

# WhaleCTF 隐写篇~

原创

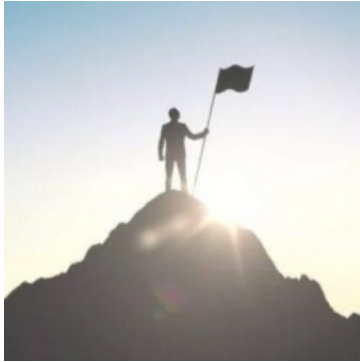
[A\\_dmins](#) 于 2019-08-22 14:48:47 发布 990 收藏 4

分类专栏: [CTF题 WhaleCTF](#)

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/qq\\_42967398/article/details/96971725](https://blog.csdn.net/qq_42967398/article/details/96971725)

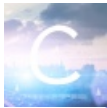
版权



[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~

[Find](#)

[被我吃了](#)

[合体鲸鱼](#)

[亚种](#)

[下雨天](#)

[这是什么](#)

[IHDR](#)

[愤怒的小猪](#)

[底片](#)

[真是动图](#)

[模糊的图片](#)

[错误压缩](#)

[最低位的亲吻](#)

emmm这好像是之前写的一篇wp, , , , ,  
忘记发了, 记录一下吧~

## Find

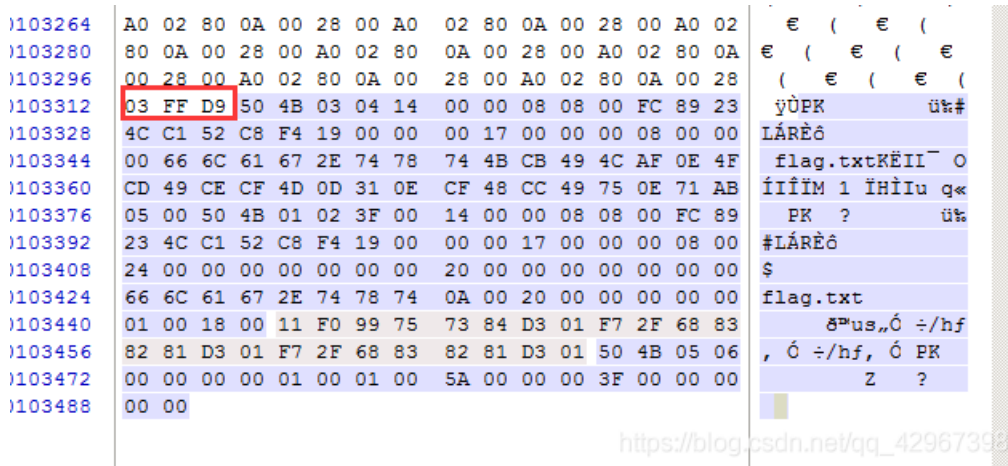
图片另存为，使用stegsolve打开：



扫码可得flag

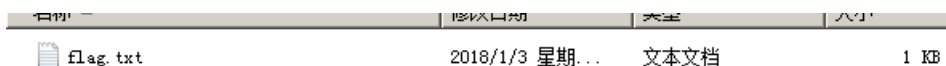
## 被我吃了

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

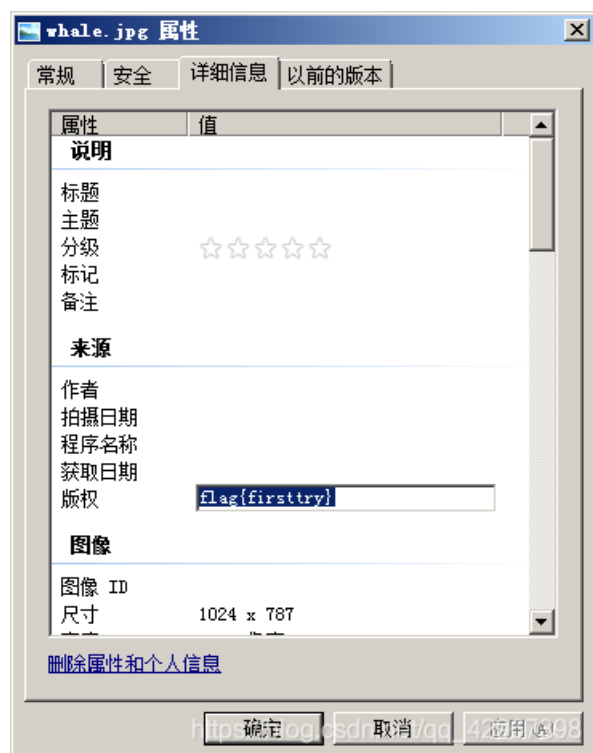
打开可以看见flag:

**flag{youfindmeWHALE}**

[https://blog.csdn.net/qq\\_42967398](https://blog.csdn.net/qq_42967398)

## 亚种

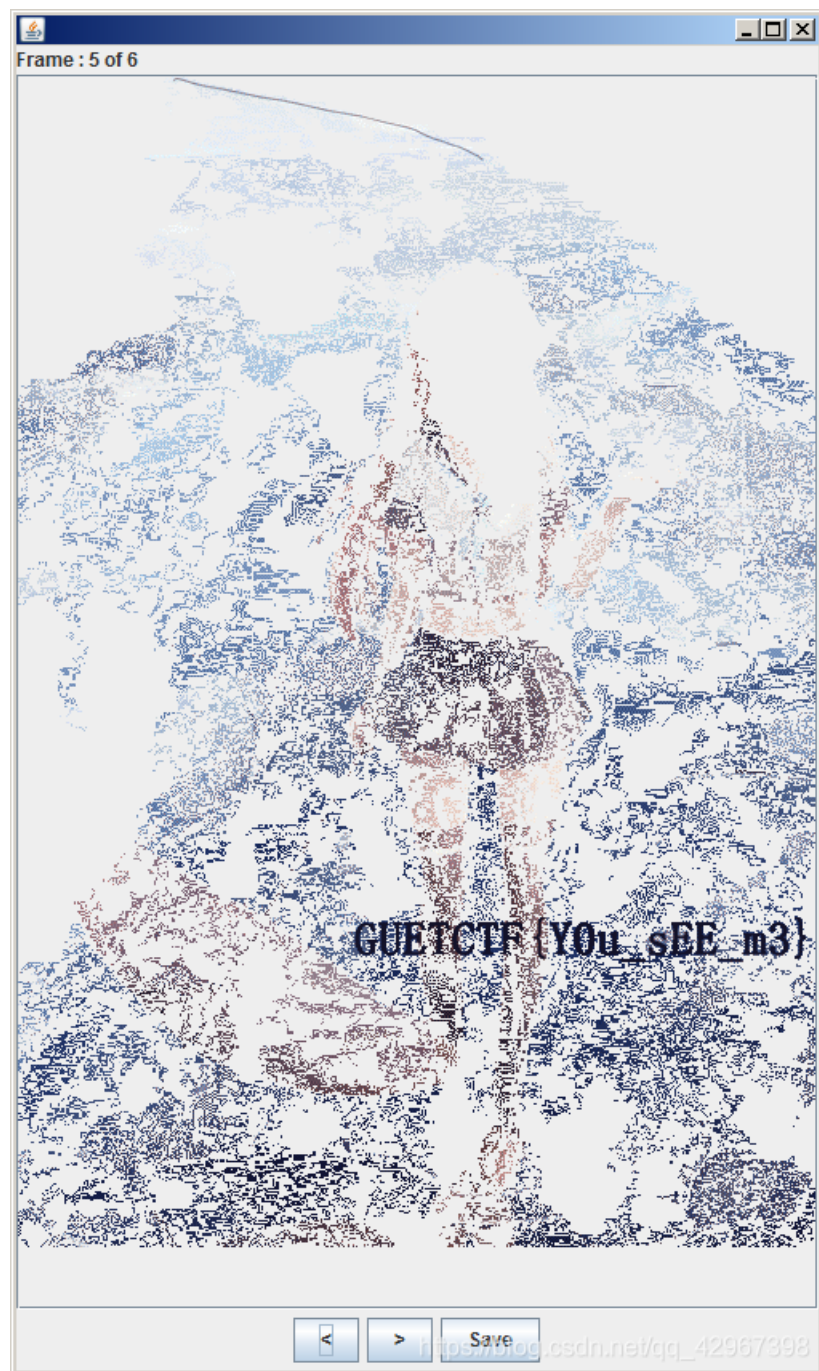
直接图片另存为，查看属性可得flag:



## 下雨天

图片另存，winhex查看为gif文件

该文件后缀名~丢进stegslope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 77 0°ZŽ 0$Q`y w
009440 E6 4D E6 C7 FD F5 FC C5 3E AB 9B 1B 63 D6 20 79 æMæÇýðüÄ>«> cÖ y
009456 CF 53 D6 AC 50 4B E5 E8 D9 26 26 26 23 31 30 32 ĪSÖ~PKâèÛ&&&#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：

Unicode编码	UTF-8编码	URL编码/解码	Unix时间戳	Ascii/Native编码互转
&#102;&#108;&#97;&#103;&#123;&#112;&#112;&#69;&#51;&#107;&#81;&#122;&#109;&#97;&#77;&#78;&#125;				flag{pE3kQzmaMN}

IHDR

根据提示IHDR，可能是修改高度，丢进winhex修改高度：

0000	89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52	%PNG IHDR
0016	00 00 0C F0 00 00 19 90 08 06 00 00 00 CA F3 04	δ █ Èó
0032	E6 00 00 00 01 73 51 47 42 00 AE CE 1C E9 00 00	æ sRGB @İ é
0048	00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00	gAMA ± ũa
0064	00 09 70 48 59 73 00 00 0E C4 00 00 0E C4 01 95	pHYs Ä Ä •
0080	2B 0E 1B 00 00 FF A5 49 44 41 54 78 5E A4 FD 69	+ ŷŸIDATx^ŷı
0096	B3 A5 D9 79 9E 89 ED CC 3C 99 59 13 80 42 55 01	*ŷÛyžŷıİ<ŷY €BU
0112	C4 40 91 50 88 30 A2 21 93 AD B6 14 21 DA 41 49	Ä@'P^Oc!""-ŷ !ÚAI
0128	5F 1C EE 08 B7 BE 28 2C 3A 7A F0 07 FF 39 FD 00	_ İ ŷ(, :zδ ŷ9ŷ
0144	FA BB 42 E1 6E C2 B4 6C 93 14 49 34 08 75 93 12	ú»BánÄ'l" I4 u"

得到：

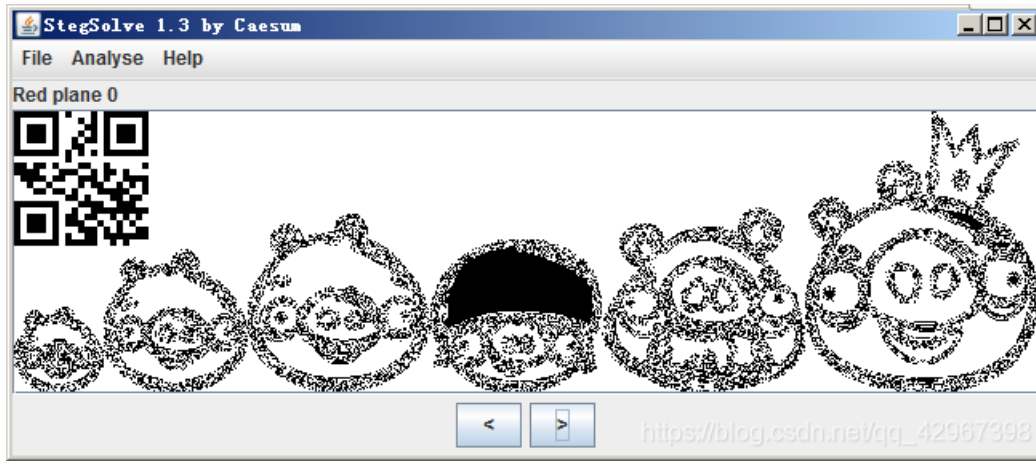


[https://blog.csdn.net/qq\\_41410961/article/details/10410961](https://blog.csdn.net/qq_41410961/article/details/10410961)

