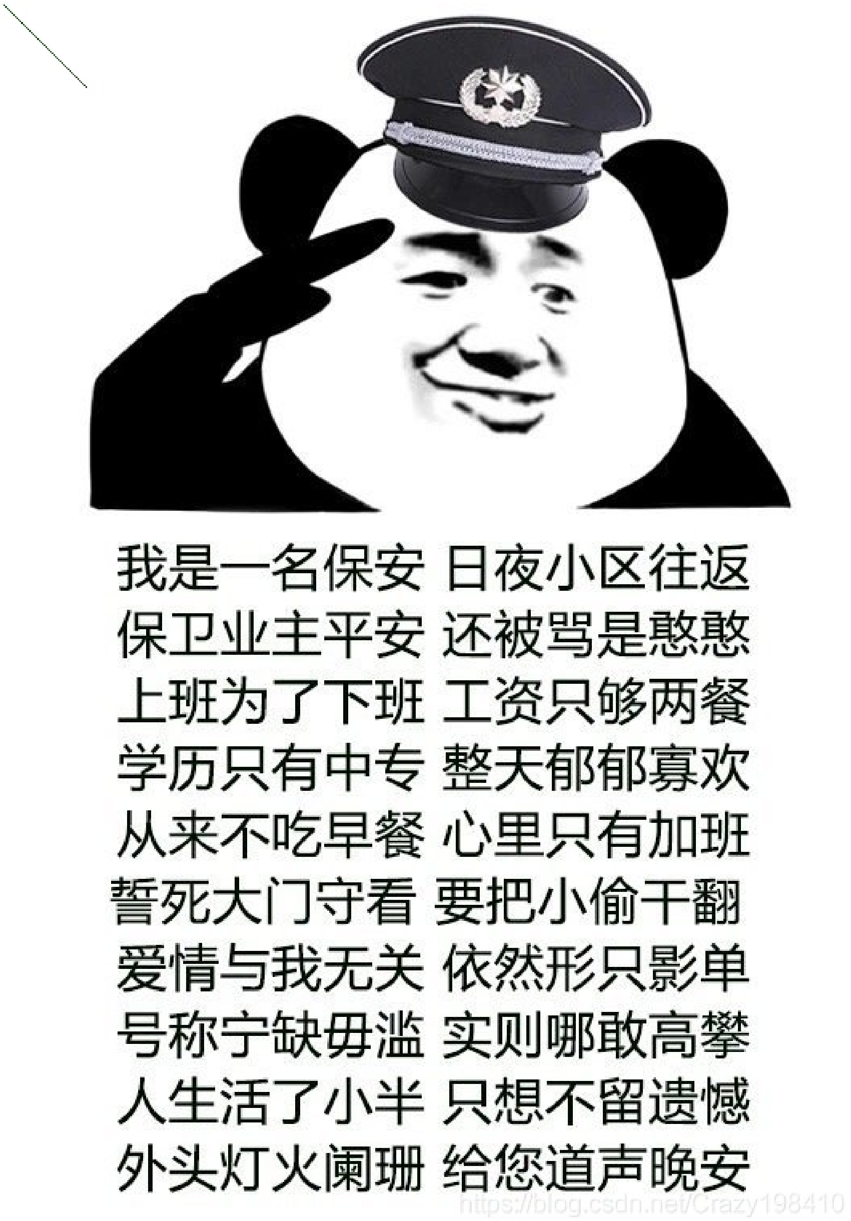
2020sdnisc-左上角的秘密

下载附件，是一个文件和一个脚本。打开脚本：

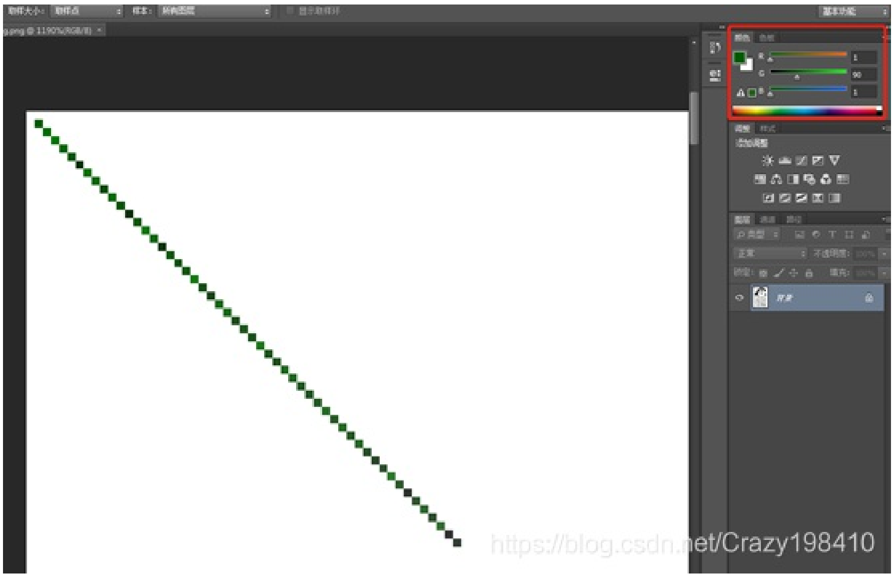
是一段代码，对图片内容进行了加密，得到附件中的文件。写脚本进行逆运算：

```
flag_dec = open("flag.png","wb")
def file_decode(flag):
    i = 1
    while True:
        byte_str = flag.read(1)
        if (byte_str == b''):
            exit()
        byte_str = hex_decode(byte_str)
        file_write(flag_dec, byte_str)
        # print(byte_str, end="")
        i = i + 1

def hex_decode(byte_str):
    tmp = int.from_bytes(byte_str, byteorder="big")
    tmp = tmp ^ 128
    if (tmp % 2 == 0):
        tmp = tmp + 1
    else:
        tmp = tmp - 1
    tmp = bytes([tmp])
    return tmp

def file_write(flag_dec, byte_str):
    flag_dec.write(byte_str)

if __name__ == '__main__':
    with open("./flag_enc.hex", "rb") as flag:
        file_decode(flag)
flag_dec.close()
```

还原出图片


我是一名保安 日夜小区往返
保卫业主平安 还被骂是憨憨
上班为了下班 工资只够两餐
学历只有中专 整天郁郁寡欢
从来不吃早餐 心里只有加班
誓死大门守看 要把小偷干翻
爱情与我无关 依然形只影单
号称宁缺毋滥 实则哪敢高攀
人生活了小半 只想不留遗憾
外头灯火阑珊 给您道声晚安
https://blog.csdn.net/Crazy198410

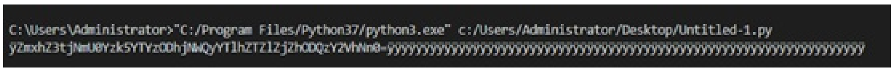
发现图片左上有条线。查看颜色：



猜测秘密在绿色中

使用脚本得到密文:

```
from PIL import Image
image =Image.open('flag.png')
c=0
aa=''
for i in range(120):
    aa+=chr(image.getpixel((c,c))[-2])
    c+=1
print (aa)
```

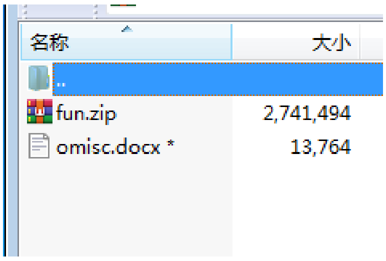


```
C:\Users\Administrator>"C:/Program Files/Python37/python3.exe" c:/Users/Administrator/Desktop/Untitled-1.py
yZmxhZ3tjNmU0Yzk5YTYzODhjNWQyYTlhZTZlZjZhODQzY2VhNn0=yyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy
```

