bugku ——杂项

原创

Captain Hammer 于 2019-10-14 19:00:47 发布
分类专栏： CTF题解

Table of Contents

24,好多压缩包

25，一个普通的压缩包

26， 2B

27， 好多压缩包

图片类型题 隐藏flag的几种方法：
1， 在图片的属性栏中
2，在图片的16 进制编码中 会有提示（可能直接给出flag，可能 给出一段密文，MD5，16进制码，unicoda码 等）
3，如果是 Exif 图片 用MagicExif工具查看一下这个图片的Exif信息 可能隐藏在其中
4，隐写 ，给出的压缩图片不完整， 导致隐藏了一部分图片内容，解决办法：将图片托入winHEX 中调整图片的宽和高
5，LSB图像隐藏，就用Stegsolve查看一下
6，最多的就是 将几个压缩包隐藏在图片中 这时就需要 进行 文件的分离：具体如 1 啊哒 题操作
解压缩包有时还会需要密码，再在图片中寻找密码

1，啊哒

（1）用binwalk 打开 ada.jpg 发现里面隐藏着一个 flag.txt 压缩文档：
然后 用dd 命分离出flag.txt的压缩包
解压 unzip
用vim 打开 flag.txt
![在这里插入图片描述](https://img-blog.csdnimg.cn/2019042615470137.png?x-oss-process=image/watermark,type_ZmFuZ3poZW5naGVpdGk,shadow_10,text_aHR0cHM6Ly9ibG9nLmNzZG4ub
2 pdf.pdf 文件，暗示：图片下面隐藏着信息
思路：将pdf 文档转为word 文档

5多种该方法解决

题目提示会在解题过程中得到一个二维码，顺着这个思路往下做，将得到的KEY.EXE的后缀改为txt 打开还想是一堆base64密文，把密文还原成图片，得到一张二维码，用手机扫后得到flag

6 Linux ：

（1）又是一个没有文件拓展名的文件，提示是Linux 基础，那放入Linux中 用grep 命令查找 key ：

得到 flag

（2）先用binwalk 分析一下，发现是ext3 文件，把后缀改为ext3 打开，再结尾处发现 flag（文件太大了，不适合）

7，爆照

（1）下载下来是一张图片，查看属性信息没发现什么

（2）用foremost 分离出一堆没有后缀名的文件

（3）放进winhex 中发现

文件头为 FFD8FFE0

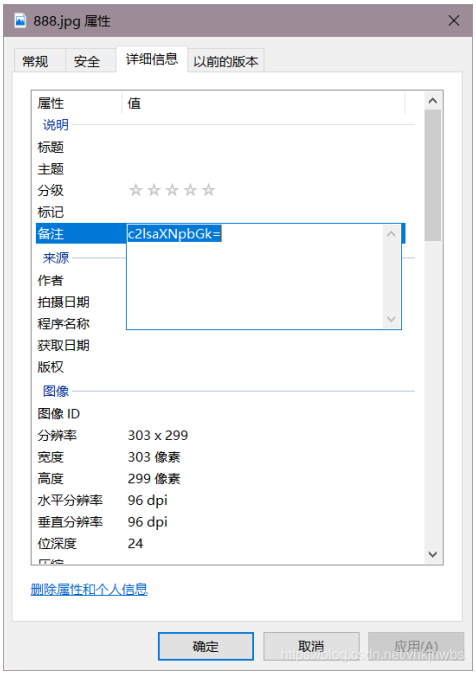
JPEG (jpg)，文件头：FFD8FFE0或FFD8FFE1或FFD8FFE8

（4）把8个文件都改为.jpg 发现第二张图片中有二维码：

扫码是：bilibili 输入 发现不对！！！仔细看提示**flag形式 flag{xxx_xxx_xxx} 有三部分**，肯定还有其他的两部分flag

查看8个图片的属性 发现 第 2,3,4 张图片 的大小为 20k 其他的为 90k

查看第三张图片：发现一个base64 码：



接着 分离 第四张图片 发现有一份里面还一张图片 是一个二维码 扫码得到 panama

于是得到 flag{bilibili_silisili_panama}

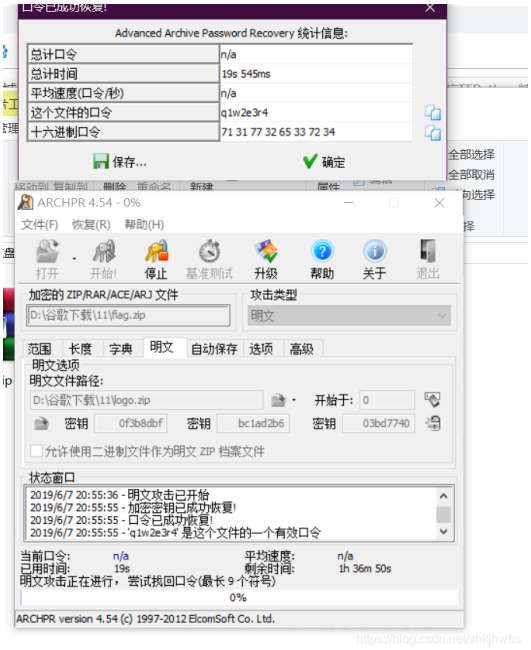
8,神秘的文件

（1）下载下来一个压缩包，解压里面 还有一个压缩包 和 一张图片，而且压缩包是加密的 eneneen

（2）明文攻击是一种较为高效的攻击手段，大致原理是当你不知道一个zip的密码，但是你有zip中的一个已知文件（文件大小要大于12Byte）或者已经通过其他手段知道zip加密文件中的某些内容时，因为同一个zip压缩包里的所有文件都是使用同一个加密密钥来加密的，所以可以用已知文件来找加密密钥，利用密钥来解锁其他加密文件

（3）ok 利用明文攻击，先将logo.png 压缩为 跟加密包相同的 类型 logo.zip

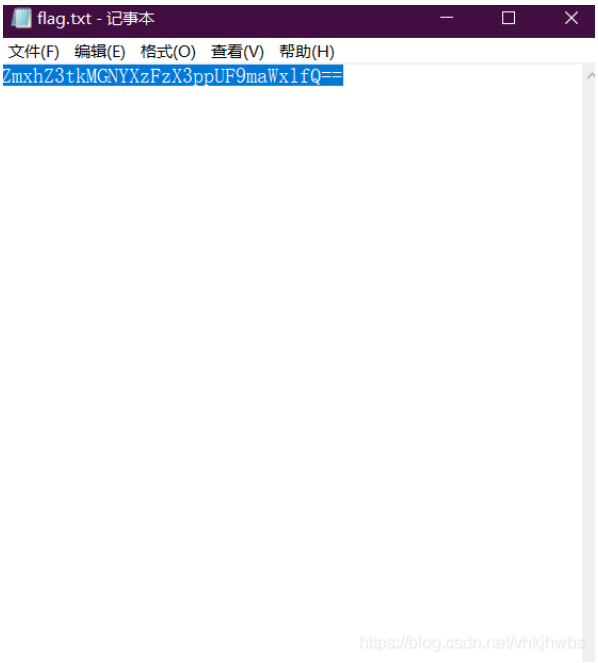
再用 ARCHPR 工具进行明文解密：（其实加密压缩包里面的文件是可以删除的）



解密出来 一张logo 和一个 .doxc 文件 ， 发现打不开，ok 改后缀为.txt 利用 搜索 一下flag （万一直接给了呢？）

然而并没有，只是搜到了 flag.txt 字样，ok 知道这个文件还隐藏有东西 利用foremost 分离一下

然后 打开分离出来的文件夹 找 flag.txt 文件 ：



发现是一串base64 编码，解密 得到：

flag{d0cX_1s_ziP_file}

9，论剑

（1），下载下来一张图片，用binwalk 审查一下，发现里面有两张图片，

```
root@ubuntu:/home/luohao/桌面# binwalk lunjian.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
17569         0x44A1          JPEG image data, JFIF standard 1.01

root@ubuntu:/home/luohao/桌面#
```

（2），用 winhex 审查 这三张图片，发现lunjian.jpg 中间有一段 二进制很可疑：

并且 发现了一段被修改的 7z压缩包的文件头，修复文件头，再用binwalk 审查一下：

```
root@ubuntu:/home/luohao/桌面# binwalk lunjian.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
9591          0x2577          7-zip archive data, version 0.4
17569         0x44A1          JPEG image data, JFIF standard 1.01

root@ubuntu:/home/luohao/桌面# dd if=lunjian.jpg of=lunjian.7z skip=9591 seek=17
569 bs=1
12399+0 records in
12399+0 records out
12399 bytes (12 kB, 12 KiB) copied, 0.0256373 s, 484 kB/s
```


再用foremost 分离，（没反应！！）

dd 提取：

```
dd if=lunjian.jpg of=lunjian.7z skip=9591 seek=17569 bs=1
```

```
F3 4F 1C E2 9E 30 B4 5F  70 BF 8F FD 3A FF D9 30 | óO âž0´_p¿ ý:ÿÙ0
31 31 30 31 31 30 31 20  30 31 31 31 31 30 30 31 | 1101101 01111001
20 30 31 31 30 31 31 31  30 20 30 31 31 30 30 30 |  01101110 011000
30 31 20 30 31 31 30 31  31 30 31 20 30 31 31 30 | 01 01101101 0110
30 31 30 31 20 30 31 31  30 31 30 30 31 20 30 31 | 0101 01101001 01
31 31 30 30 31 31 20 30  31 31 30 31 30 31 31 20 | 110011 01101011
30 31 31 30 30 31 30 31  20 30 31 31 31 31 30 30 | 01100101 0111100
31 20 30 30 31 30 30 30  30 31 20 30 30 31 30 30 | 1 00100001 00100
30 30 31 20 30 30 31 30  30 30 30 31 20 30 31 31 | 001 00100001 011
30 31 30 30 30 20 30 31  31 30 31 30 30 30 20 30 | 01000 01101000 0
31 31 30 31 30 30 30 37  7A BC AF 27 1C 00 04 CB | 11010007z¼¯'   Ë
B2 17 DF A0 1E 00 00 00  00 00 00 6A 00 00 00 00 | ² ß          j
00 00 00 6D C5 15 1E D8  C3 E1 A9 0E D6 5B E2 33 |    mÅ  ØÃá© Ö[â3
```


解压7z文件，发现需要密码，eneneen

回过头来 找密码，先处理这个二进制串：

```
sr = "01101101 01111001 01101110 01100001 01101101 01100101 01101001 01110011 01101011 01100101 01111001 00
flag = ""
for i in sr.split(" "):
 flag += chr(int(i,2))
print(flag)
```

得到： **mynameiskey!!!hhh**

这就是7z文件的密码：

解压又得到一张图片：fuck！又是一张图片

放进 winhex 审查

修改图片的高：

```
09 0C 0B 0C 16 0D 0D 16  32 21 1C 21 32 32 32 32        2! !2222
32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32  32 32 32 32 32 32 32 32  2222222222222222
32 32 32 32 32 32 32 32  32 32 32 32 32 32 FF C2  22222222222222ÿÂ
00 11 08 00 92 02 6E 03  01 22 00 02 11 01 03 11      ' n   "
01 FF C4 00 1B 00 01 00  02 03 01 01 00 00 00 00   ÿÄ
00 00 00 00 00 00 00 03  04 02 05 06 01 07 FF C4                ÿÄ
00 16 01 01 01 01 00 00  00 00 00 00 00 00 00 00
```

