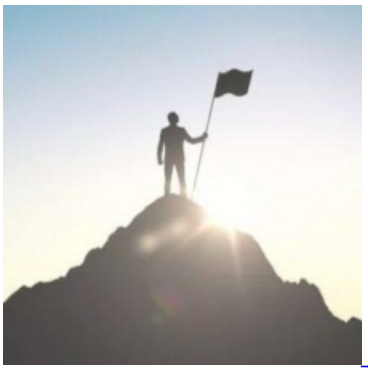
WhaleCTF 隐写篇~

原创

A_dmins 于 2019-08-22 14:48:47 发布 990 收藏 4

分类专栏： CTF题 WhaleCTF

本文链接：https://blog.csdn.net/qq_42967398/article/details/96971725



CTF题 同时被 2 个专栏收录

115 篇文章 11 订阅

订阅专栏

WhaleCTF

3 篇文章 0 订阅

订阅专栏

WhaleCTF 隐写篇~

emmm这好像是之前写的一篇wp，，，，，

忘记发了，记录一下吧~

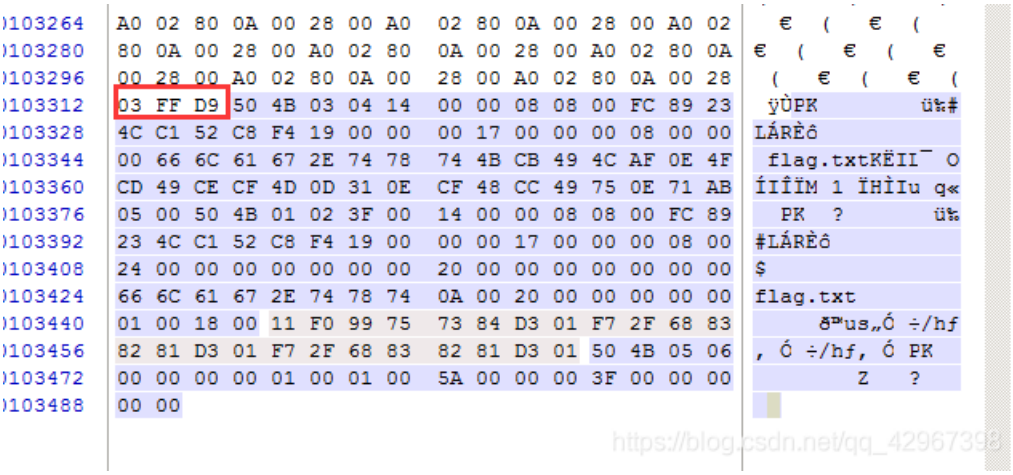
Find

图片另存为，使用stegsolve打开：

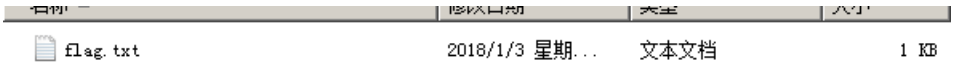
扫码可得flag

被我吃了

图片另存为，用winhex打开，由于是jpg搜索FFD9：



文末包含一个压缩包~
直接选择另存为一个新文件，解压可得flag:



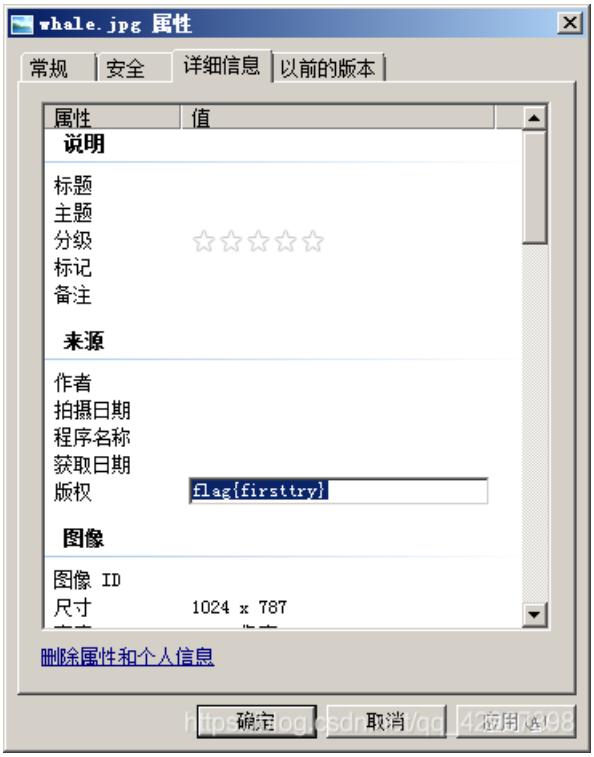
合体鲸鱼

根据题目意思大概知道是binwalk了
直接丢kali里面binwalk或者foremost，分解出另一张图片
打开可以看见flag：

flag{youfindmeWHALE}

亚种

直接图片另存为，查看属性可得flag：



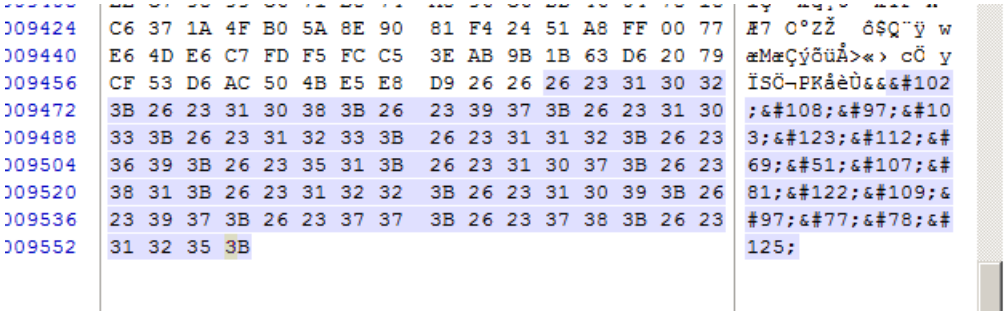
下雨天

图片另存，winhex查看为gif文件
该文件后缀名~丢进stegslove分解帧数可得：

这是什么

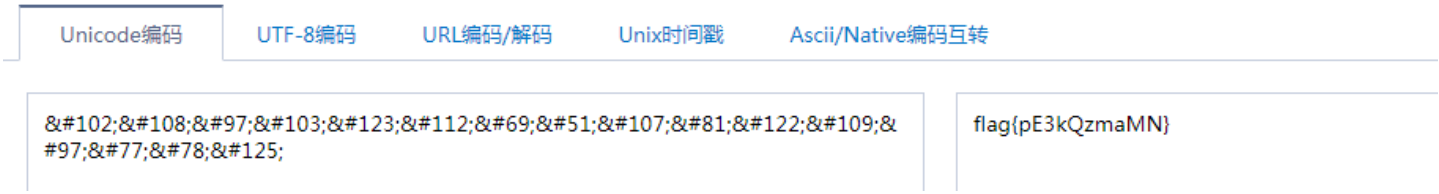
根据提示，兔子脚下~

丢进winhex发现最下面有东西：



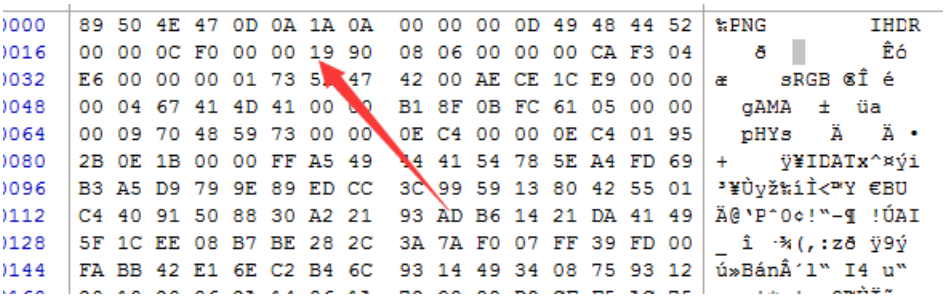
```
009424  C6 37 1A 4F B0 5A 8E 90  81 F4 24 51 A8 FF 00 77  Æ7 O°ZŽ  ô$Q¨ÿ w
009440  E6 4D E6 C7 FD F5 FC C5  3E AB 9B 1B 63 D6 20 79  æMæÇýõüÅ>«› cÖ y
009456  CF 53 D6 AC 50 4B E5 E8  D9 26 26 26 23 31 30 32  ÏSÖ¬PKåèÙ&&&#102
009472  3B 26 23 31 30 38 3B 26  23 39 37 3B 26 23 31 30  ;&#108;&#97;&#10
009488  33 3B 26 23 31 32 33 3B  26 23 31 31 32 3B 26 23  3;&#123;&#112;&#
009504  36 39 3B 26 23 35 31 3B  26 23 31 30 37 3B 26 23  69;&#51;&#107;&#
009520  38 31 3B 26 23 31 32 32  3B 26 23 31 30 39 3B 26  81;&#122;&#109;&
009536  23 39 37 3B 26 23 37 37  3B 26 23 37 38 3B 26 23  #97;&#77;&#78;&#
009552  31 32 35 3B                                       125;
```