发现其中秘密：
ZmxhZ3tjNmU0Yzk5YTYzODhjNWQyYTlhZTZlZjZhODQzY2VhNn0=
Base64解密后得到flag:
flag{c6e4c99a6388c5d2a9ae6ef6a843cea6}

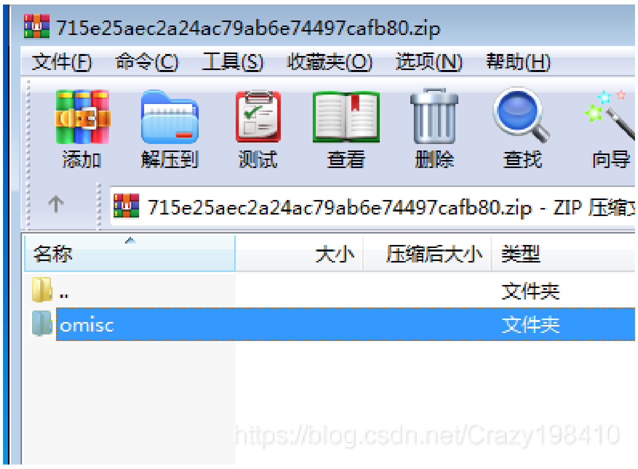
泰湖杯-MISC

下载附件，是一个压缩包，里面是一个文件和一个压缩包。



名称	大小
..	
fun.zip	2,741,494
omisc.docx *	13,764

尝试各种方法无果后，将附件中的fun.zip删除。

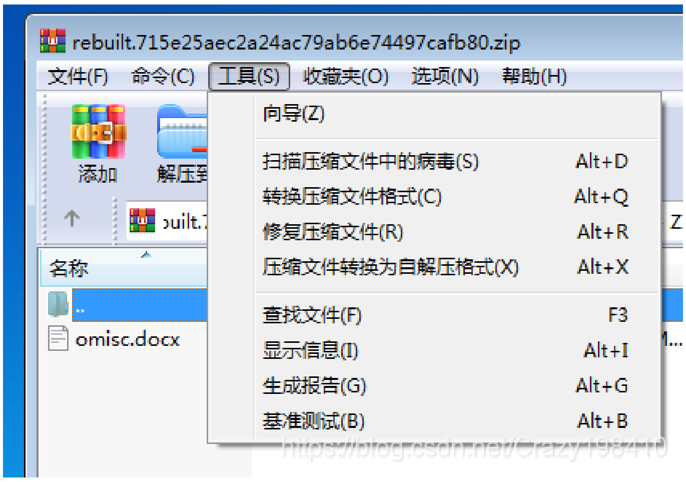


715e25aec2a24ac79ab6e74497cafb80.zip
文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)
添加
解压到
测试
查看
删除
查找
向导
715e25aec2a24ac79ab6e74497cafb80.zip - ZIP 压缩

名称	大小	压缩后大小	类型
..			文件夹
omisc			文件夹

https://blog.csdn.net/Crazy198410

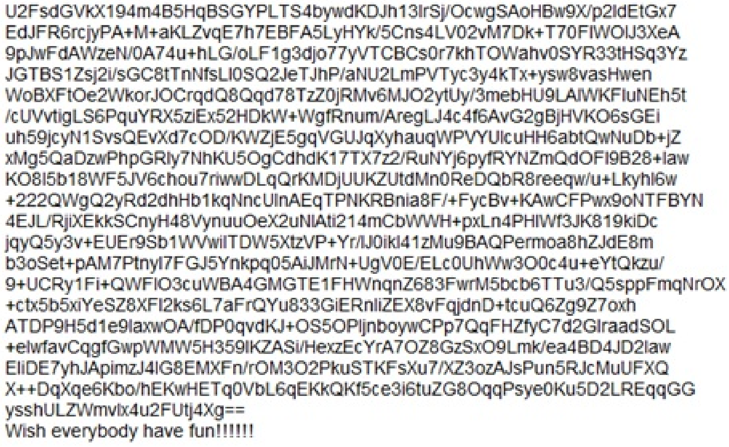
再修复压缩包，可以发现密码没有了。



rebuilt.715e25aec2a24ac79ab6e74497cafb80.zip
文件(F) 命令(C) 工具(S) 收藏夹(O) 选项(N) 帮助(H)
添加
解压到
向导(Z)
扫描压缩文件中的病毒(S) Alt+D
转换压缩文件格式(C) Alt+Q
修复压缩文件(R) Alt+R
压缩文件转换为自解压格式(X) Alt+X
查找文件(F) F3
显示信息(I) Alt+I
生成报告(G) Alt+G
基准测试(B) Alt+B
名称
..
omisc.docx