得到半部分flag ：

继续修改其他三张图片的高 ， 发现其他三张图片隐藏内容一样：

于是得到flag ：

```
Not flag{666C61677B6D795F6E616D655F482121487D}hhhh
```

说不是flag ！！！ 试一下 果然不是 ！！！！ 我靠，默默骂了一句傻逼玩意

那就解密吧，那么疑问来了，这到底是什么类型的密文呢？？？？？

搞了半天，查看了大牛的 wp 原来是 base16 ：

```
import base64
cha= '666C61677B6D795F6E616D655F482121487D'
print(base64.b16decode(cha))
```

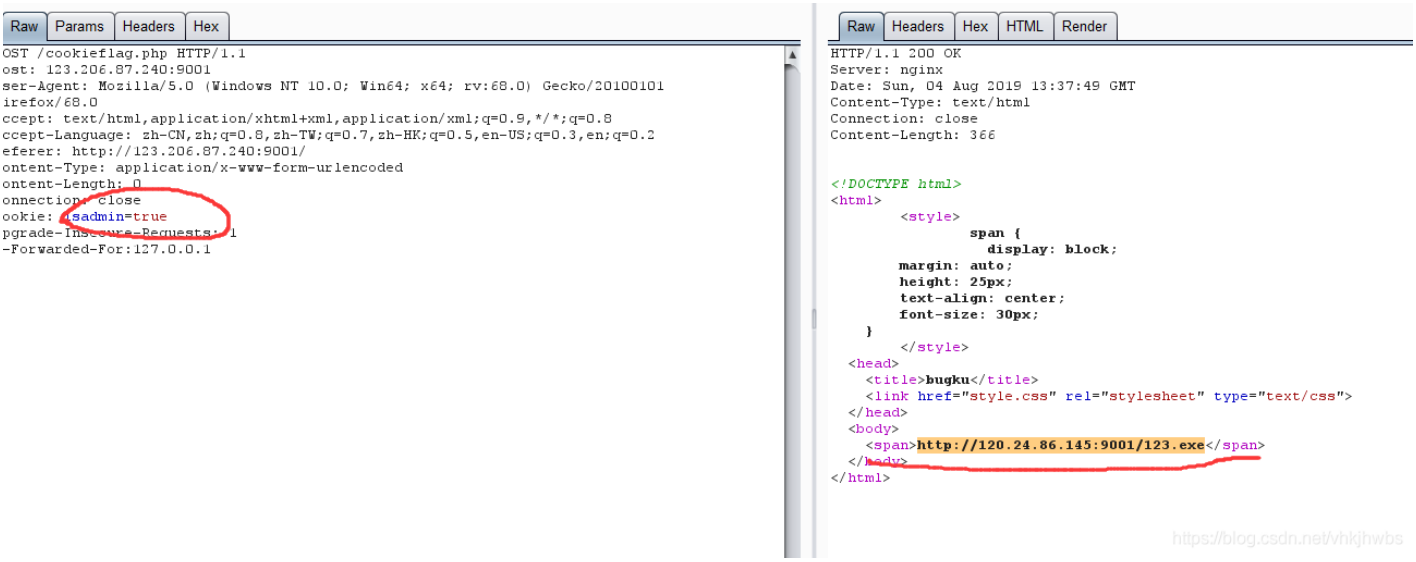
输出：

```
flag{my_name_H!!H}
```

默默说了一句 好变态啊这题！！！！

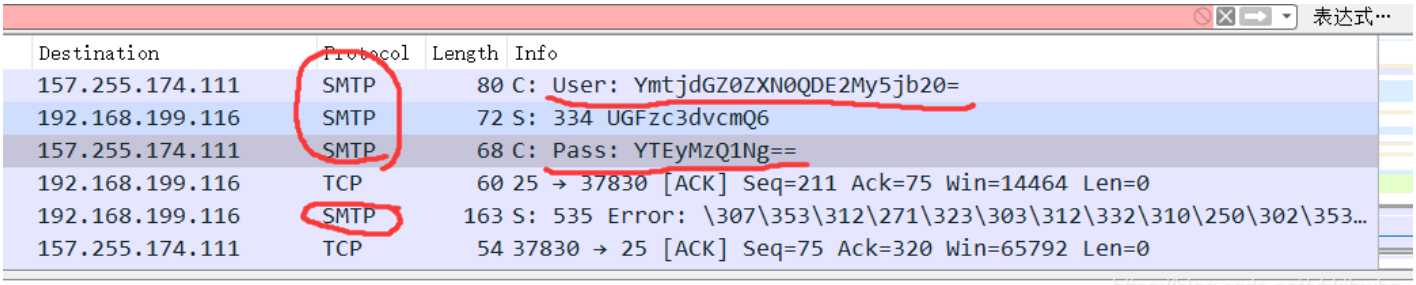
10，账号被盗了

1，用burp suit 抓包：将 isadmin = false 改为 true：得到一个 ip



自动下载一个文件，

用wireshark 抓取数据包，分析数据包：



有base64 密文，解密得到一个邮箱：

bkctftest@163.com

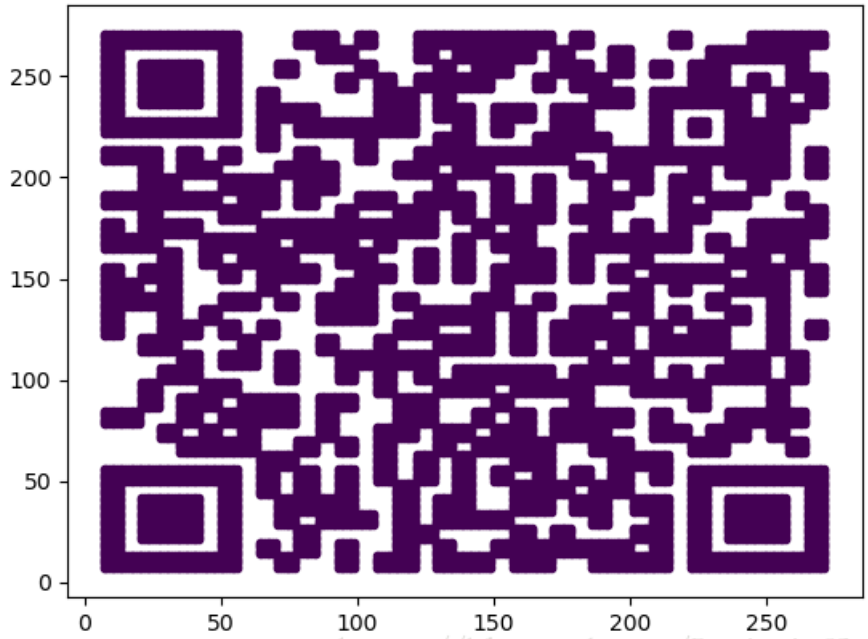
a123456

登录进去就找到了 flag：
flag{182100518+725593795416}

11 图穷匕见

1，下载得到一个图片，用winhex打开发现jpg文件尾FF D8后面有大量16进制数据，复制后面的数据，用notepad++中插件convert，hex->ascii转换后是一个个坐标点。
将括号和逗号去掉保存为txt文件。

2，用gnuplot（windows）打开刚才的txt文件得到，一个二维码：



扫描得到：

flag{40fc0a979f759c8892f4dc045e28b820}

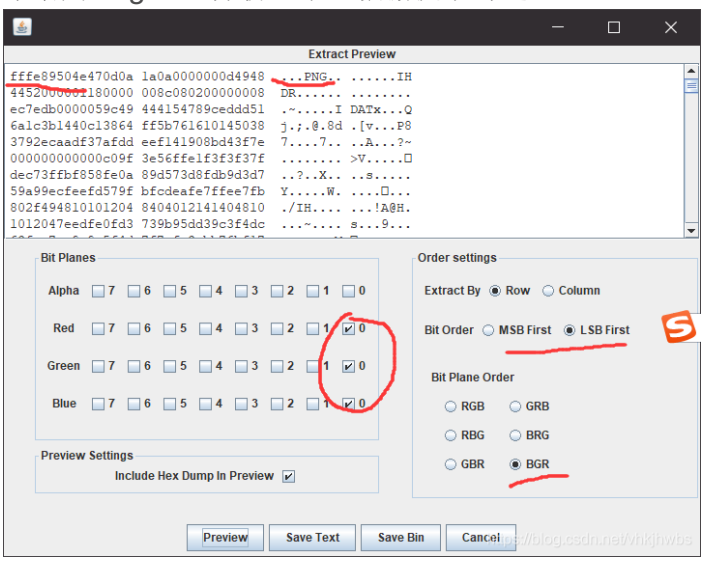
12，猫片（安恒杯）这题是真的.....

1下载下来一个文件名为png 的没有后缀名的文件：然后又看了下题中的提示，猜测是个png图片

修改后缀名后打开：

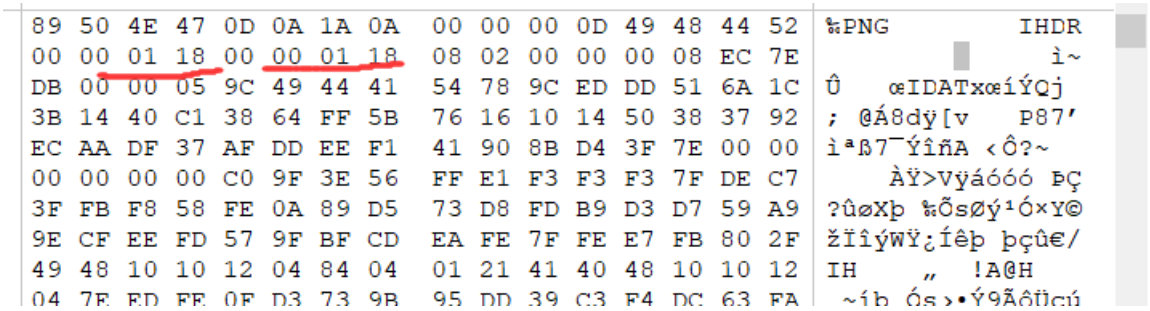
用binwalk看一下没有隐藏什么压缩文件：

果断用stegsolve分析一下，根据提示勾选：



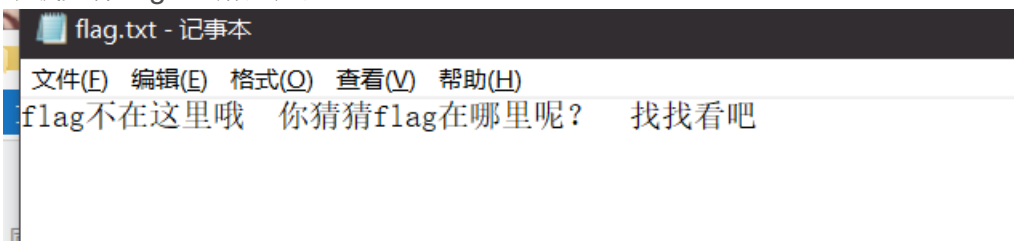
之后发现是一个文件头损坏的png文件，点击save bin 保存为png文件后用 winhex 把文件头前面的ff fe删去保存