直接进行Unicode转码，得到：



IHDR

根据提示IHDR，可能是修改高度，丢进winhex修改高度：

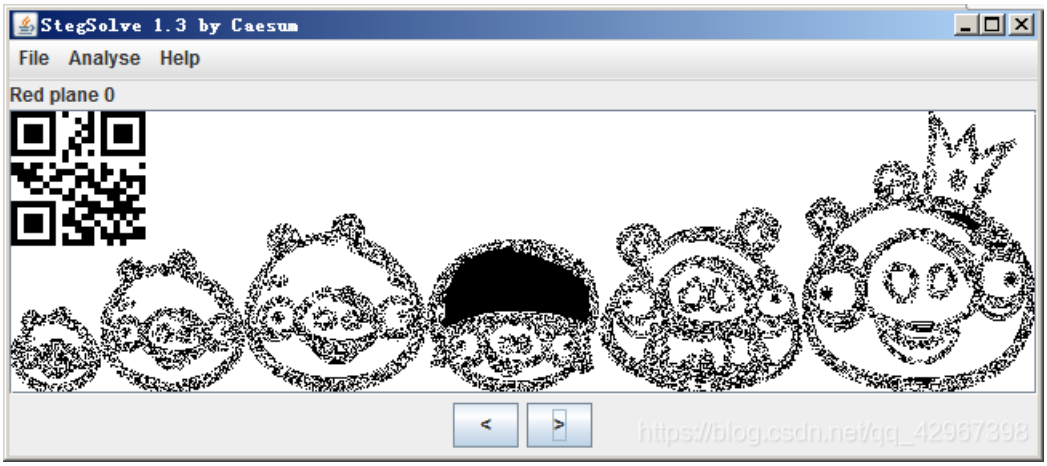


得到：

FLAG{ihDR_ALSO_FUN}

愤怒的小猪

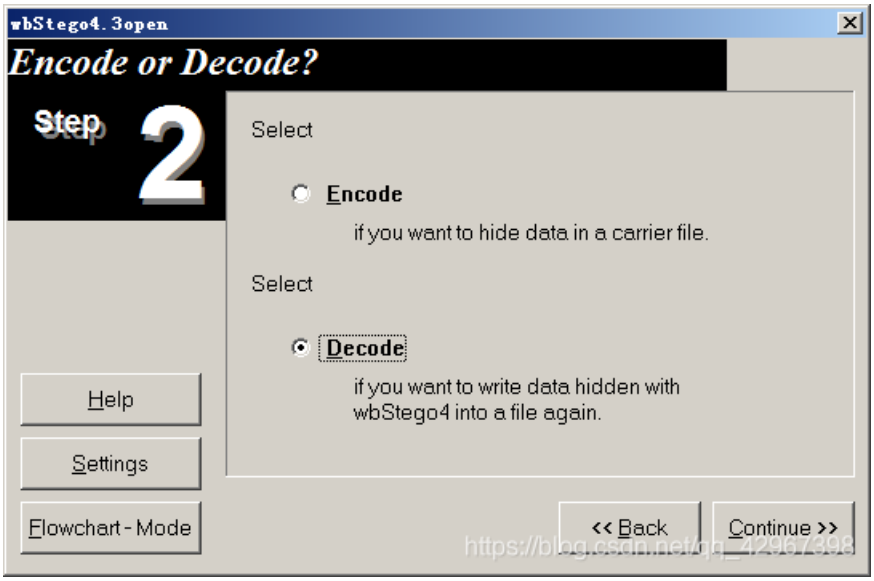
另存为，steg打开：



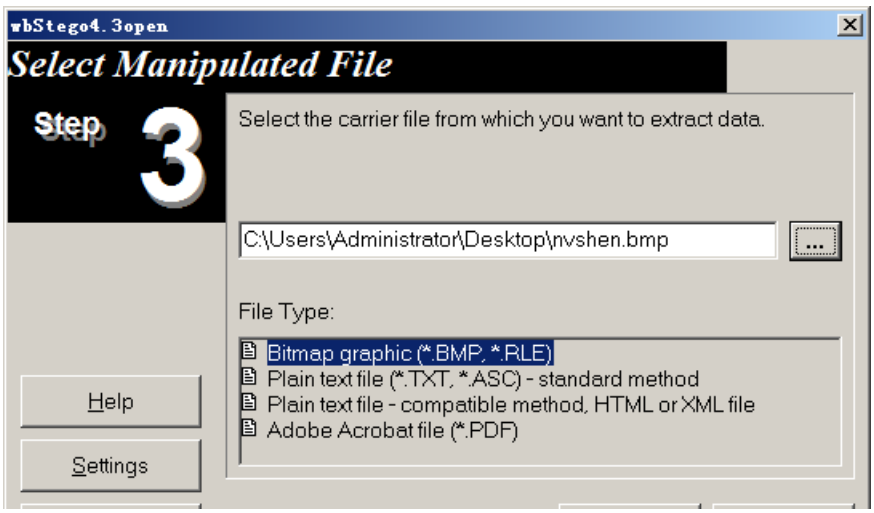
扫码可得flag!