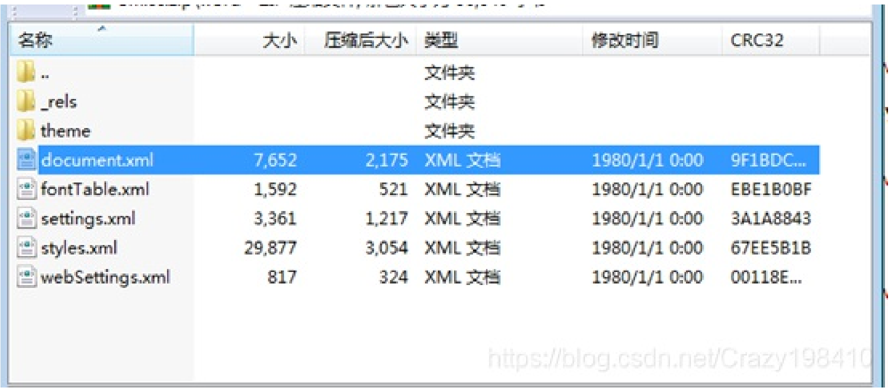
https://blog.csdn.net/Crazy198410

打开后，是一段密文



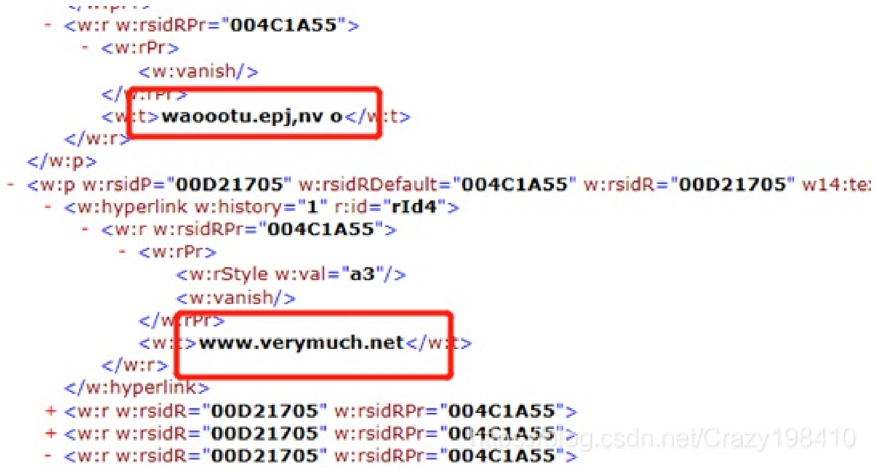
```
U2FsdGVkX194m4B5HqBSGYPLTS4bywdKDJh13IrSj/OcwgSAoHBw9X/p2IdEtGx7
EdJFR6rcjyPA+M+aKLZvqE7h7EBFA5LyHYk/5Cns4LV02vM7Dk+T70FIWOlJ3XeA
9pJwFdAWzeN/0A74u+hLG/oLF1g3djo77yVTCBCs0r7khTOWahv0SYR33tHSq3Yz
JGTBS1Zsj2i/sGC8tTnNfsLI0SQ2JeTJhP/aNU2LmPVTyc3y4kTx+ysw8vasHwen
WoBXFtOe2WkorJOCrqdQ8Qqd78TzZ0jRMv6MJO2ytUy/3mebHU9LAIWKFIuNEh5t
/cUVvtigLS6PquYRX5ziEx52HDkW+WgfRnum/AregLJ4c4f6AvG2gBjHVKO6sGEi
uh59jcyN1SvsQEvXd7cOD/KWZjE5gqVGUJqXyhauqWPVYUlcuHH6abtQwNuDb+jZ
xMg5QaDzwPhpGRly7NhKU5OgCdhdK17TX7z2/RuNYj6pyfRYNZmQdOFI9B28+law
KO8I5b18WF5JV6chou7riwwDLqQrKMDjUUKZUtdMn0ReDQbR8reeqw/u+Lkyhl6w
+222QWgQ2yRd2dhHb1kqNncUlnAEqTPNKRBnia8F/+FycBv+KAwCFPwx9oNTFBYN
4EJL/RjiXEkkSCnyH48VynuuOeX2uNlAti214mCbWWH+pxLn4PHlWf3JK819kiDc
jqyQ5y3v+EUEr9Sb1WVwilTDW5XtzVP+Yr/lJ0ikl41zMu9BAQPermoa8hZJdE8m
b3oSet+pAM7PtnyI7FGJ5Ynkpq05AiJMrN+UgV0E/ELc0UhWw3O0c4u+eYtQkzu/
9+UCRy1Fi+QWFIO3cuWBA4GMGTE1FHWnqnZ683FwrM5bcb6TTu3/Q5sppFmqNrOX
+ctx5b5xiYeSZ8XFI2ks6L7aFrQYu833GiERnliZEX8vFqjdnD+tcuQ6Zg9Z7oxh
ATDP9H5d1e9laxwOA/fDP0qvdKJ+OS5OPljnboywCPp7QqFHZfyC7d2GIraadSOL
+elwfavCqgfGwpWMW5H359IKZASi/HexzEcYrA7OZ8GzSxO9Lmk/ea4BD4JD2law
EliDE7yhJApimzJ4IG8EMXFn/rOM3O2PkuSTKFsXu7/XZ3ozAJsPun5RJcMuUFXQ
X++DqXqe6Kbo/hEKwHETq0VbL6qEKkQKf5ce3i6tuZG8OqqPsye0Ku5D2LREqqGG
ysshULZWmvlx4u2FUtj4Xg==
Wish everybody have fun!!!!!!
```

改后缀。并打开：



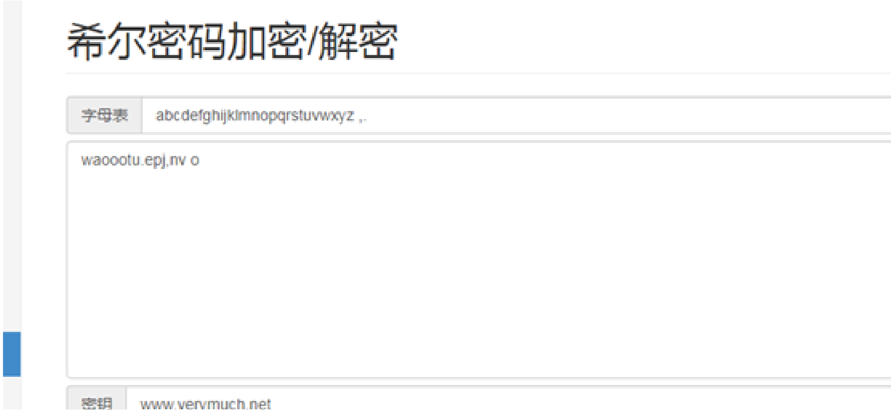
名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
_rels			文件夹		
theme			文件夹		
document.xml	7,652	2,175	XML 文档	1980/1/1 0:00	9F1BDC...
fontTable.xml	1,592	521	XML 文档	1980/1/1 0:00	EBE1B0BF
settings.xml	3,361	1,217	XML 文档	1980/1/1 0:00	3A1A8843
styles.xml	29,877	3,054	XML 文档	1980/1/1 0:00	67EE5B1B
webSettings.xml	817	324	XML 文档	1980/1/1 0:00	00118E...

发现多出现段文字：



```
  - <w:r w:rsidRPr="004C1A55">
    - <w:rPr>
        <w:vanish/>
      </w:rPr>
      <w:t>waoootu.epj,nv o</w:t>
    </w:r>
  </w:p>
- <w:p w:rsidP="00D21705" w:rsidRDefault="004C1A55" w:rsidR="00D21705" w14:te
  - <w:hyperlink w:history="1" r:id="rId4">
    - <w:r w:rsidRPr="004C1A55">
      - <w:rPr>
          <w:rStyle w:val="a3"/>
          <w:vanish/>
        </w:rPr>
        <w:t>www.verymuch.net</w:t>
      </w:r>
    </w:hyperlink>
  + <w:r w:rsidR="00D21705" w:rsidRPr="004C1A55">
  + <w:r w:rsidR="00D21705" w:rsidRPr="004C1A55">
  - <w:r w:rsidR="00D21705" w:rsidRPr="004C1A55">
```

测试各种密码，发现希尔密码能解新发现的两句话：

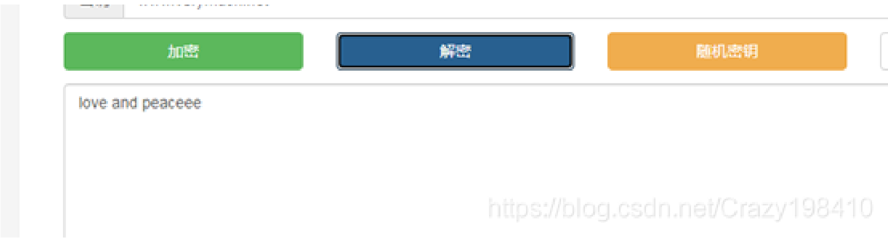


希尔密码加密/解密

字母表 abcdefghijklmnopqrstuvwxyz ,.