打开后发现是半张二维码，那就在修改一下图片的高吧，



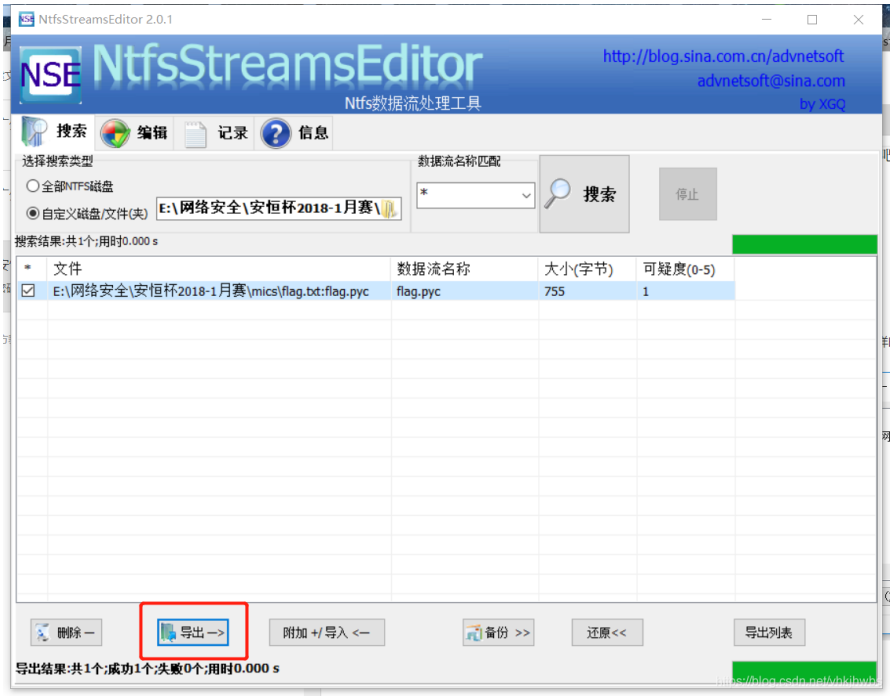
之后得到完整的二维码，，但却扫不出来东西，用stegslove 反色一下，扫出来是一个百度网盘的文件地址

下载文件flag.rar 解压后是



嗯……？？真是太恶心了，然后就不知道怎么找了，看一下别人的博客，发现需要搞数据流？？？？？

接着用ntfstreamseditor，查看解压的文件夹里面的数据流，然后把他导出来，得到一个pyc文件，也就是py编译后的文件，因此需要扔到网上去在线反编译一下



所有 开发类 站长类 极客类 其它 HR 码农文库 奇淫巧技 软件推荐

请选择pyc文件进行解密。支持所有Python版本

选择文件 未选择任何文件

```
#!/usr/bin/env python
# encoding: utf-8
# 访问 http://tool.lu/pyc/ 查看更多信息
import base64

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

    return ciphertext[::-1]

ciphertext = [
    '96',
    '65',
    '93',
    '123',
    '91',
```

美化(Beautify) 下载(Download)

搞一个解密脚本泡一下：

```
def decode():
    cc = ['96', '65', '93', '123', '91', '97', '22', '93', '70', '102', '9
    flag=''
    cc.reverse()
    for i in range(len(cc)):
        if i%2==0:
            s = int(cc[i]) - 10
        else:
            s = int(cc[i]) + 10
        s = chr(i^s)
        flag += s
        print(flag)
decode()
```

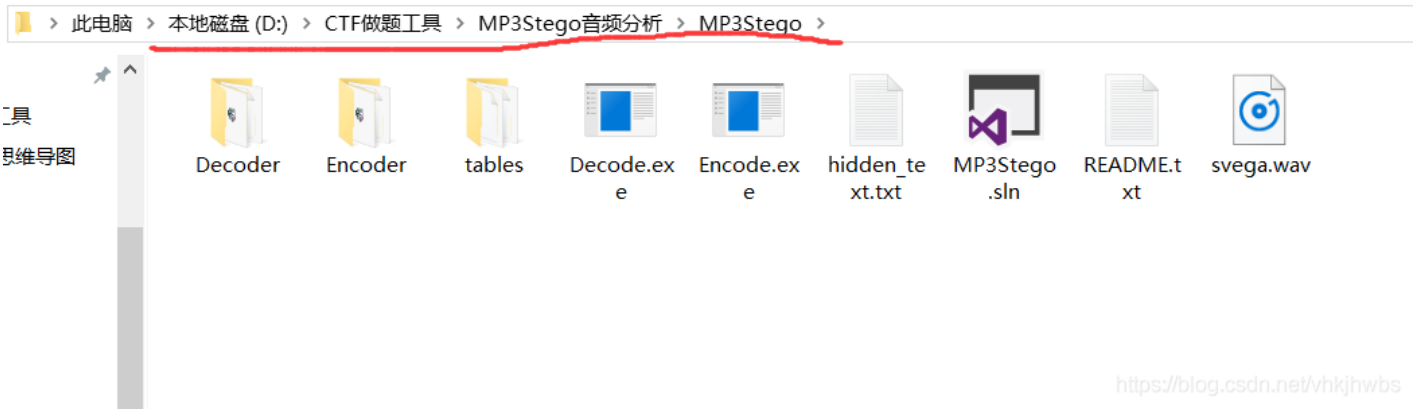

最后跑出flag：flag{Y@e_Cl3veR_C1Ever!}　　真难啊！

13，旋转跳跃

1，根据提示，音频中隐藏的有东西，果断用mp3stego

2，方法下载MP3stego 后解压

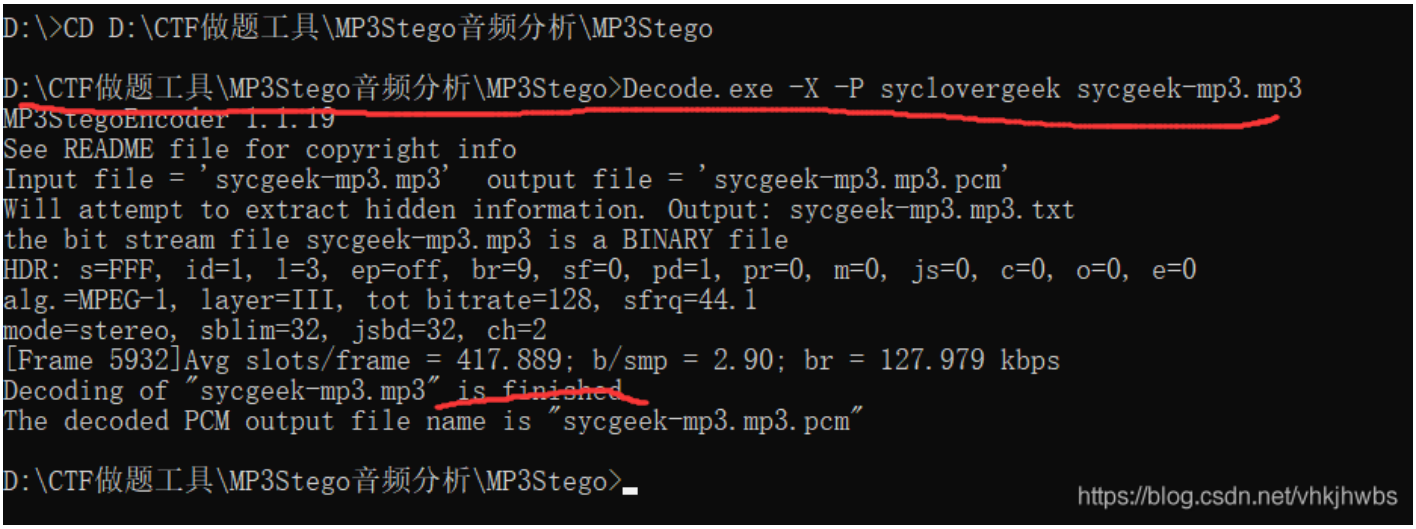
打开cmd 切换到 MP3stego所在的目录：



然后把刚才下载的音频放在文件夹里面

再输入命令：

```
Decode.exe -X -P syclovergeek sycgeek-mp3.mp3
```



解密出来的文件就在刚才的目录，打开找到flag：
SYC{Mp3_B15b1uBiu_W0W}

14，多彩

简单记录一下，

1，把色条图片下载下来，放到stegslove中分析一下，发现有一个pk的文件头

2，以rar格式保存下来，解压发现需要密码

3，回想还有一张色条，难道是口红色号，看一下别人的writeup 还Tm真是，一顿操作猛如虎，最后得到密码：白学家

4解压得到flag：

flag{White_Album_is_Really_worth_watching_on_White_Valentine's_Day}

15，convert

本题打开是一大串二进制，题名为 convert ，所以把数据放进 converter 里面尝试解码，**解出来了个 Rar!**

难道是一个rar文件的二进制形式？？

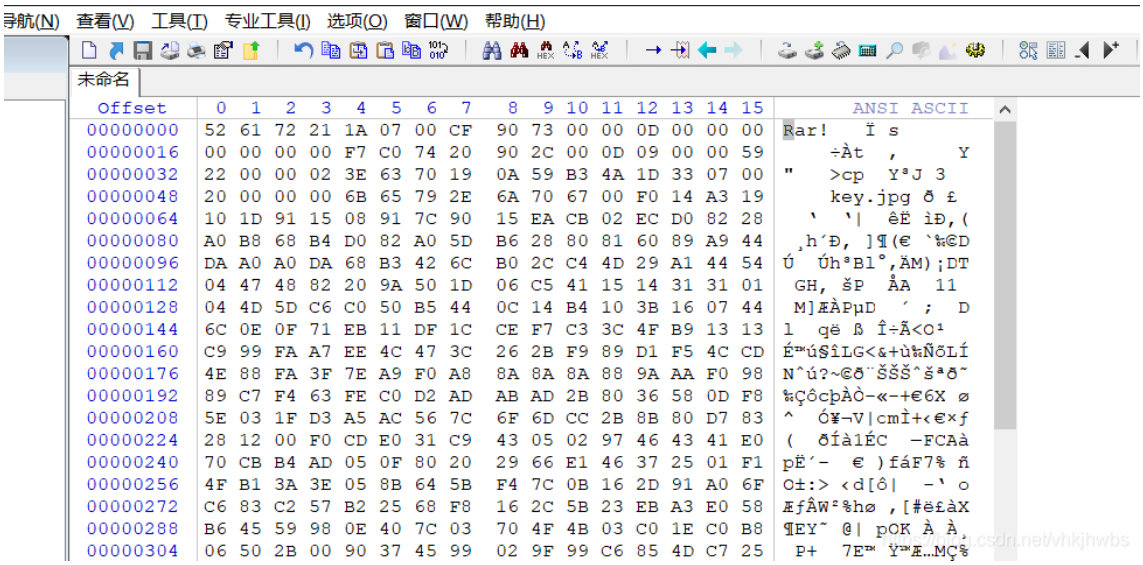
这里画一个知识点 二进制文件 直接把改后缀名改为rar 是不可以

要先 把二进制转为 16进制 文件在将转为字符 去改后缀名

这里用脚本先把二进制转为十进制再转为16进制

```
a ="0101001001100001011100100010000100011010000001110000000001100111110010000011100110000000000000000000000110
beginnum = 0
endnum = 4
# allnum=19104
b = []
f = open('D://A.txt','w')
while endnum <=19104:
    b.append(hex(int(a[beginnum:endnum], 2)))
    #print(hex(int(a[beginnum:endnum],2)))
    beginnum = endnum
    endnum += 4
for i in b:
    i = i[2:]
    print(i)
    f.write(i)
f.close()
```