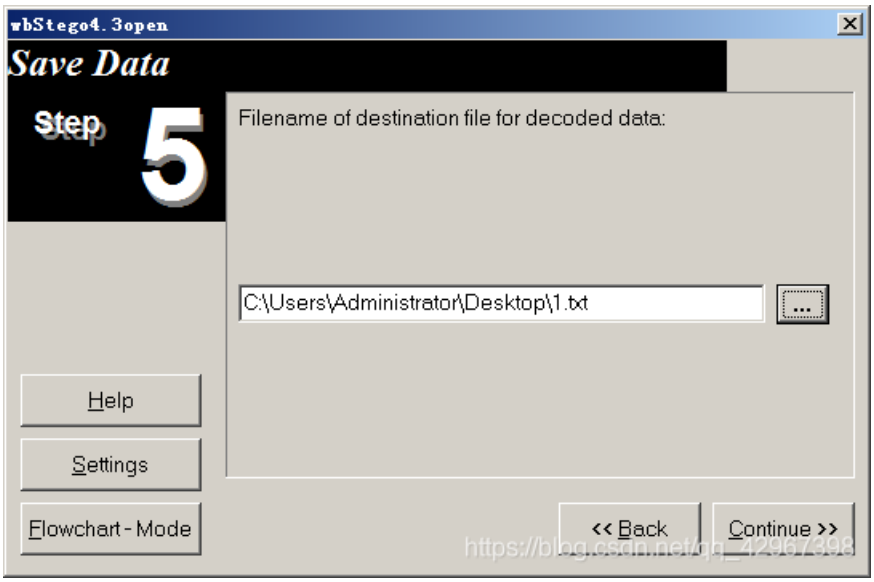
底片

这道题目有点意思~ bmp的隐写~
另存为winhex打开发现是bmp文件~
使用工具wbStego4.3open进行解密：

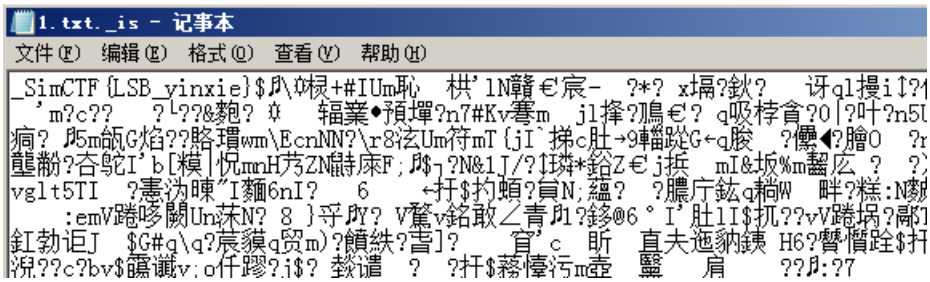


继续选择路径~





没有密码一直继续，保存文件自己选一个路径加文件名就好了
打开文本文档得到：



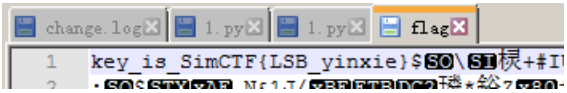
但是提交一直错误~~ 套他的猴子！！！！！
难道下划线前面还有东西！！！ 自己不会了，菜哭~~
查看了大佬的WP~~引用下大佬的话以及jio本：

由于图片是bmp类型，像素在图片中是倒叙存储，即以b,g,r的方式存储，而且当高度为正时图片中最后一行像素是存储在第一行的位置（对于正常二位坐标系，当以左下角为原点高度才会为正），用PIL的getdata读出的像素只是像素点在图片的相对位置，并非在文本文件的相对位置，所以以文本文件的方式读取图片

附上脚本python2：

```
bmpfi=open("nvshen.bmp","rb")
bmpstr=bmpfi.read()
bmpfi.close()

bfOffBits=int(bmpstr[13:9:-1].encode("hex"),16)
str1=""
for j in xrange(bfOffBits,len(bmpstr)):
 str1+=bin(ord(bmpstr[j]))[-1]
i=0
lst=""
while i<len(str1):
 str2=str1[i:i+8]
 lst+=chr(int(str2,2))
 i=i+8

fi=open("flag",'wb')
fi.write(lst)
fi.close()
```