waoootu.epj,nv o

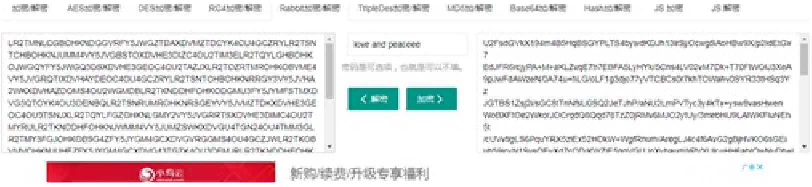
密钥 www.verymuch.net



love and peaceee

得到一句新的字符串：
love and peaceee

用这句话做密码，使用rabbit解密剩余的一大段话：

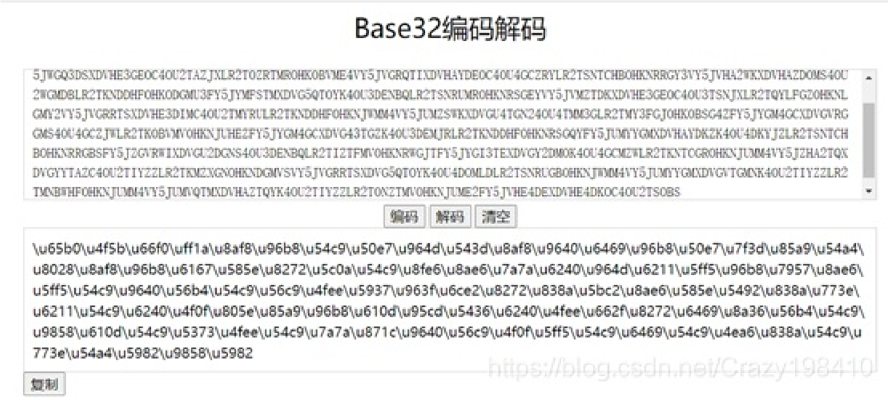


love and peaceee

得到：

```
LR2TMNLCGBOHKNDGGVRFY5JWGZTDAXDVMZTDCYK4OU4GCZRYLR2TSNTCHBOHKNJUMM4VY5JVGBSTOXDVHE3DIZC4OU2TIM3ELR2TQYLGHBOHKOJW
GQYFY5JWGQ3DSXDVHE3GEOC4OU2TAZJXLR2TOZRTMROHKOBVME4VY5JVGRQTIXDVHAYDEOC4OU4GCZRYLR2TSNTCHBOHKNRRGY3VY5JVHA2WKXDV
HAZDOMS4OU2WGMDBLR2TKNDDHFOHKODGMU3FY5JYMFSTMXDVG5QTOYK4OU3DENBQLR2TSNRUMROHKNRSGEYVY5JVMZTDKXDVHE3GEOC4OU3TSNJX
LR2TQYLFGZOHKNLGMY2VY5JVGRRTSXDVHE3DIMC4OU2TMYRULR2TKNDDHFOHKNJWMM4VY5JUMZSWKXDVGU4TGN24OU4TMM3GLR2TMY3FGJOHKOBS
G4ZFY5JYGM4GCXDVGVRGGMS4OU4GCZJWLR2TKOBVMVOHKNJUHEZFY5JYGM4GCXDVG43TGZK4OU3DEMJRLR2TKNDDHFOHKNRSGQYFY5JUMYYGMXDV
HAYDKZK4OU4DKYJZLR2TSNTCHBOHKNRRGBSFY5JZGVRWIXDVGU2DGNS4OU3DENBQLR2TIZTFMVOHKNRWGJTFY5JYGI3TEXDVGY2DMOK4OU4GCMZW
LR2TKNTCGROHKNJUMM4VY5JZHA2TQXDVGYYTAZC4OU2TIYZZLR2TKMZXGNOHKNDGMVSVY5JVGRRTSXDVG5QTOYK4OU4DOMLDLR2TSNRUGBOHKNJW
MM4VY5JUMYYGMXDVGVTGMNK4OU2TIYZZLR2TMNBWHFOHKNJUMM4VY5JUMVQTMXDVHAZTQYK4OU2TIYZZLR2TONZTMVOHKNJUME2FY5JVHE4DEXDV
HE4DKOC4OU2TSOBS
```

再进行base32的解密：

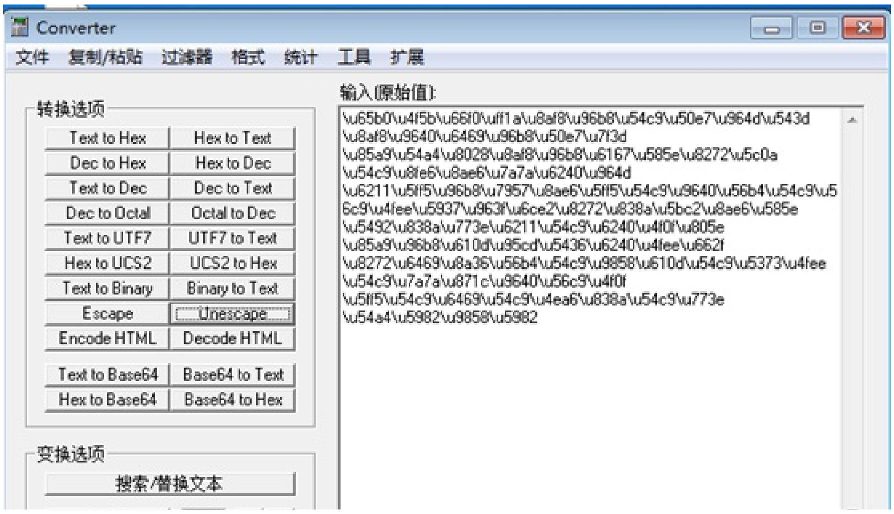


Base32编码解码

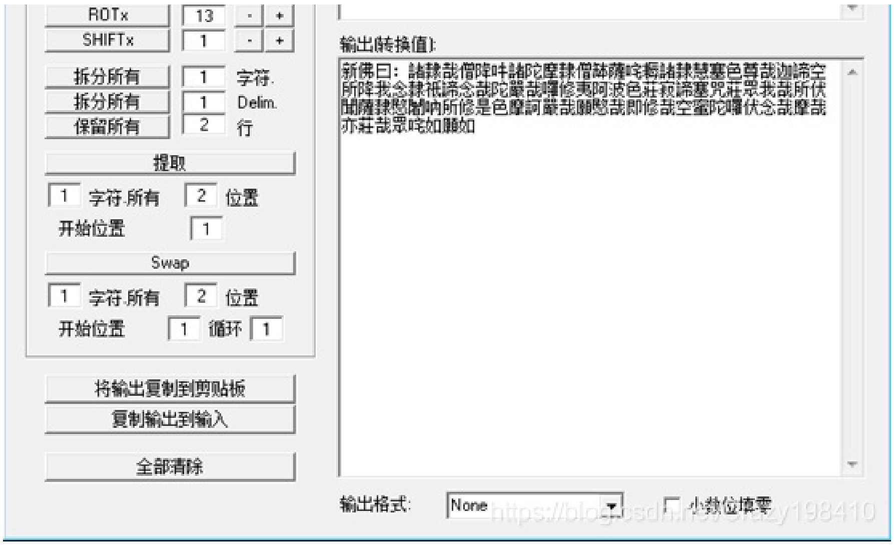
```
5JWGQ3DSXDVHE3GEOC4OU2TAZJXLR2TOZRTMROHKOBVME4VY5JVGRQTIXDVHAYDEOC4OU4GCZRYLR2TSNTCHBOHKNRRGY3VY5JVHA2WKXDVHAZDOMS4OU
2WGMDBLR2TKNDDHFOHKODGMU3FY5JYMFSTMXDVG5QTOYK4OU3DENBQLR2TSNRUMROHKNRSGEYVY5JVMZTDKXDVHE3GEOC4OU3TSNJXLR2TQYLFGZOHKNL
GMY2VY5JVGRRTSXDVHE3DIMC4OU2TMYRULR2TKNDDHFOHKNJWMM4VY5JUMZSWKXDVGU4TGN24OU4TMM3GLR2TMY3FGJOHKOBSG4ZFY5JYGM4GCXDVGVRG
GMS4OU4GCZJWLR2TKOBVMVOHKNJUHE2FY5JYGM4GCXDVG43TGZK4OU3DEMJRLR2TKNDDHFOHKNRSGQYFY5JUMYYGMXDVHAYDKZK4OU4DKYJZLR2TSNTCH
BOHKNRRGBSFY5JZGVRWIXDVGU2DGNS4OU3DENBQLR2TIZTFMVOHKNRWGJTFY5JYGI3TEXDVGY2DMOK4OU4GCMZWLR2TKNTCGROHKNJUMM4VY5JZHA2TQX
DVGYYTAZC4OU2TIYZZLR2TKMZXGNOHKNDGMVSVY5JVGRRTSXDVG5QTOYK4OU4DOMLDLR2TSNRUGBOHKNJWMM4VY5JUMYYGMXDVGVTGMNK4OU2TIYZZLR2
TMNBWHFOHKNJUMM4VY5JUMVQTMXDVHAZTQYK4OU2TIYZZLR2TONZTMVOHKNJUME2FY5JVHE4DEXDVHE4DKOC4OU2TSOBS
```

```
\u65b0\u4f5b\u66f0\uff1a\u8af8\u96b8\u54c9\u50e7\u964d\u543d\u8af8\u9640\u6469\u96b8\u50e7\u7f3d\u85a9\u54a4\
u8028\u8af8\u96b8\u6167\u585e\u8272\u5c0a\u54c9\u8fe6\u8ae6\u7a7a\u6240\u964d\u6211\u5ff5\u96b8\u7957\u8ae6\
u5ff5\u54c9\u9640\u56b4\u54c9\u56c9\u4fee\u5937\u963f\u6ce2\u8272\u838a\u5bc2\u8ae6\u585e\u5492\u838a\u773e\
u6211\u54c9\u6240\u4f0f\u805e\u85a9\u96b8\u610d\u95cd\u5436\u6240\u4fee\u662f\u8272\u6469\u8a36\u56b4\u54c9\
u9858\u610d\u54c9\u5373\u4fee\u54c9\u7a7a\u871c\u9640\u56c9\u4f0f\u5ff5\u54c9\u6469\u54c9\u4ea6\u838a\u54c9\u
773e\u54a4\u5982\u9858\u5982
```