将得到的 A.txt文件里的内容 复制 放到winhex 中 注意是复制过去 不是直接在winhex中打开文件



可以看到是一个rar文件头

直接存为 rar文件

解压是一张图片，在图片的属性信息中发现一串 base64密文 解码后得到 flag

flag{01a25ea3fd6349c6e635a1d0196e75fb}

16,好多数值

打开1.txt 文件 发现是3列数值：

255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255
255, 255, 255

刚开始我以为是，三维坐标，然后用 gnuplot 画图工具 画了个3D 图 什么也没发现

后来一想，数值有点特殊啊 0~255 这不是 rgb嘛 我去。。。。

找了个Python脚本 ：

```
from PIL import Image
x = 503 #x坐标  通过对txt里的行数进行整数分解
y = 122 #y坐标  x*y = 行数

im = Image.new("RGB",(x,y))#创建图片
file = open('1.txt') #打开rbg值文件

#通过一个个rgb点生成图片
for i in range(0,x):
  for j in range(0,y):
    line = file.readline()#获取一行
    rgb = line.split(",")#分离rgb
    im.putpixel((i,j),(int(rgb[0]),int(rgb[1]),int(rgb[2])))#rgb转化为像素
im.show()
```

这里需要先装 PIL 库 windows环境下 PIL库 是在Pillow 库中

```
在高版本中PIL库包含在Pillow库中，可以通过命令: pip install Pillow 来安装
```

然后得到了一张图片：flag:{youc@n'tseeme}

flag{ youc@n'tseeme }

17，PEN_AND_APPLE

提示是Windows下的type命令，Windows下的type命令可以显示文件内容。

利用NTFS流文件隐藏

首先用记事本新建两个文本文档,分别名为“1.txt”“2.txt”,其内容为“正常文件、数据流文件”,打开CMD命令行窗口,进入两个文件所在文件夹,输入 type 2.txt>1.txt: shujuliu.txt,回车.即可将文件2.txt的内容加入1.txt,内容以数据流方式保存,该数据流名为shujuliu.txt.在资源管理器中查看宿主文件1.txt,发现文件的修改日期和文件大小 都无变化,现在删除2.txt,执行命令:notepad 1.txt:shujuliu.txt ,即可查看数据流文件中的文件内容了.

用alternatestreamview可以提取出其中的文件，得到flag

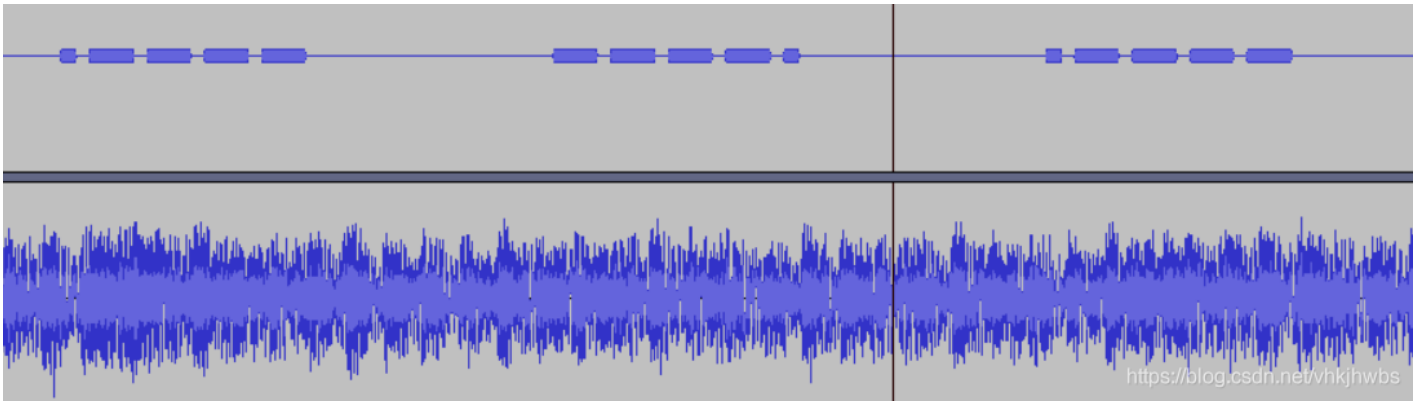
(说实话自己做了做不出来哈啊哈哈哈哈哈哈)

这道题有些脑洞，不过也是一种很常见的隐写技术，后来提示与win的type命令有关，可以找到相关资料，type命令可以用于ntfs文件写入，通过工具“`alternatestreamview“`扫描文件得到如下所示

SYC{Hei_hei_hei}

18，听首音乐

下载是一个 wav格式的音频 用 Audacity音频分析软件打开 ：



明显是一段 摩斯电码： . 空格 -

得到：

..... -... -.-. ----. ..--- ..... -.... ....- ----. -.-. -... ----- .---- ---.. ---.. ..-. ....--- . -.... .---- --... -.. --... ----- ----. ..--- ----. .---- ----. .---- -.-.

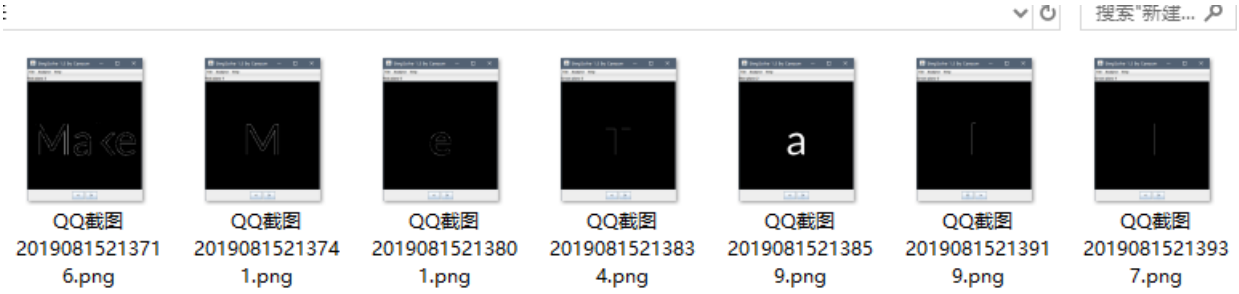
解密后得到：
5BC925649CB0188F52E617D70929191C

直接提交就可以了

19 color

打开是 7张颜色图片 看一下属性 没发现什么东西

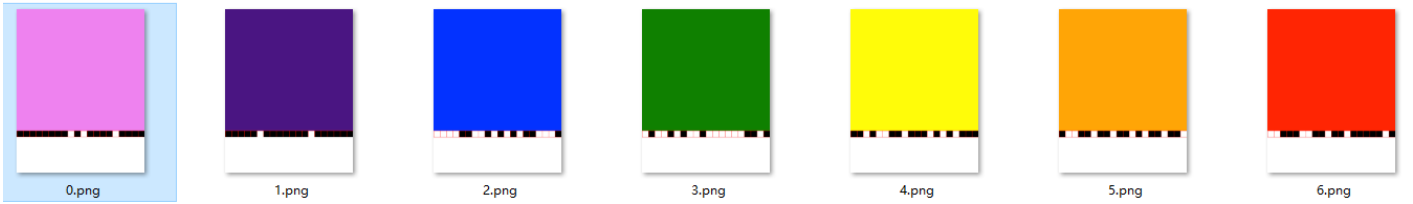
然后把图片 放进 stegsolve 里面分析 发现 有 点东西：



得到 Make Me Tall 提交不正确

那就是让 把图片 调高一点

然后得到 ：



把每张图片的 黑点 译为 1 ， 白点 译为 0

得到7组二进制码 然后发现每一列 能翻译为一个字符 然后用脚本跑一下：

```
c1 = '11111111010111101111'
c2 = '11111011111110111111'
c3 = '00001100101010110001'
c4 = '01001010010000001101'
c5 = '11010011011101010111'
c6 = '10011011011010110110'
c7 = '00111001101101111101'

flag = ''

for i in range(0, 20):
    c = c1[i] + c2[i] + c3[i] + c4[i] + c5[i] + c6[i] + c7[i]
    flag += chr(int(c, 2))

print(flag)
```

得到： flag{Png1n7erEs7iof}

20 怀疑人生

打开压缩包 发现有 三个文件 （一个压缩包 两张图片）

第三张图片是一张二维码 扫描 得到：

是flag的一部分 那肯定还有两部分在 其他两个文件里

第二个图片 用 foremost 分离出来 一个 压缩包 解压出来是一个文件 ctx2.txt 文件 是一段 OOk密文 解密出来

3oD54e

第一个 文件是一个 压缩包 有密码 用 明文解锁 得到 密码 ： password

得到 一段 base64 密文 ， 解密得到 一段 Unicode 密文 解密得到 另一段 flag：

flag{hacker

三部分拼接在一起 提交 发现不对

可能第二部分 有问题：看了一下别人的博客 发现是 base58

解密后得到：

```
misc
```

拼接后得到：
flag{hackermisc12580}

21 红绿灯

打开是一张 动态图：

猜测能动态图里面可能隐藏了 一段密文 有三种状态 红 黄 绿 ， 猜测可能是 二进制密文 、摩斯密码、也有可能是 OOK密码

放进 stegsolve 里分析一下 发现竟然有 1168 帧 居然有这么长！！！！ 那肯定是这几种密码中比较简单的 了

先猜测是二进制 ： 红 ： 1

黄 ： 分隔

绿 ： 0

当然是不可能 用 手工去一张一张 保存的

找到一位大佬的脚本：http://virgin-forest.top/2019/03/21/ctf-traffic_light/

首先分离每一帧：

```
import os
from PIL import Image

def seoaration_gif(gif_file):
    png_dir = gif_file[:-4] + '/'
    os.mkdir(png_dir)
    img = Image.open(gif_file)
    try:
        while True:
            current = img.tell()
            img.save(png_dir+str(current)+'.png')
            img.seek(current+1)
    except:
        pass

if __name__=='__main__':
    gif_file = 'Traffic_Light.gif'
    seoaration_gif(gif_file)
```

识别 颜色：（这里需要改变脚本中的 图片名 来获取 颜色的 RGB）

```
import colorsys
import PIL.Image as Image


def get_dominant_color(image):
    max_score = 0.0001
    dominant_color = None
    for count, (r, g, b) in image.getcolors(image.size[0] * image.size[1]):
        # 转为HSV标准
        saturation = colorsys.rgb_to_hsv(r / 255.0, g / 255.0, b / 255.0)[1]
        y = min(abs(r * 2104 + g * 4130 + b * 802 + 4096 + 131072) >> 13, 235)
        y = (y - 16.0) / (235 - 16)

        # 忽略高亮色
        if y > 0.9:
            continue
        score = (saturation + 0.1) * count
        if score > max_score:
            max_score = score
            dominant_color = (r, g, b)
    return dominant_color


if __name__ == '__main__':
    image = Image.open('Traffic_Light/1.png')
    image = image.convert('RGB')
    print(get_dominant_color(image))
```

得到：

经过测试获得四种颜色的RGB值：

- 灰色(172, 172, 172)
- 红色(254, 0, 0)
- 绿色(7, 253, 8)

- 黄色(254, 254, 0)