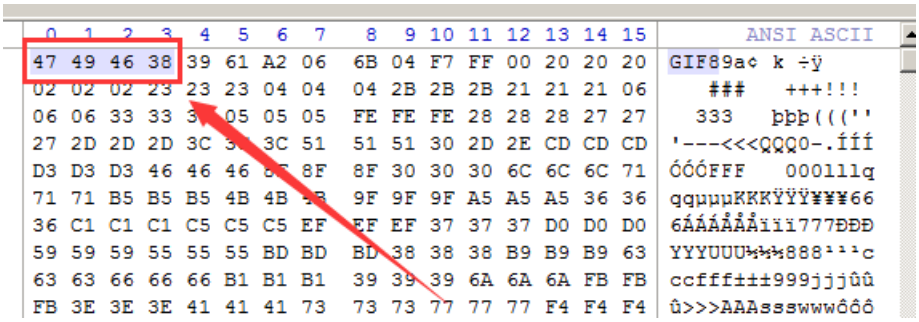
结果：



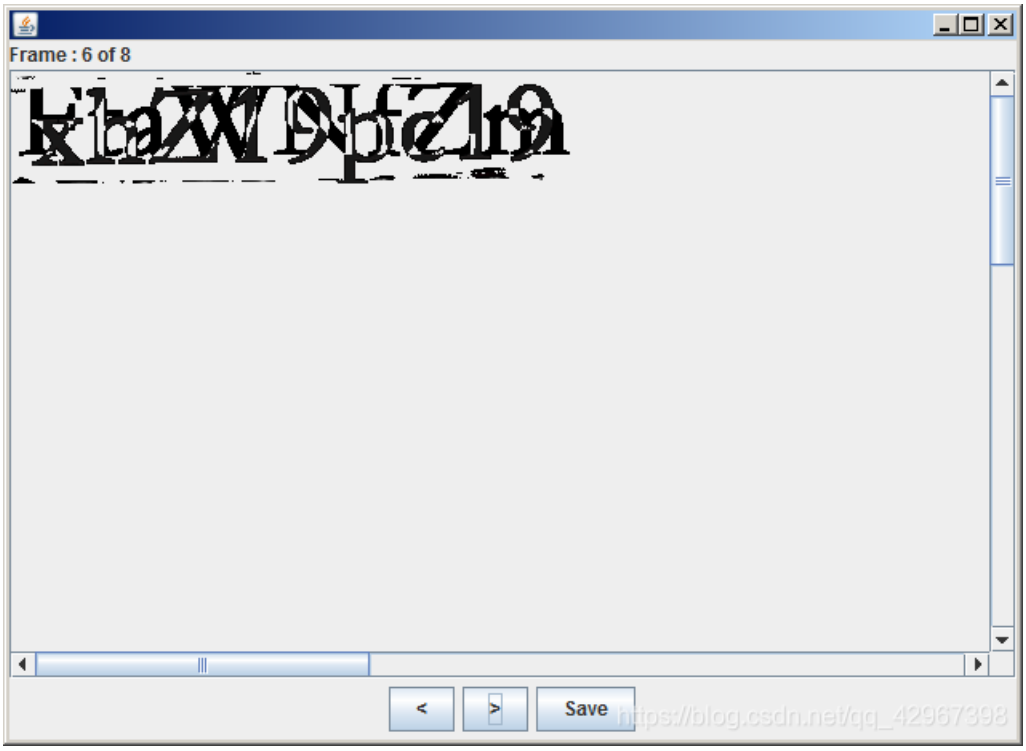
好菜啊！！！！

真是动图

用winhex打开发现缺少文件头，把文件头添加上去：



然后变成GIF图片了，直接使用steg进行查看
由于有些字符可能有点看不清，须认真查看：



得到字符串：`Y2F0Y2hfdGhlX2R5bmFtaWNfZmxhZ19pc19xdWl0ZV9zaW1wbGU=`
很明显是base64解密，得到：

模糊的图片

下载图片，模模糊糊好像能够看见一些字：

通过steg进行一些操作查看之后~~

ps又不会，，，，
模模糊糊才能看出点什么，，，，At10lSCC4_1Z__P_
还有几个字符不知道，，，，，
最后看了别人用PS调出来的图片，发现结果是：At10lSCC421ZLAPL
看了大神WP，发现还可以使用脚本跑出来>>>>>>>>抱歉真不会，，，，

借用大神脚本：

```
#coding:utf-8
import Image
img = Image.open('1.png')
X = img.size[0]
Y = img.size[1]
#print X,Y
for i in range(X-2):
 for j in range(Y-2):
   a = img.getpixel((i,j))[0]+img.getpixel((i,j))[1]+img.getpixel((i,j))[2]
   b = img.getpixel((i,j+1))[0]+img.getpixel((i,j+1))[1]+img.getpixel((i,j+1))[2]
   c = img.getpixel((i,j+2))[0]+img.getpixel((i,j+2))[1]+img.getpixel((i,j+2))[2]
   if (a > b and c > b) or (a < b and c < b):
     pass
   else:
     img.putpixel((i,j),(255,255,255))
img.show()
```