再Unicode解码：



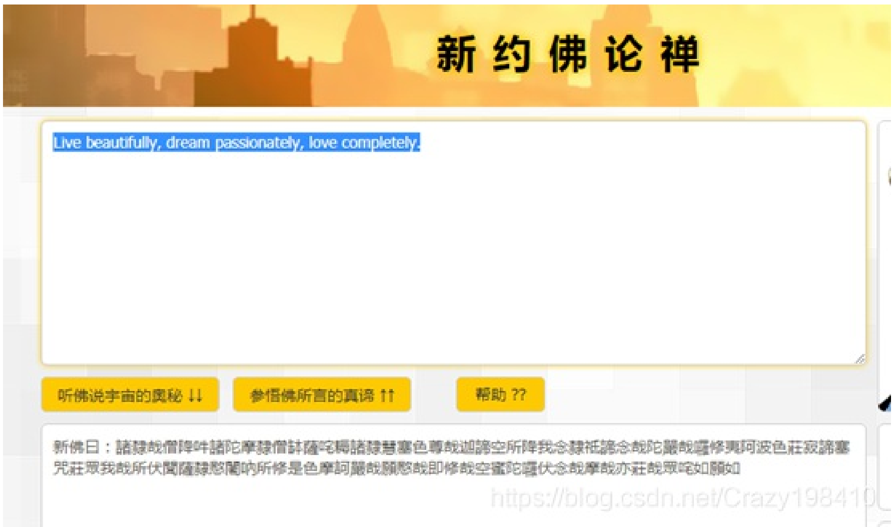
```
\u65b0\u4f5b\u66f0\uff1a\u8af8\u96b8\u54c9\u50e7\u964d\u543d
\u8af8\u9640\u6469\u96b8\u50e7\u7f3d
\u85a9\u54a4\u8028\u8af8\u96b8\u6167\u585e\u8272\u5c0a
\u54c9\u8fe6\u8ae6\u7a7a\u6240\u964d
\u6211\u5ff5\u96b8\u7957\u8ae6\u5ff5\u54c9\u9640\u56b4\u54c9\u5
6c9\u4fee\u5937\u963f\u6ce2\u8272\u838a\u5bc2\u8ae6\u585e
\u5492\u838a\u773e\u6211\u54c9\u6240\u4f0f\u805e
\u85a9\u96b8\u610d\u95cd\u5436\u6240\u4fee\u662f
\u8272\u6469\u8a36\u56b4\u54c9\u9858\u610d\u54c9\u5373\u4fee
\u54c9\u7a7a\u871c\u9640\u56c9\u4f0f
\u5ff5\u54c9\u6469\u54c9\u4ea6\u838a\u54c9\u773e
\u54a4\u5982\u9858\u5982
```



输出(转换值):

新佛曰：諸隸哉僧降吽諸陀摩隸僧缽薩吽羯諸隸慧塞色尊哉迦諦空所降我念隸祇諦念哉陀嚴哉囉修夷阿波色莊寂諦塞咒莊眾我哉所伏闍薩隸愍闍吶所修是色摩訶嚴哉願愍哉即修哉空蜜陀囉伏念哉摩哉亦莊哉眾吒如願如

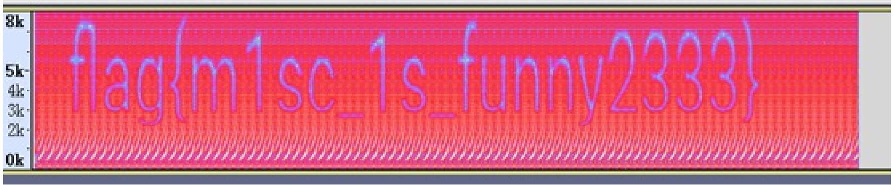
再新佛曰解码：



Live beautifully, dream passionately, love completely.

新佛曰：諸隸哉僧降吽諸陀摩隸僧缽薩吽羯諸隸慧塞色尊哉迦諦空所降我念隸祇諦念哉陀嚴哉囉修夷阿波色莊寂諦塞咒莊眾我哉所伏闍薩隸愍闍吶所修是色摩訶嚴哉願愍哉即修哉空蜜陀囉伏念哉摩哉亦莊哉眾吒如願如

得到一段话，用这然话解压fun.zip
得到音频文件。
再分析频谱，即可得flag

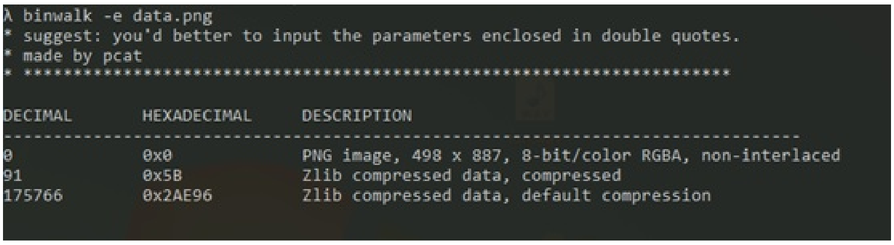


可乐加冰

下载附件，是一张图片：

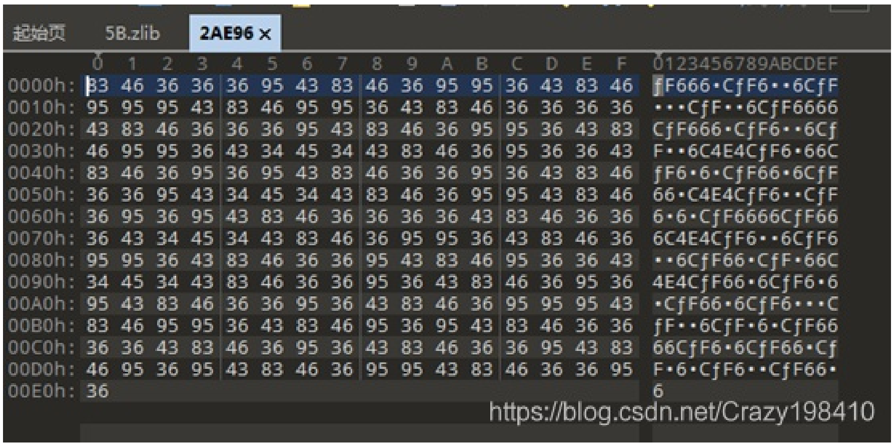
pepsi®
https://blog.csdn.net/Crazy198410