分析出颜色后的判断：

- 红为1
- 绿为0
- 黄色作为分隔
- 灰色直接跳过

将每一帧 转化为相应的 字符 并将得到的 二进制 转化为字符

```
import colorsys
import PIL.Image as Image


def get_dominant_color(image):
    max_score = 0.0001
    dominant_color = None
    for count, (r, g, b) in image.getcolors(image.size[0] * image.size[1]):
        # 转为HSV标准
        saturation = colorsys.rgb_to_hsv(r / 255.0, g / 255.0, b / 255.0)[1]
        y = min(abs(r * 2104 + g * 4130 + b * 802 + 4096 + 131072) >> 13, 235)
        y = (y - 16.0) / (235 - 16)

        # 忽略高亮色
        if y > 0.9:
            continue
        score = (saturation + 0.1) * count
        if score > max_score:
            max_score = score
            dominant_color = (r, g, b)
    return dominant_color


if __name__ == '__main__':
    numbers = ''
    flag = ''
    for i in range(1, 1168):
        file_name = 'Traffic_Light/' + str(i) + '.png'
        image = Image.open(file_name)
        image = image.convert('RGB')
        rgb = get_dominant_color(image)
        if rgb == (254, 254, 0): numbers += ' '
        if rgb == (254, 0, 0): numbers += '1'
        if rgb == (7, 253, 8): numbers += '0'

    for i in numbers.split(' '):
        flag += chr(int(i, 2))

    print(flag)
```

执行后得到 flag

flag{Pl34s3_p4y_4tt3nt10n_t0_tr4ff1c_s4f3ty_wh3n_y0u_4r3_0uts1d3}

说实话 我对 Pillow 库中的image 不太了解 ： https://blog.csdn.net/yjwx0018/article/details/52852067

去找了个教程看了一下，有兴趣的同学可以去看看

22，不简单的压缩包

23，一枝独秀

24,好多压缩包

25，一个普通的压缩包

题目下载下来是一个 zip.rar,打开后发现里面有个flag.txt,打开以后…是 flag is not here

所以目标点应该不在这。

懒得看文件头啥的了,直接拖入kali,binwalk一下

```
root@kali:~/桌面/tmp# binwalk zip.rar

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
118           0x76            RAR archive data, first volume type: MAIN_HEAD
5909          0x1715          End of Zip archive
```

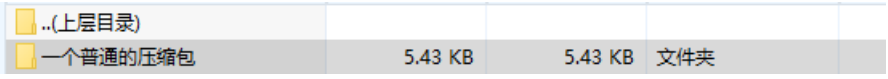
- 发现有一个rar文件和一个zip文件。直接foremost分离它们。

```
1
```

```
foremost zip.rar
```

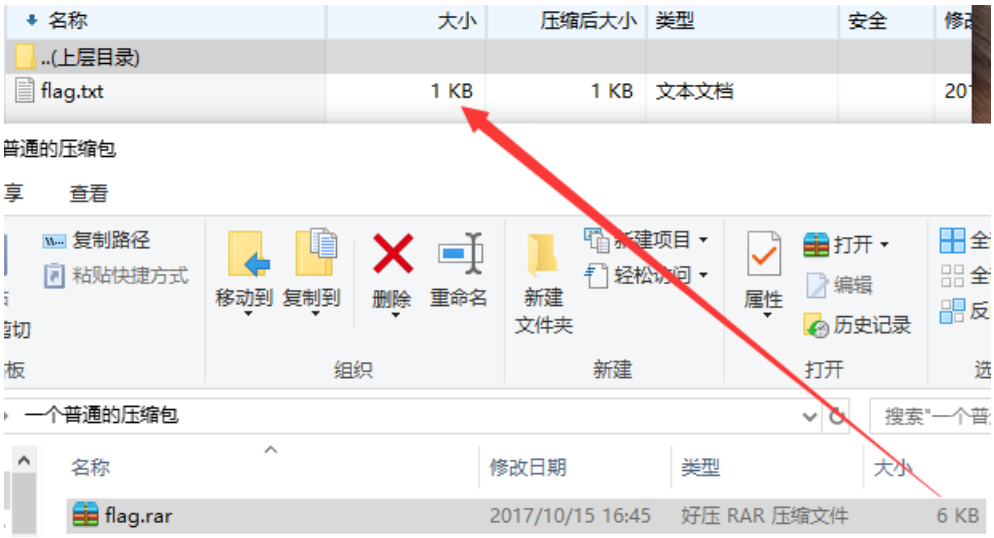
分离得到的rar就是原来的那个zip.rar脱去那个zip压缩包后的rar。

而zip压缩包里面有一个名为“一个普通的压缩包”的文件夹

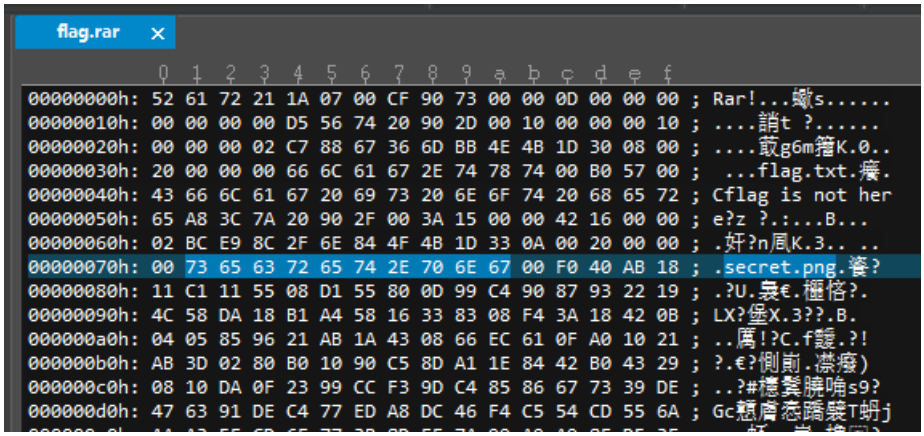


- 文件夹里面是一个flag.rar,打开rar文件以后里面还是flag.txt,内容依旧是flag is not here

但是值得注意的是：rar文件大小有6kb,但是那个flag.txt只有1kb



- 用十六进制查看器对这个rar压缩包查看了一番, 发现里面有一个secret.png



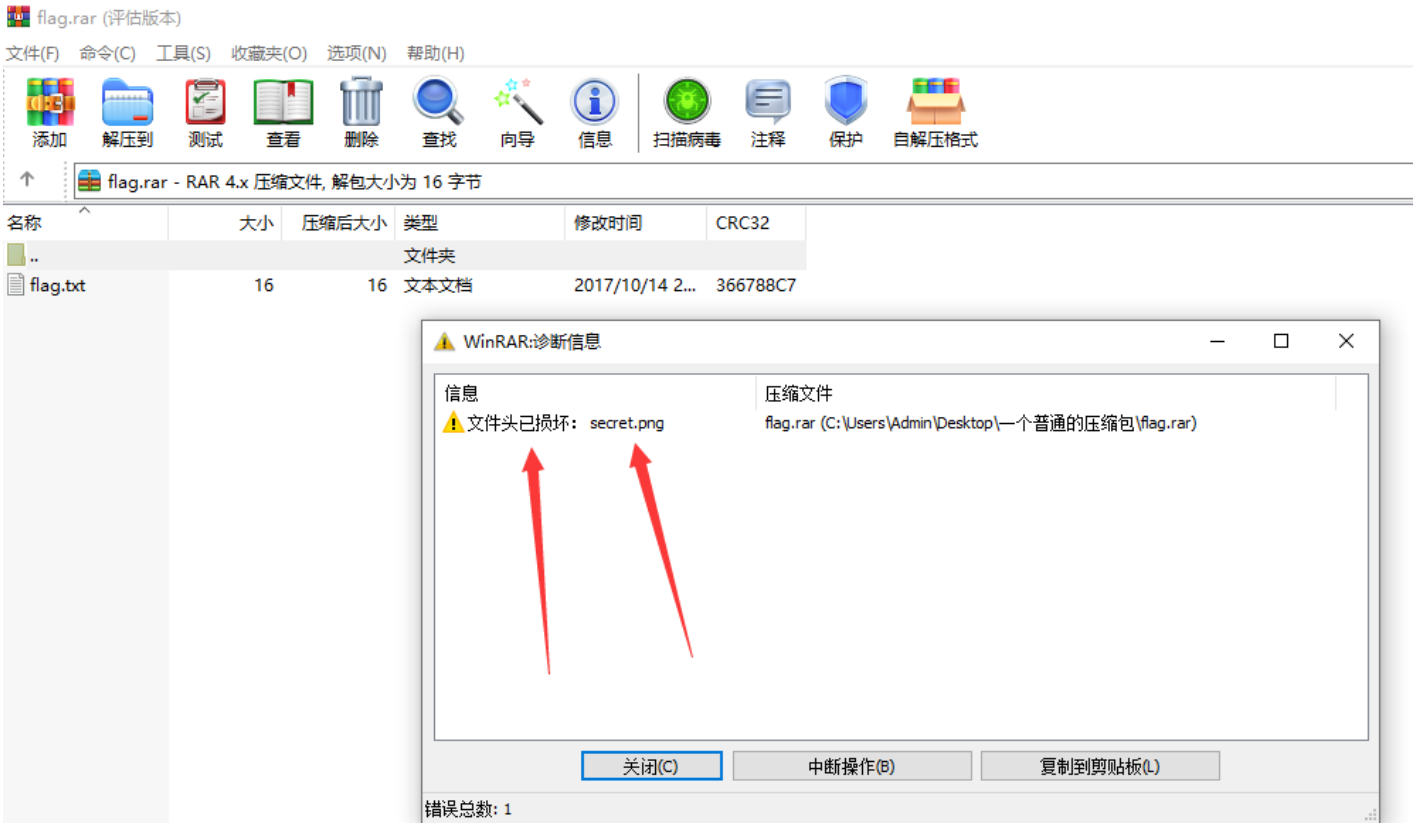
但是为什么双击打开rar的时候却没显示这个png文件？

搜索一番后得知, 这是因为我用的2345解压, 这款软件会自动过滤掉不完整的文件,这个不完整是指缺少rar文件的一些标识。

具体知识可以在下面网站中学习。

传送门

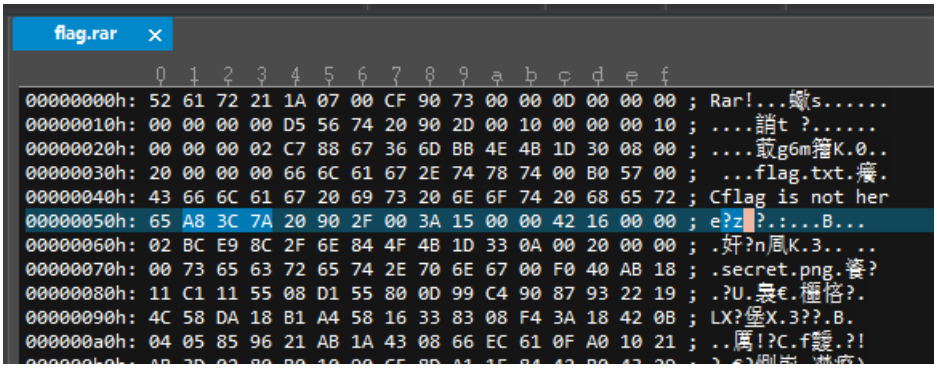
- 回归原始,用winrar打开这个rar压缩包,提示如下



所以目标就变成修复这个png文件的文件头。

用winrar和2345解压自带的修复功能都没有用… 只能手动了。

在flag.txt结束的地方就是下一个块开始的地方,flag.txt结束于 flag is not here的最后一个字母e。 则新块开始于图中A8 3C这个位置。