好像说运行完也能看见，，，，本地python2的PIL模块安装失败，
这个脚本就未进行验证了~ 也不知道行不行，就在这里先留个坑吧~~

错误压缩

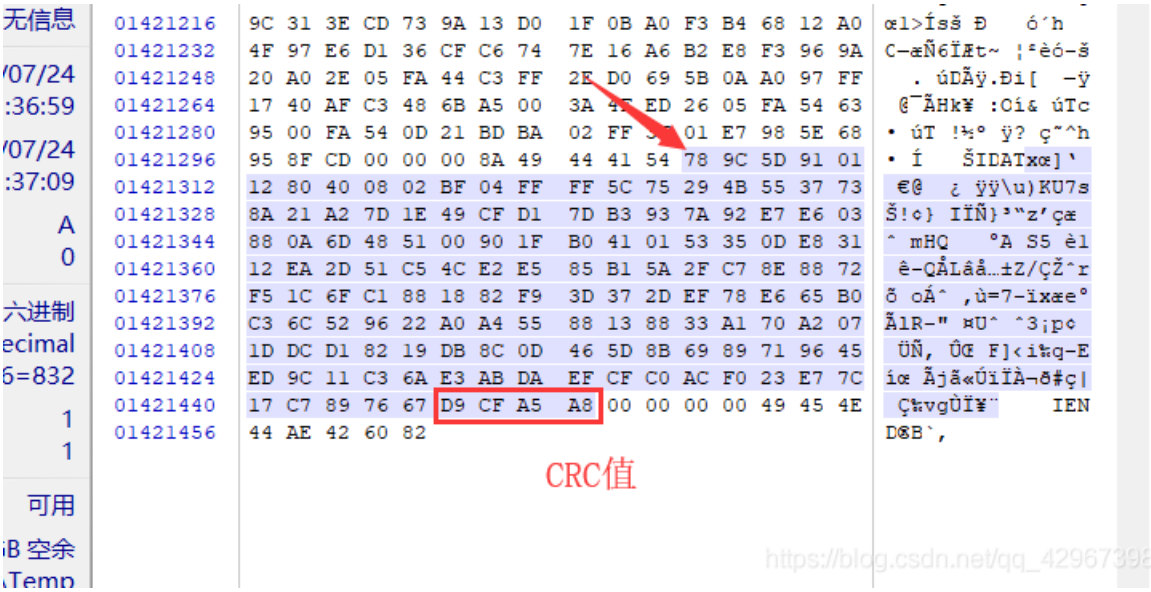
是png图片，一套steg常规操作，，，并没有发现什么东西~
放进kali里面进行binwalk~~ zlib压缩数据有两个？

用tweakpng查看一下png图片的详细信息：

Chunk	Length	CRC	Attributes	Contents
IHDR	13	5871e019	critical	PNG image header: 1000×562, 8 bits/sample, truecolor+alpha, noninterlaced
sRGB	1	aece1ce9	ancillary, unsafe to...	sRGB color space, rendering intent: Perceptual
gAMA	4	0bfc6105	ancillary, unsafe to...	file gamma = 0.45455
pHYs	9	952b0e1b	ancillary, safe to c...	pixel size = 3780×3780 pixels per meter (96.0×96.0 dpi)
IDAT	65445	3c52e386	critical	PNG image data
IDAT	65524	21a250d6	critical	PNG image data
IDAT	65524	dd582fbe	critical	PNG image data
IDAT	65524	939f6ecf	critical	PNG image data
IDAT	65524	cc5b8b36	critical	PNG image data
IDAT	65524	34e41cee	critical	PNG image data
IDAT	65524	526d60fe	critical	PNG image data
IDAT	65524	e5c2ad0c	critical	PNG image data
IDAT	65524	7c5eafb4	critical	PNG image data
IDAT	65524	87f6163d	critical	PNG image data
IDAT	65524	00a5e59f	critical	PNG image data
IDAT	65524	0df4a0fa	critical	PNG image data
IDAT	65524	2ab5b183	critical	PNG image data
IDAT	65524	b07349a3	critical	PNG image data

IDAT	65524	41bd8f1f	critical	PNG image data
IDAT	65524	251f6df9	critical	PNG image data
IDAT	65524	8787049d	critical	PNG image data
IDAT	65524	c1cf4fce	critical	PNG image data
IDAT	65524	6dd6c304	critical	PNG image data
IDAT	65524	c5bd9fb1	critical	PNG image data
IDAT	65524	1e255491	critical	PNG image data
IDAT	45027	68958fcd	critical	PNG image data
IDAT	138	d9cfa5a8	critical	PNG image data
IEND	0	ae426082	critical	end-of-image marker

貌似是存在问题的，前面都是65524，后面一个45027，后面还有一个138？
这个138为什么不放到未满的45027里面？？？这138个数据有问题！
winhex走一波：



取出CRC值前面的138个值，，，，，使用zlib模块
脚本：

```
import zlib

s = open('sctf.png', 'rb').read()[0x15AFFB:0x15B085]
data = zlib.decompress(s)
binstr = str(data)
print(binstr)
```