用binwalk分析：



```
λ binwalk -e data.png
* suggest: you'd better to input the parameters enclosed in double quotes.
* made by pcat
* ***********************************************************************

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 498 x 887, 8-bit/color RGBA, non-interlaced
91            0x5B            Zlib compressed data, compressed
175766        0x2AE96         Zlib compressed data, default compression
```

得到若干文件。
逐一查看，在2AE96文件中发现规律字符：



```
       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: 83 46 36 36 36 95 43 83 46 36 95 95 36 43 83 46  ƒF666•CƒF6••6CƒF
0010h: 95 95 95 43 83 46 95 95 36 43 83 46 36 36 36 36  •••CƒF••6CƒF6666
0020h: 43 83 46 36 36 36 95 43 83 46 36 95 95 36 43 83  CƒF666•CƒF6••6Cƒ
0030h: 46 95 95 36 43 34 45 34 43 83 46 36 95 36 36 43  F••6C4E4CƒF6•66C
0040h: 83 46 36 95 36 95 43 83 46 36 36 95 36 43 83 46  ƒF6•6•CƒF66•6CƒF
0050h: 36 36 95 43 34 45 34 43 83 46 36 95 95 43 83 46  66•C4E4CƒF6••CƒF
0060h: 36 95 36 95 43 83 46 36 36 36 36 43 83 46 36 36  6•6•CƒF6666CƒF66
0070h: 36 43 34 45 34 43 83 46 36 95 95 36 43 83 46 36  6C4E4CƒF6••6CƒF6
0080h: 95 95 36 43 83 46 36 36 95 43 83 46 95 36 36 43  ••6CƒF66•CƒF•66C
0090h: 34 45 34 43 83 46 36 36 95 36 43 83 46 36 95 36  4E4CƒF66•6CƒF6•6
00A0h: 95 43 83 46 36 36 95 36 43 83 46 36 95 95 95 43  •CƒF66•6CƒF6•••C
00B0h: 83 46 95 95 36 43 83 46 95 36 95 43 83 46 36 36  ƒF••6CƒF•6•CƒF66
00C0h: 36 36 43 83 46 36 95 36 43 83 46 36 36 95 43 83  66CƒF6•6CƒF66•Cƒ
00D0h: 46 95 36 95 43 83 46 36 95 95 43 83 46 36 36 95  F•6•CƒF6••CƒF66•
00E0h: 36                                               6
```

https://blog.csdn.net/Crazy198410

对其转为字符串:

```
s='8346363636954383463695953643834695959543834695953643834636363636438346363636954383463695953643834695953643344
5344383463695363643834636953695438346363695364383463636954334453443834636959543834636953695438346363636364383463
6363643344534438346369595364383463695953643834636369543834695363643344534438346363695364383463695369543834636369
5364383463695959543834695953643834695369543834636363636438346369536438346363695438346953695438346369595438346363
69536'
print (s)
for i in range(0,len(s),2):
    print (chr(int(s[i:i+2])),end='')
```

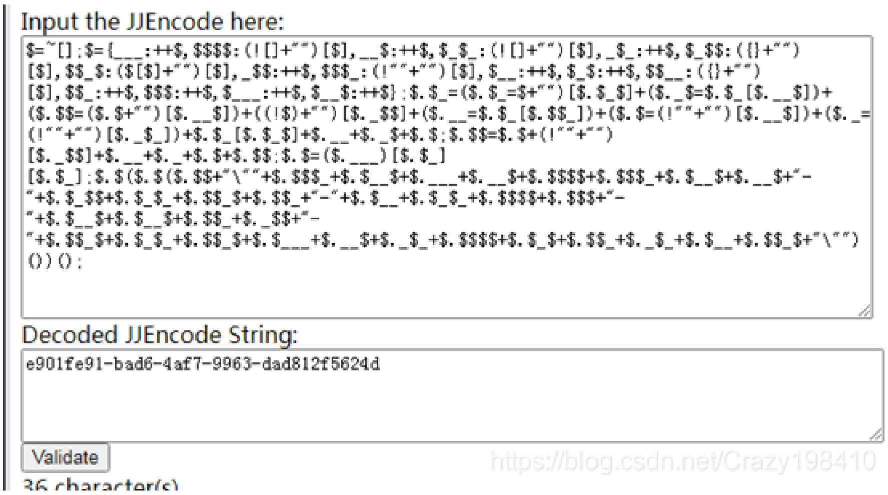
得到：

```
S.$$$_+S.$__$+S.___+S.__$+S.$$$$+S.$$$_+S.$__$+S.__$+"-"+S.$_$$+S.$_$_+S.$$_$+S.$$_+"-"+S.$__+S.$_$_+S.$$$$+S.$$
$+"-"+S.$__$+S.$__$+S.$$_+S._$$+"-"+S.$$_$+S.$_$_+S.$$_$+S.$___+S.__$+S._$_+S.$$$$+S.$_$+S.$$_+S._$_+S.$__+S.$$_
$
```

很像jjcode。将”S”替换为”$”
再加上固定的开关和结尾：

```
$=~[];$={___:++$,$$$$:(![]+"")[$],__$:++$,$_$_:(![]+"")[$],_$_:++$,$_$$:({}+"")[$],$$_$:($[$]+"")[$],_$$:++$,$$$
_:(!""+"")[$],$__:++$,$_$:++$,$$__:({}+"")[$],$$_:++$,$$$:++$,$___:++$,$__$:++$};$.$_=($.$_=$+"")[$.$_$]+($._$=$
.$_[$.__$])+($.$$=($.$+"")[$.__$])+((!$)+"")[$._$$]+($.__=$.$_[$.$$_])+($.$=(!""+"")[$.__$])+($._=(!""+"")[$._$_
])+$.$_[$.$_$]+$.__+$._$+$.$;$.$$=$.$+(!""+"")[$._$$]+$.__+$._+$.$+$.$$;$.$=($.___)[$.$_][$.$_];$.$($.$($.$$+"\"
"+这里放密文+"\"")())();
```

再用jjcode解码：



Input the JJEncode here:

```
$=~[];$={___:++$,$$$$:(![]+"")[$],__$:++$,$_$_:(![]+"")[$],_$_:++$,$_$$:({}+"")
[$],$$_$:($[$]+"")[$],_$$:++$,$$$_:(!""+"")[$],$__:++$,$_$:++$,$$__:({}+"")
[$],$$_:++$,$$$:++$,$___:++$,$__$:++$};$.$_=($.$_=$+"")[$.$_$]+($._$=$.$_[$.__$])+
($.$$=($.$+"")[$.__$])+((!$)+"")[$._$$]+($.__=$.$_[$.$$_])+($.$=(!""+"")[$.__$])+($._=
(!""+"")[$._$_])+$.$_[$.$_$]+$.__+$._$+$.$;$.$$=$.$+(!""+"")
[$._$$]+$.__+$._+$.$+$.$$;$.$=($.___)[$.$_]
[$.$_];$.$($.$($.$$+"\""+$.$$$_+$.$__$+$.___+$.__$+$.$$$$+$.$$$_+$.$__$+$.__$+"-
"+$.$_$$+$.$_$_+$.$$_$+$.$$_+"-"+$.$__+$.$_$_+$.$$$$+$.$$$+"-
"+$.$__$+$.$__$+$.$$_+$._$$+"-
"+$.$$_$+$.$_$_+$.$$_$+$.$___+$.__$+$._$_+$.$$$$+$.$_$+$.$$_+$._$_+$.$__+$.$$_$+"\"")
())();
```