- 结合文档

2.2.3 文件头(FILE_HEAD)

HEAD_CRC 2 字节 从 HEAD_TYPE 到 FILEATTR 的 CRC 结构和文件名

HEAD_TYPE 1 字节 头类型: 0x74

HEAD_FLAGS 2 字节 位标记:

0x01 - 文件在前一卷中继续

0x02 - 文件在后一卷中继续

0x04 - 文件使用密码加密

0x08 - 文件注释存在

RAR 3.x 使用分开的注释块，不设置这个标记。

0x10 - 前一文件信息被使用(固实标记)

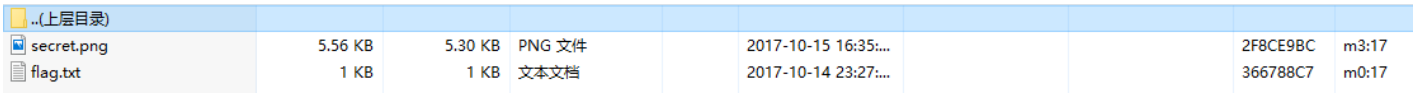
(对于 RAR 2.0 和以后版本)

7 6 5位(对于 RAR 2.0 和以后版本)

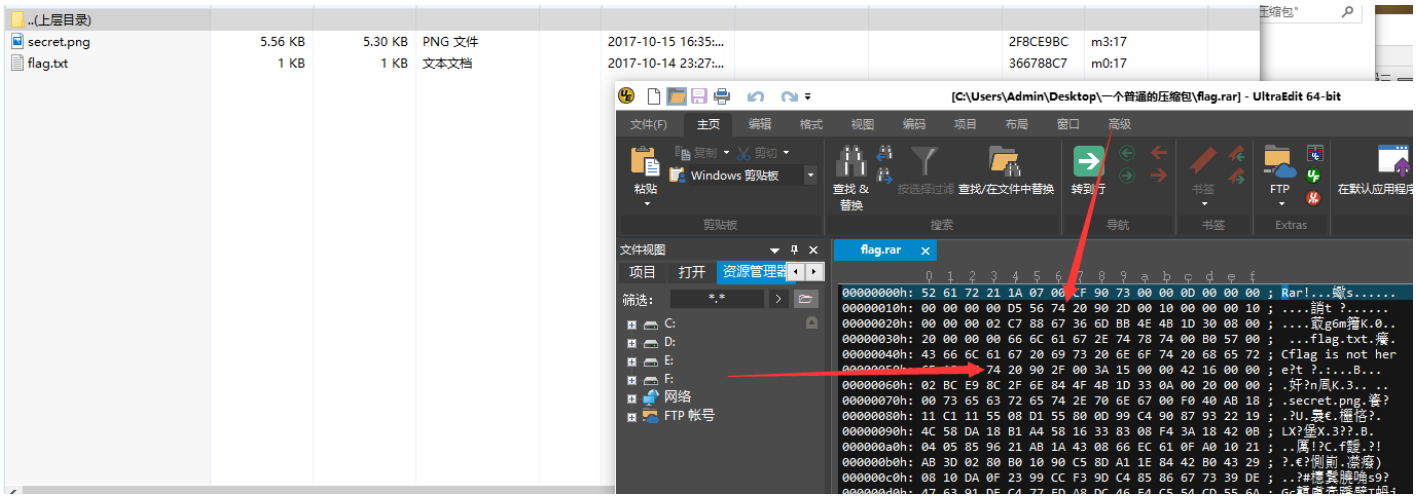
可知A8 3C就是对应的HEAD_CRC

但是HEAD_TYPE却不是0x74而是0x7A,将其改为0x74保存。

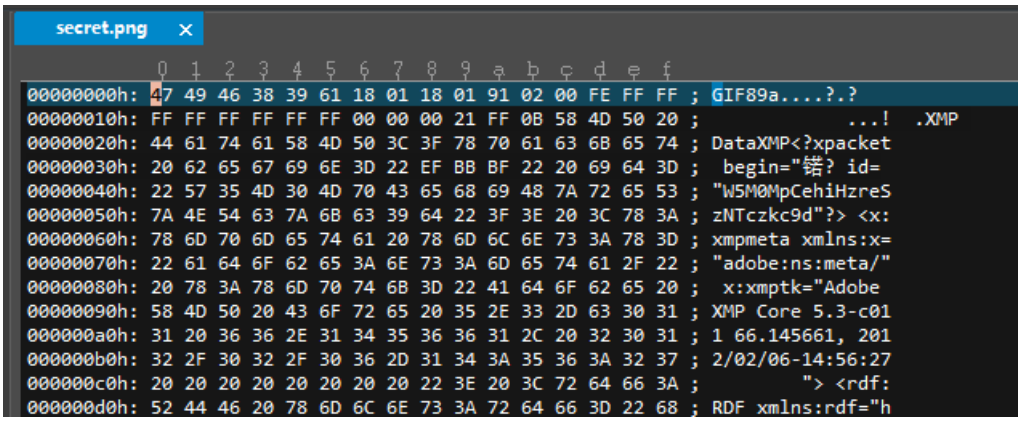
再次打开rar以后发现了secret.png, 文件总大小也差不多是rar文件的大小了。



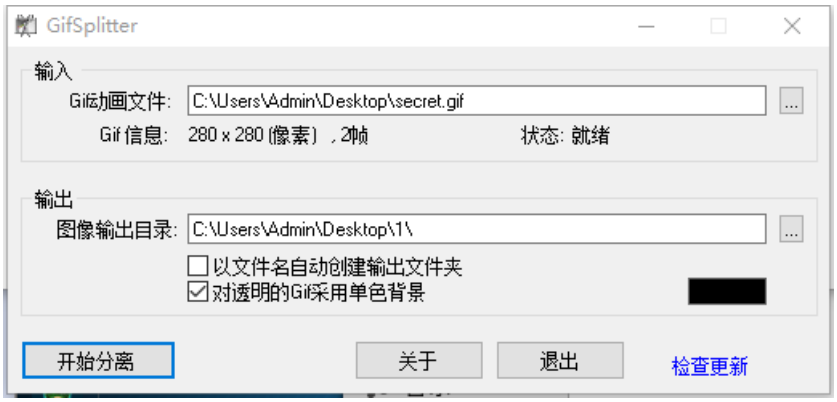
- 观察可以发现在十六进制查看其器中 HEAD_TYPE 总是位于该文件CRC32的第三位的上方。这一点可以用来验证,修改的位置是不是HEAD_TYPE。



- 下面对secret.png进行分析



- 是个gif文件,拓展名改为gif后开始分离



- 分离得到两张图片, 分别拖入Stegsolve

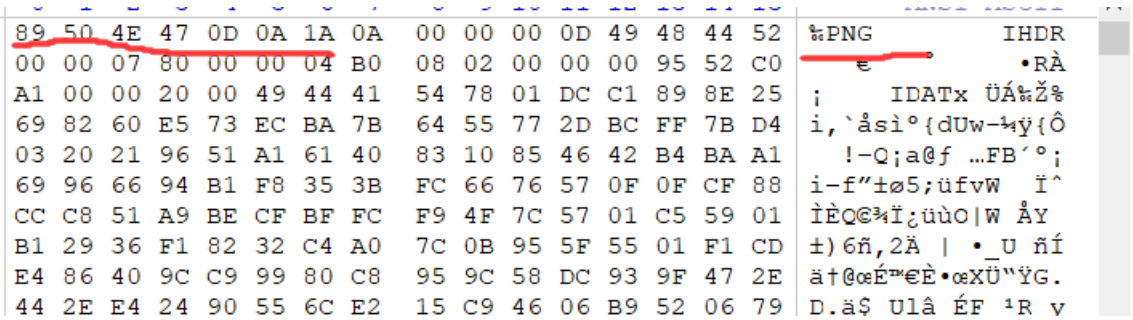
- 得到一张下半部分的二维码和一张不完整的上半部分的二维码。补全对应位置得到完整二维码。

- 扫码得到:

1	flag{yanji4n_bu_we1shi}

26， 2B

一，下载下来是一个没有后缀名的文件，放进 winhex中发现是 png头



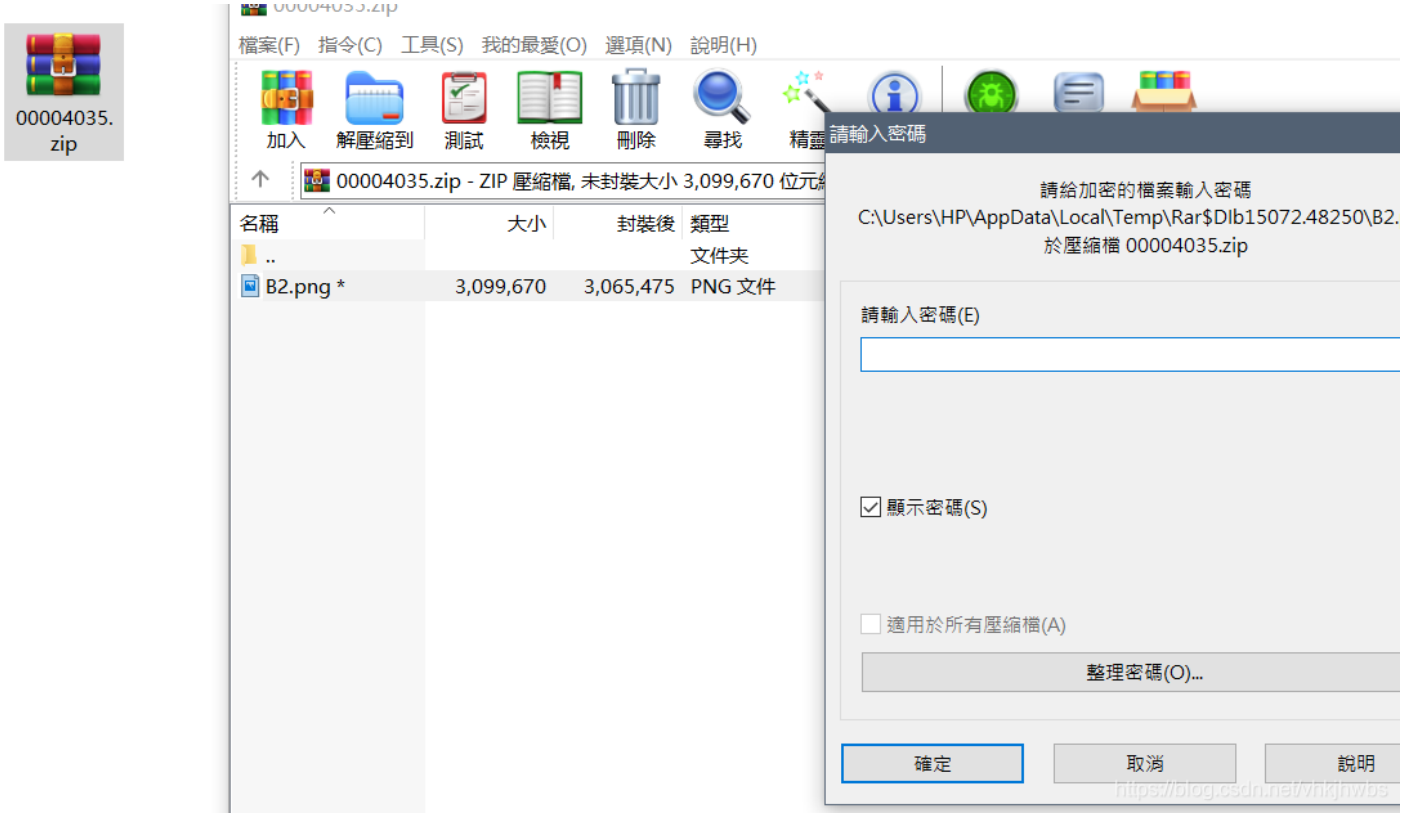
89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
00	00	07	80	00	00	04	B0	08	02	00	00	00	95	52	C0
A1	00	00	20	00	49	44	41	54	78	01	DC	C1	89	8E	25
69	82	60	E5	73	EC	BA	7B	64	55	77	2D	BC	FF	7B	D4
03	20	21	96	51	A1	61	40	83	10	85	46	42	B4	BA	A1
69	96	66	94	B1	F8	35	3B	FC	66	76	57	0F	0F	CF	88
CC	C8	51	A9	BE	CF	BF	FC	F9	4F	7C	57	01	C5	59	01
B1	29	36	F1	82	32	C4	A0	7C	0B	95	5F	55	01	F1	CD
E4	86	40	9C	C9	99	80	C8	95	9C	58	DC	93	9F	47	2E
44	2E	E4	24	90	55	6C	E2	15	C9	46	06	B9	52	06	79