得到binstr:

```
11111110001000011011111111000001011100101101000001101110101000000000101110110111010010000000010111011011101011101101001011101100000101010110110100000111111110101010101011111110000000010111011100000000111010011000001010011101101111010101001000011100000000000101000000001001001101000100111001111011100111100001110111111000110010100011001110001010001101000111101011000001010001011000001101110110010000111001110010000101111111010000000011010100100011110111111011100001101011011100000100001100110001111010111010001101001111100001011101011000111010011100101110100100111011011000110000010110001101000110001111111011010110111011001
```

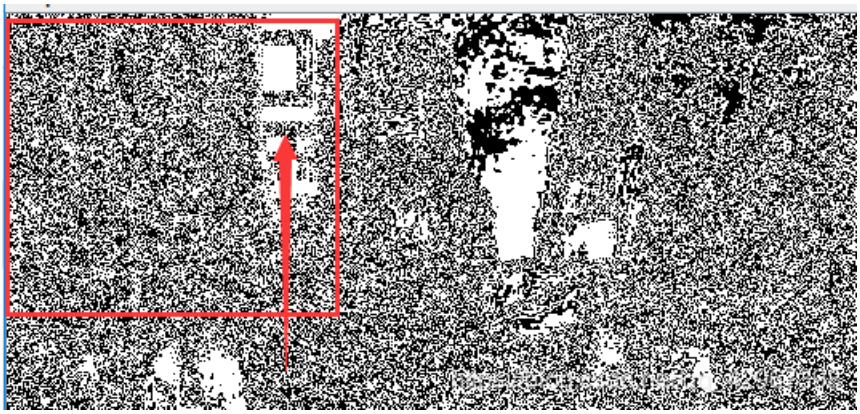
得到一串01字符串~
一般CTF中01要么是ASCII码，当然这里肯定不可能了
要么就是二维码的01~ 查看一下长度为625，刚好是25的平方
直接进行转图片~
脚本：

```
from PIL import Image

s = "11111110001000011011111111000001011100101101000001101110101000000000101110110111010010000000010111011011101
01110110100101110110000010101011011010000011111111010101010111111100000000101110111000000001101001100000101001
11011011110101010010000111000000000001010000000010010011010001001110011111011100111100001110111110001100101000110
01110000101010001101000111101011000001010001011000001101110110010000111001110010000101111111010000000011010100100
00111101111111011100001101011011100000100001100110001111010111010001101001111100001011101011000111010011100101111
01001001110110110001100000101100011010001100011111111011010110111011011"

x = 25   #width    #x坐标  通过对txt里的行数进行整数分解
y = 25    #height    #y坐标  x * y = 行数
im = Image.new("RGB", (x, y))   #创建图片

k = 0
for i in range(0, x):
 for j in range(0, y):
  if(s[k] == '0'):
   im.putpixel((i, j), (255,255,255))    #将rgb转化为像素
  else:
   im.putpixel((i, j), (0,0,0))
  k += 1

im.save("flag.jpg")   #im.save('flag.jpg')保存为jpg图片
```

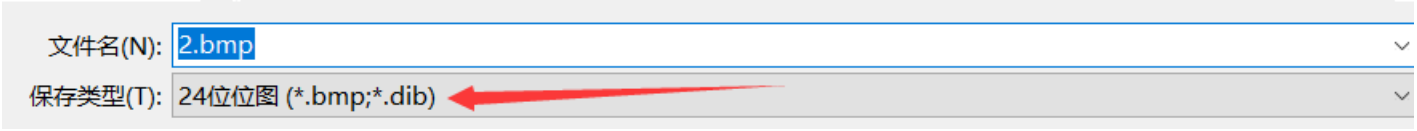
得到一张贼小的图片，，，，，放到扫码可得~~

最低位的亲吻

看题目名字就觉得是lsb的隐写~
直接丢进steg：



是不是觉得很像一个二维码~~这里肯定有问题
这里用的是一个骚操作~~~ 直接将原图用画图打开，另存为：



这里选择24位~ 保存下来，再用steg打开这个新的图片，get：