Decoded JJEncode String:

```
e901fe91-bad6-4af7-9963-dad812f5624d
```

36 character(s)

https://blog.csdn.net/Crazy198410

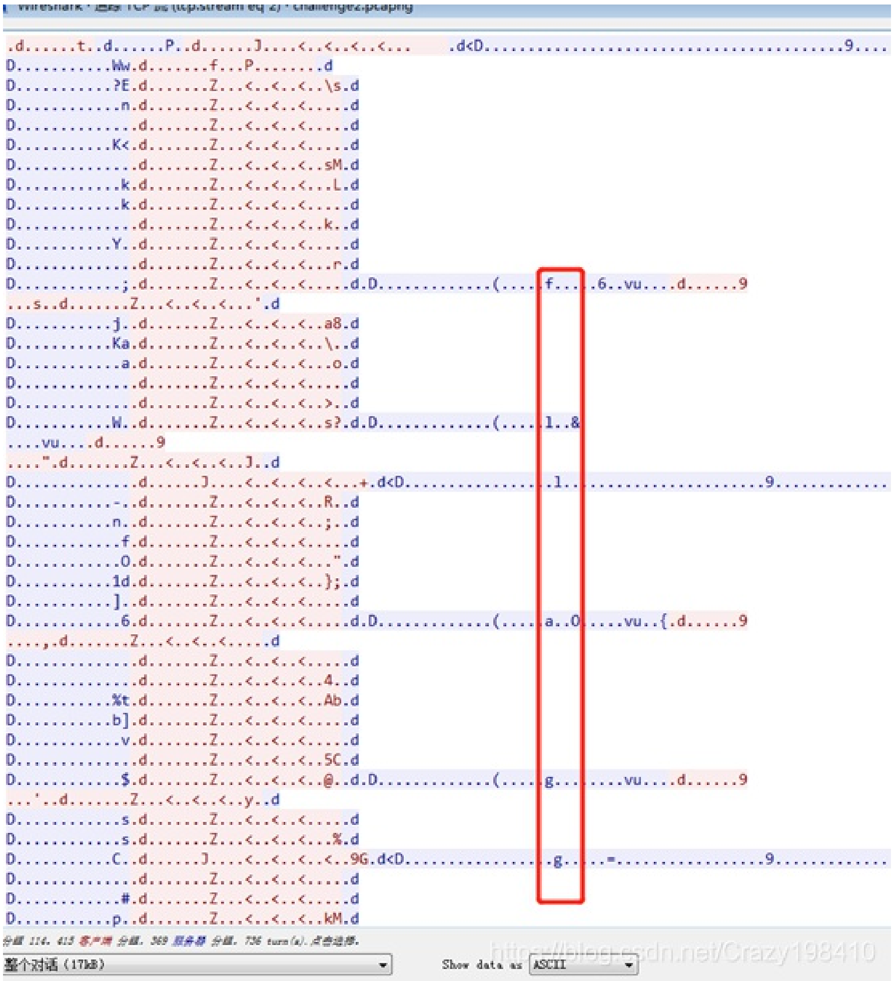
既得flag

pcap

下载附件，是一个流量包。用wireshark打开：

并追踪tcp流：

可以看到其中一个流如下图：



Wireshark · 追踪 TCP 流 (tcp.stream eq 2) · challenge2.pcapng

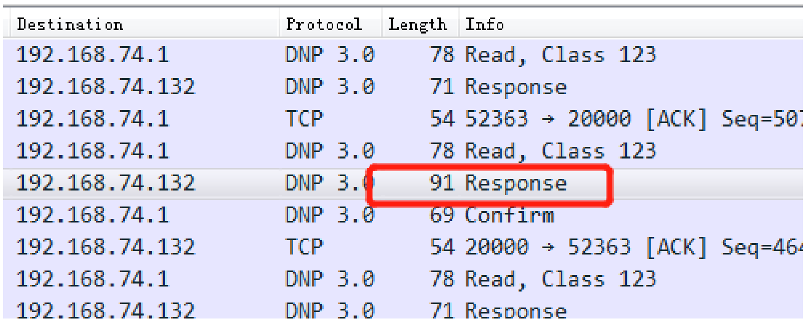
```
.d......t..d......P..d......J....<..<..<..<...     .d<D.......................................9....
D...........Ww.d.......f...P........d
D...........?E.d.......Z...<..<..<..\s.d
D............n.d.......Z...<..<..<.....d
D..............d.......Z...<..<..<.....d
D...........K<.d.......Z...<..<..<.....d
D..............d.......Z...<..<..<..sM.d
D............k.d.......Z...<..<..<...L.d
D............k.d.......Z...<..<..<.....d
D..............d.......Z...<..<..<..k..d
D...........Y..d.......Z...<..<..<.....d
D..............d.......Z...<..<..<...r.d
D............;.d.......Z...<..<..<.....d.D.............(....f.....6..vu....d......9
...s..d.......Z...<..<..<...'.d
D...........j..d.......Z...<..<..<..a8.d
D...........Ka.d.......Z...<..<..<..\..d
D............a.d.......Z...<..<..<...o.d
D..............d.......Z...<..<..<.....d
D..............d.......Z...<..<..<..>..d
D...........W..d.......Z...<..<..<..s?.d.D.............(....1..&
....vu....d......9
....".d.......Z...<..<..<..J..d
D..............d......J....<..<..<..<...+.d<D................1.......................9............
D...........-..d.......Z...<..<..<..R..d
D...........n..d.......Z...<..<..<..;..d
D............f.d.......Z...<..<..<.....d
D............O.d.......Z...<..<..<...".d
D...........1d.d.......Z...<..<..<..};.d
D...........]..d.......Z...<..<..<.....d
D............6.d.......Z...<..<..<.....d.D.............(....a..O.....vu..{.d......9
....,.d.......Z...<..<..<.....d
D..............d.......Z...<..<..<.....d
D..............d.......Z...<..<..<..4..d
D...........%t.d.......Z...<..<..<..Ab.d
D...........b].d.......Z...<..<..<.....d
D............v.d.......Z...<..<..<.....d
D..............d.......Z...<..<..<..5C.d
D............$.d.......Z...<..<..<..@..d.D.............(....g.........vu....d......9
...'..d.......Z...<..<..<..y..d
D............s.d.......Z...<..<..<.....d
D............s.d.......Z...<..<..<...%.d
D...........C..d......J....<..<..<..<..9G.d<D................g.....=.................9............
D..............d.......Z...<..<..<.....d
D............#.d.......Z...<..<..<.....d
D...........p..d.......Z...<..<..<..kM.d
```

分组 114. 415 客户端 分组, 369 服务器 分组, 736 turn(s). 点击选择.