%PNG												IHDR			
				€										•RÀ	
¡				IDATx				ÜÁ%Ž%							
i,`åsì°{dUw-¼ÿ{Ô															
				!-Q¡a@ƒ				…FB´°¡							
i-f"±ø5;üfvW								Ï^							
ÌÈQ©¾Ï¿üùO|W								ÅY							
±)6ñ,2Ä | •_U								ñÍ							
ä†@œÉ™€È•œXÜ"ŸG.															
D.ä$ Ulâ								ÉF				¹R			v

并且发现在文件尾部还发现了 B2.png 和 pk头

```
05131664  5A F6 52 22 36 CF 2C E1   5E BE 26 E4 B7 FB FE 8F  ZöR"6Ï,á^¾&ä·ûþ
05131680  1B 19 0B 42 C4 E4 7F 6F   AC 7E AB 02 FD 56 7A 49     BÄä o¬~« ýVzI
05131696  6D 2B FA C4 4F AF A8 3D   AF BB E9 FD 5E 79 6B DD  m+úÄO¯¨=¯»éý^ykÝ
05131712  00 6A AF 9D 79 99 A7 EC   DD 41 64 DF 19 E5 1F AC   j¯ y™§ìÝAdß å ¬
05131728  8F F9 7D 4B 70 AD F1 9F   C9 FF 07 50 4B 01 02 1F   ù}Kp-ñŸÉÿ PK
05131744  00 14 00 01 00 08 00 D0   00 BE 4C A4 20 76 B1 83         Ð ¾L¤ v±ƒ
05131760  C6 2E 00 16 4C 2F 00 06   00 24 00 00 00 00 00 00  Æ.  L/    $
05131776  00 20 00 00 00 00 00 00   00 42 32 2E 70 6E 67 0A            B2.png
05131792  00 20 00 00 00 00 00 01   00 18 00 FD 82 C6 02 67              ý,Æ g
05131808  F7 D3 01 7E BC 98 DF 67   F7 D3 01 08 C5 D7 5D 67  ÷Ó ~¼˜ßg÷Ó  Å×]g
05131824  F7 D3 01 50 4B 05 06 00   00 00 00 01 00 01 00 58  ÷Ó PK           X
05131840  00 00 00 A7 C6 2E 00 00   00                       §Æ.
```


把文件改为 .png 后 用 foremost 分离 出来一个压缩包 ， 解压时发现需要密码：



用暴力破解，破解不开，只能把希望寄托于 zip 伪加密 上了

但 自己对手工修改 文件头 一知半解 那最好是使用工具修复文件了

使用ZipCenOp.jar清除密码

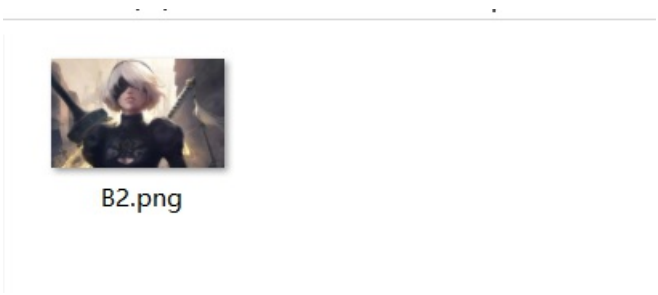
ZipCenOp.jar 下载地址：https://pan.baidu.com/s/1GHcUYA36X9reZL7rcmWNfA 提取码：ugyn

下载后 把它和压缩包放在同一个文件夹里 打开cmd 切换到 这个文件夹 执行：

```
java -jar ZipCenOp.jar r 00004035.zip
```

```
Windows PowerShell
PS D:\火狐下载\outfile\zip> java -jar ZipCenOp.jar r 00004035.zip
success 1 flag(s) found
PS D:\火狐下载\outfile\zip>
```

然后就得到一张 B2.png



两个一样的图片 很容易联想到 盲水印

接下来分离盲水印

附上一个分离盲水印的脚本：（将文件下载下来，解压）

```
https://github.com/linyacool/blind-watermark
```

将 B2.png 和 2B.png 放在方才下载的 blind-watermark 文件里

打开 文件夹 右上角的 windows powershall 或者 用 cmd 切换到当前目录 （powershall 功能比cmd更加强大）

输入指令：

```
TypeError: 'float' object cannot be interpreted as an integer
PS D:\迅雷下载\blind-watermark-master\blind-watermark-master> python2 decode.py --original B2.png  --image 2B.png --result flag.png
PS D:\迅雷下载\blind-watermark-master\blind-watermark-master>
```



```
python decode.py --original B2.png  --image 2B.png --result flag.png
```

或者：（如果电脑同时装了 Python 2 和 Python 3 的话 用下面的指令）

```
python2 decode.py --original B2.png  --image 2B.png --result flag.png
```

如果你没有安装 opencv 和 **matplotlib 这两个库 的话 可能会报错 得先安装这两个 库才能执行脚本**

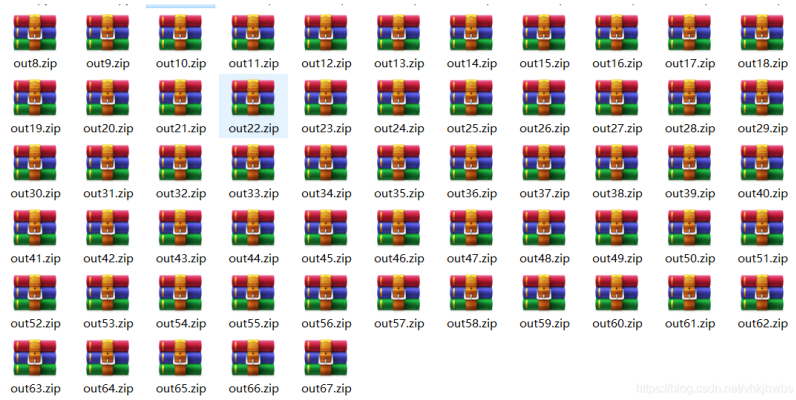
安装指令cmd 中 ：

```
window 10 环境下

// 安装opencv
pip install opencv-python

（如果你像我一样电脑中同时装了Python2  和 Python 3 的话 并设置好了相关路径的话，请用 下面的指令安装）

pip2 install opencv-python



//安装matplotlib

pip install matplotlib

或

pip2 install matplotlib
```

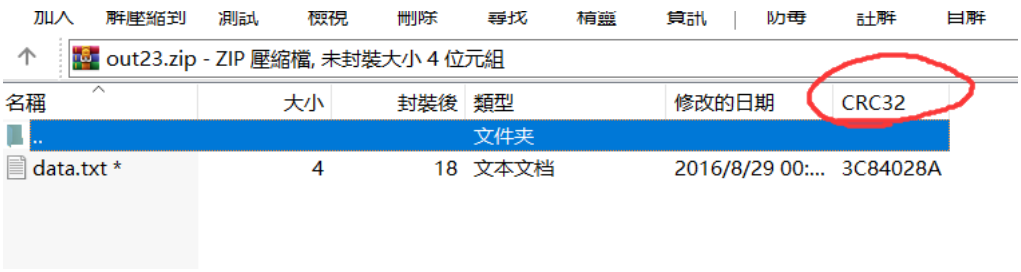
就会在当前文件夹 里 生成一张 flag.png 的图片：

27， 好多压缩包

开123.zip有68个压缩包，一脸懵逼，



随便打开一个压缩包，看到了 一个 crc32 值，



查了一些crc32 的相关知识：CRC全称Cyclic Redundancy Check，又叫循环冗余校验。CRC32跟md5，sha1一样都是哈希算法的一种。crc32的优势是速度快，它被设计的目的是用来检测数据在网络传输过程中可能出现的随机错误，他出现碰撞的几率很大，可以根据crc32 的值 逆向找到 相同crc32的字符串

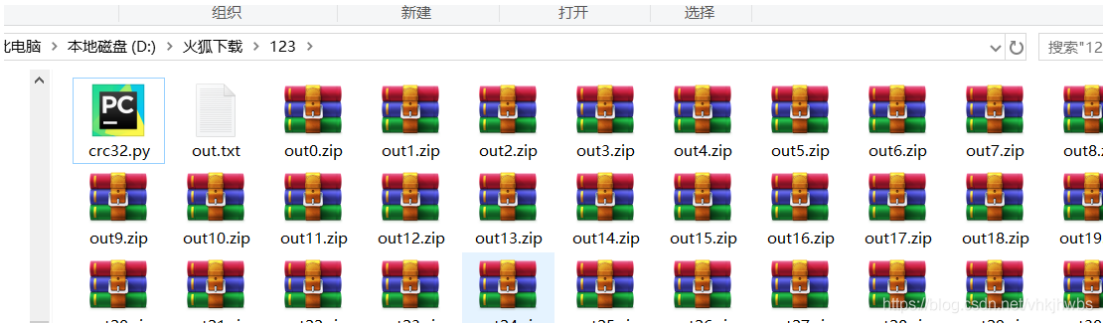
crc32 发生碰撞的几率： 1820w 个随机数中 冲突的量 有38638 个