整个对话 (17kB)　　Show data as ASCII

https://blog.csdn.net/Crazy198410

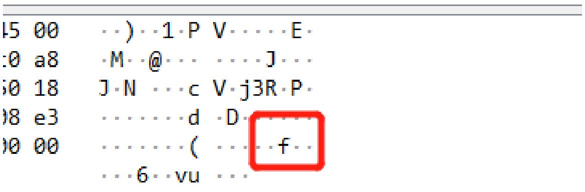
可以看到flag。对流量观察规律：



Destination	Protocol	Length	Info
192.168.74.1	DNP 3.0	78	Read, Class 123
192.168.74.132	DNP 3.0	71	Response
192.168.74.1	TCP	54	52363 → 20000 [ACK] Seq=50
192.168.74.1	DNP 3.0	78	Read, Class 123
192.168.74.132	DNP 3.0	91	Response
192.168.74.1	DNP 3.0	69	Confirm
192.168.74.132	TCP	54	20000 → 52363 [ACK] Seq=46
192.168.74.1	DNP 3.0	78	Read, Class 123
192.168.74.132	DNP 3.0	71	Response

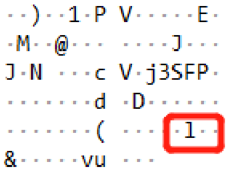
发现含有flag信息的流长度都是91。按长度排序：

逐流进行拼接：



```
45 00   ..)..1.P V.....E.
c0 a8   .M..@...  ....J...
50 18   J.N  ...c V.j3R.P.
08 e3   .......d  .D......
00 00   .......(  .....f..
        ...6..vu  ...
```

https://blog.csdn.net/Crazy198410

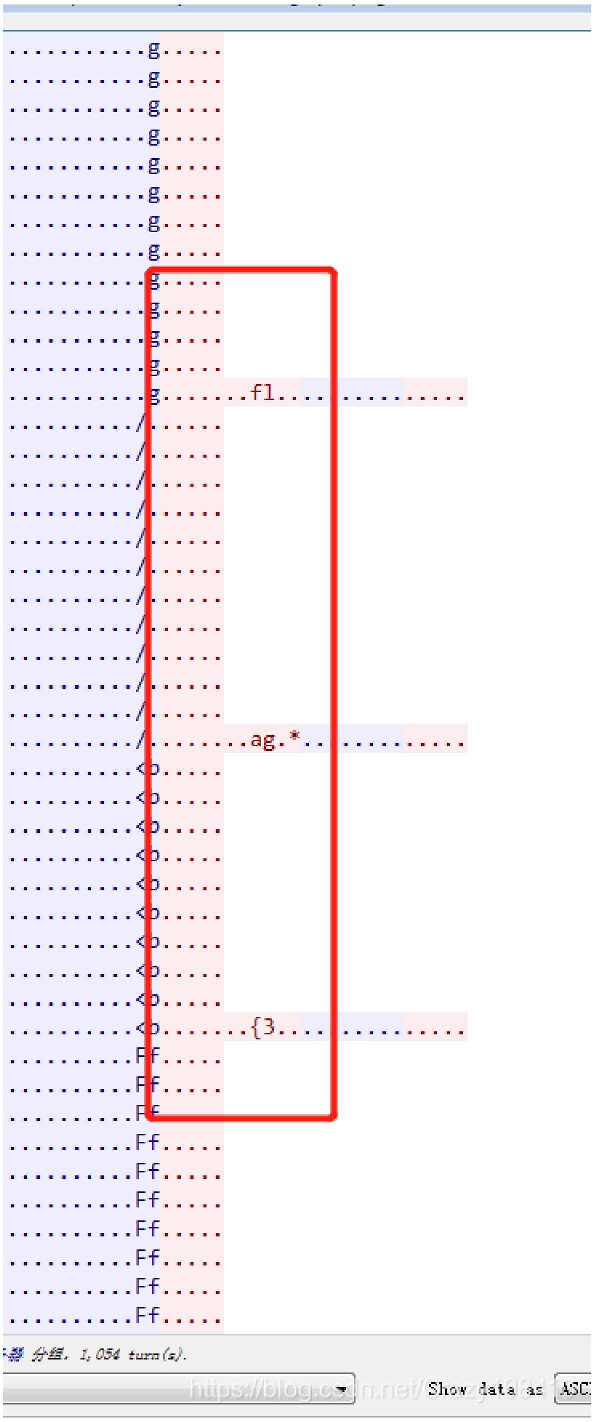


```
··)··1·P V·····E·
·M··@···  ····J···
J·N  ···c V·j3SFP·
·······d ·D······
·······( ····l··
&·····vu ···
```

可以得到flag:
flag{d989e2b92ea671f5d30efb8956eab1427625c}

pcap_analysis

下载附件，为一个流量包。
用wireshark打开。并追踪tcp流
在其中一个流中发现如下图：



```
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.....
...........g.......fl.... ..........
........../......
........../......
........../......
........../......
........../......
........../......
........../......
........../......
........../......
........../......
........../......
........../.......ag.*.. ..........
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.....
..........<p.......{3.... ..........
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
..........Ff.....
```

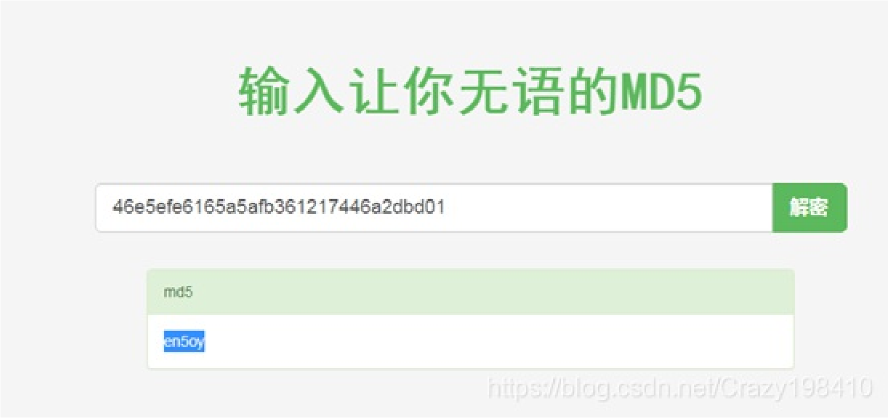
分组, 1,054 turn(s).

https://blog.csdn.net/

拼接得到flag: flag{323f986d429a689d3b96ad12dc5cbc701db0af55}

网鼎杯 2020 boom

下载附件，为一个exe文件，打开。
先是给了一段Md5码，在线进行爆破：



输入让你无语的MD5

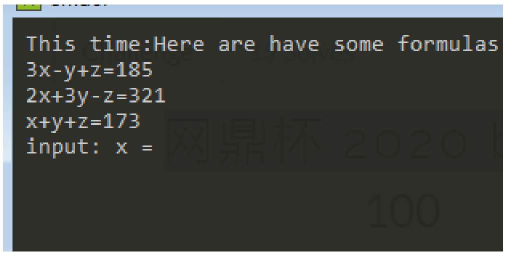
46e5efe6165a5afb361217446a2dbd01

md5

en5oy

https://blog.csdn.net/Crazy198410

得到通关密码：en5oy
第二关是是一个三元一次方程

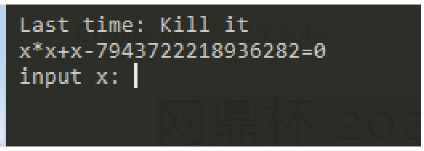


```
This time:Here are have some formulas
3x-y+z=185
2x+3y-z=321
x+y+z=173
input: x =
```

进行爆破：

```
for x in range(100):
    for y in range (100):
        for z in range(100):
            if (3*x-y+z==185)&(2*x+3*y-z==321)&(x+y+z==173):
                print (x,y,z)
```

得到解：
74
68
31
第三关是一个一元二次方程



```
Last time: Kill it
x*x+x-7943722218936282=0
input x:
```

一样爆破：

```
for x in range(1000000000):
    if(x*x+x==7943722218936282):
        print (x)
        break
```

得到解：89127561
最后得到flag：flag{en5oy_746831_89127561}