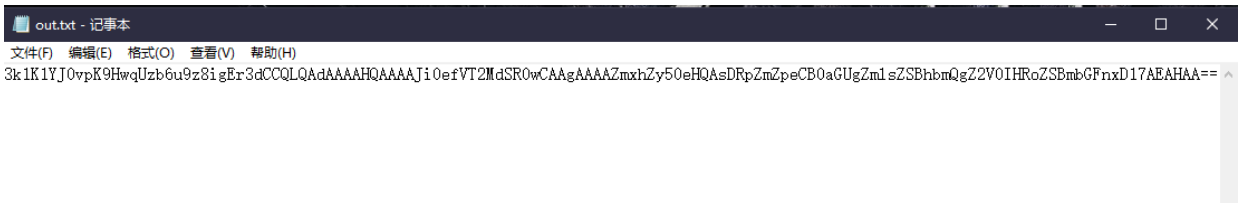
知道了要用 crc32 碰撞 还原出 每个 data.txt 中的数据

找了个代码：（python 2.7）

```
#coding:utf-8
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for p in dic:
                for q in dic:
                    s = i + j + p + q
                    if crc == (binascii.crc32(s) & 0xffffffff):
                        #print s
                        f.write(s)
                        return

def CrackZip():
    for I in range(68):
        file = 'out' + str(I) + '.zip'
        f = zipfile.ZipFile(file, 'r')
        GetCrc = f.getinfo('data.txt')
        crc = GetCrc.CRC
        #以上3行为获取压缩包CRC32值的步骤
        #print hex(crc)
        CrackCrc(crc)

dic = string.ascii_letters + string.digits + '+/='

f = open('out.txt', 'w')
CrackZip()
f.close()
```

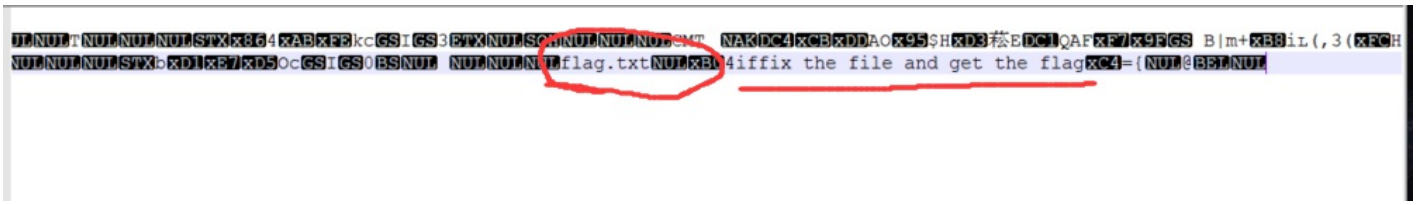
将该脚本保存为 crc32.py 然后和压缩包放在一起，打开cmd，cd到当前目录，执行该脚本就会生成一个out.txt文件





发现一段base64密文，用Notepad++ 打开out.txt ，然后用Notepad++中的 插件 》 MIME Tools》Base64 Decode

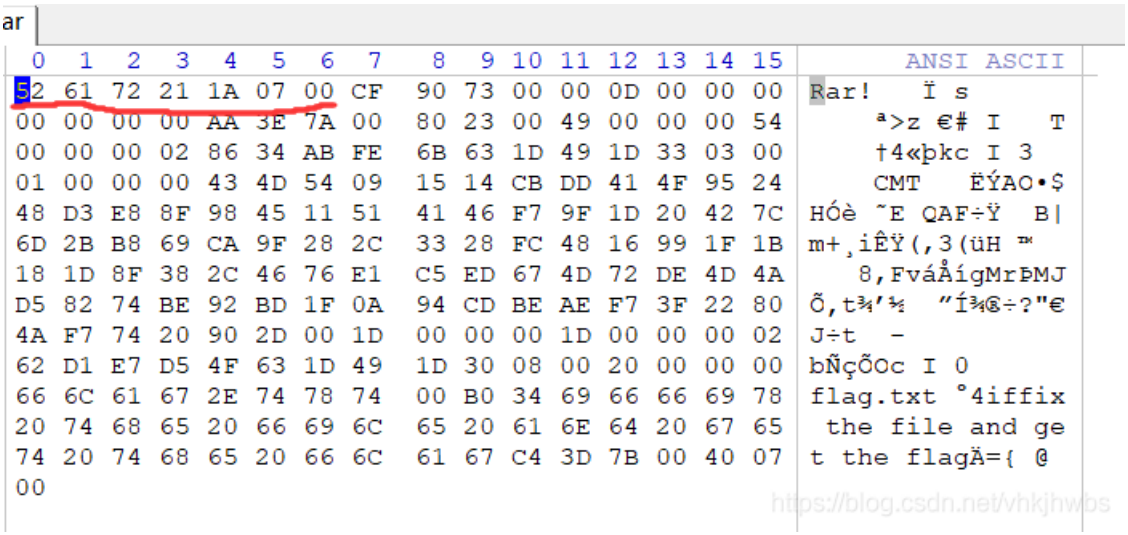
将这段密文解码：（**这不能直接将 密文直接拿到 解密网站解密 再复制保存，可能是有与编码的方式不能导致有些数据丢失了吧，反正这样做，后面解密的时候打不开文件**）



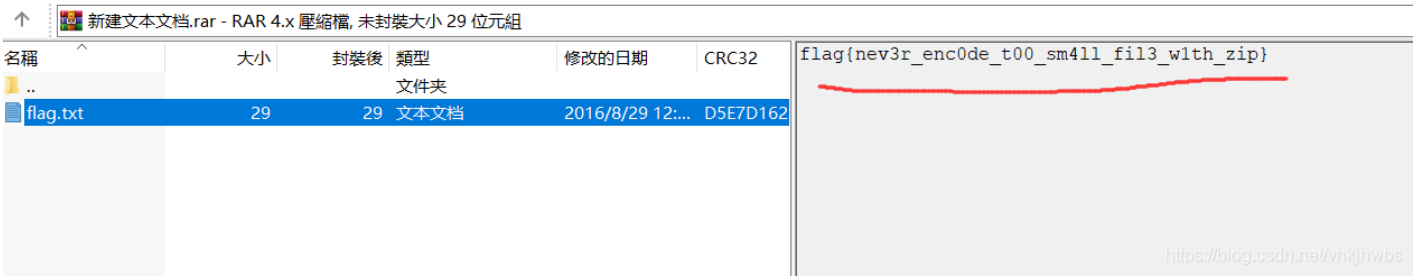
提示： fix the file and get the flag

让修复文件，保存之后 用 winhex 打开，发现是 rar文件尾　却没有rar文件头，

加上 rar文件头： 52 61 72 21 1A 07 00



然后将文件 保存为 rar文件　打开：



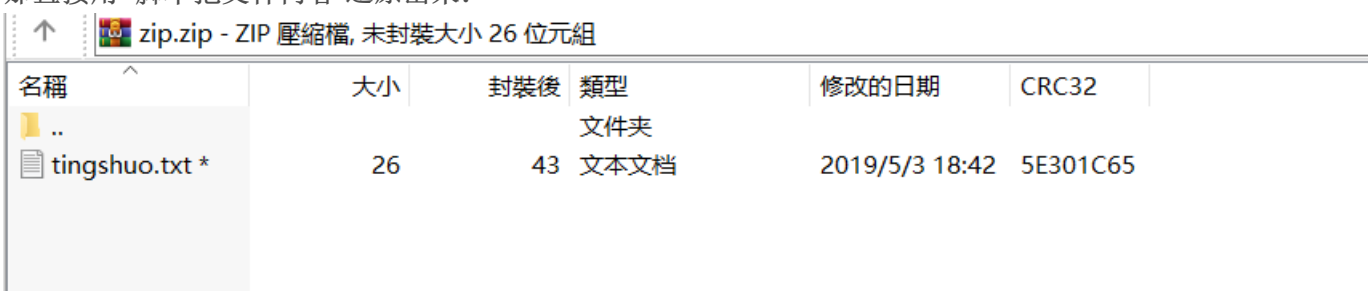
得到flag：

flag{nev3r_enc0de_t00_sm4ll_fil3_w1th_zip}

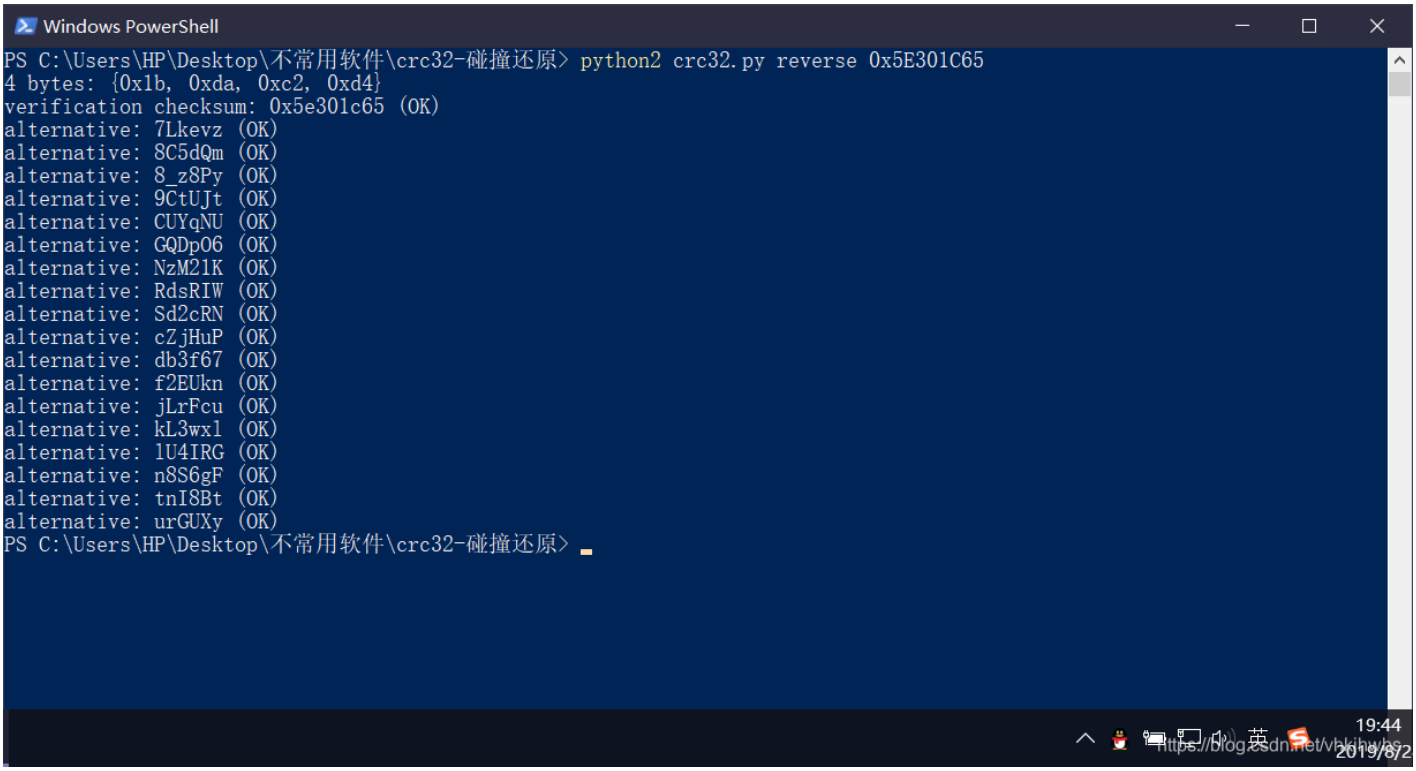
28 ， 不简单的压缩包

把 zip加上 zip后缀 后打开，文件有加密，发现有一段 crc32

那直接用　脚本把文件内容 还原出来：



上工具：https://github.com/theonlypwner/crc32



跑出来了，几个跟文件内容的冗余码相同的 几个字符串 然后 .....然后就没有然后了，不知道接下来怎么搞