愤怒的小猪



另存为，steg打开：



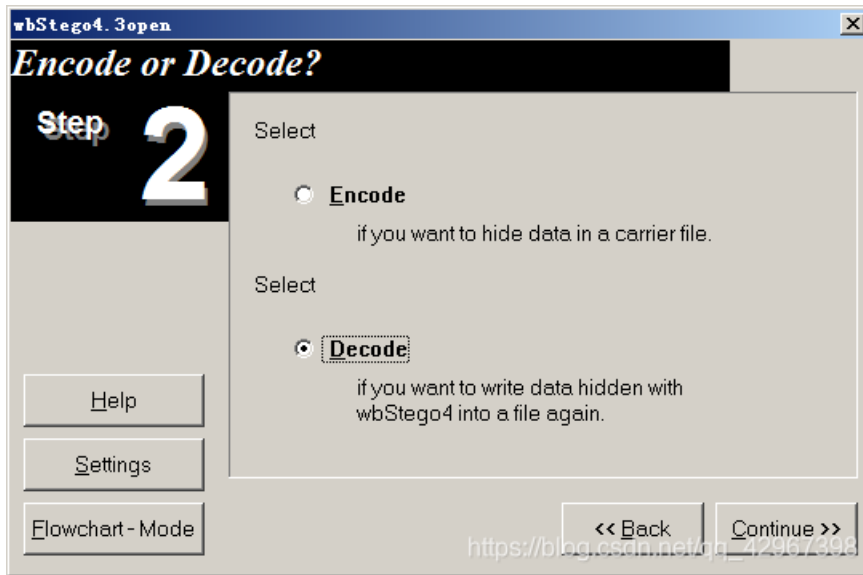
扫码可得flag!

## 底片

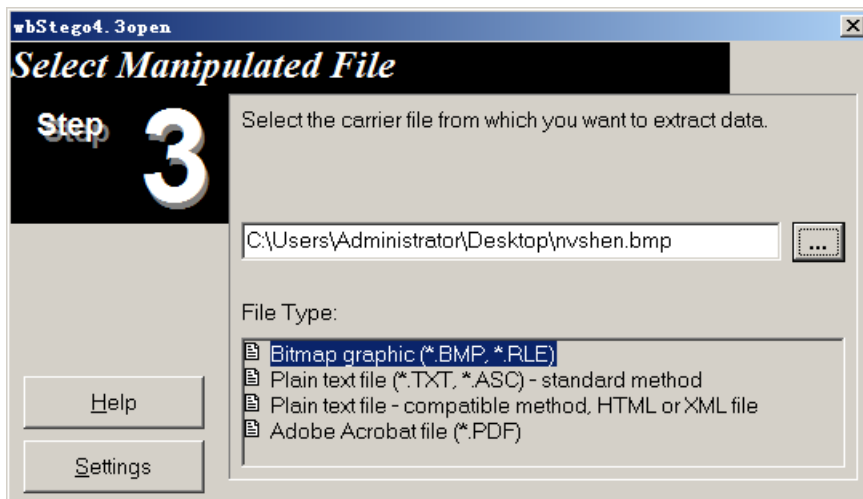
这道题目有点意思~ bmp的隐写~

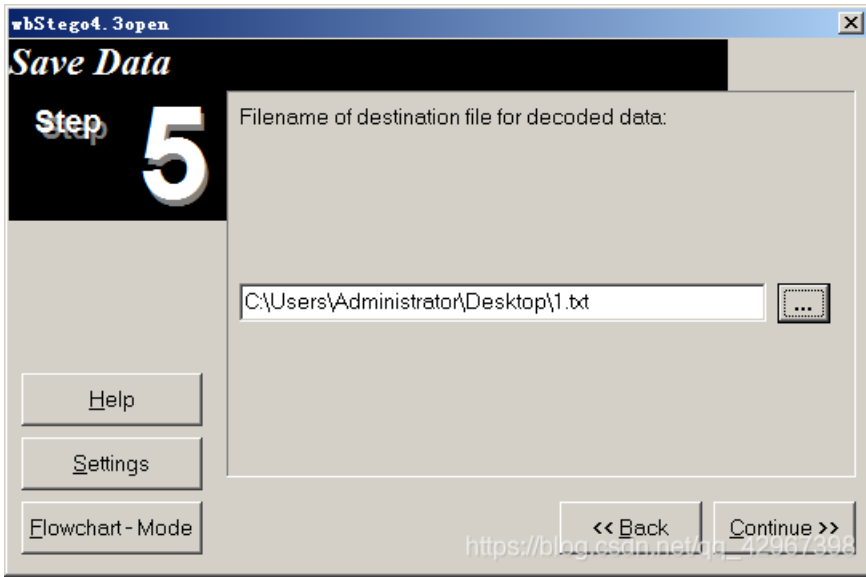
另存为winhex打开发现是bmp文件~

使用工具wbStego4.3open进行解密：

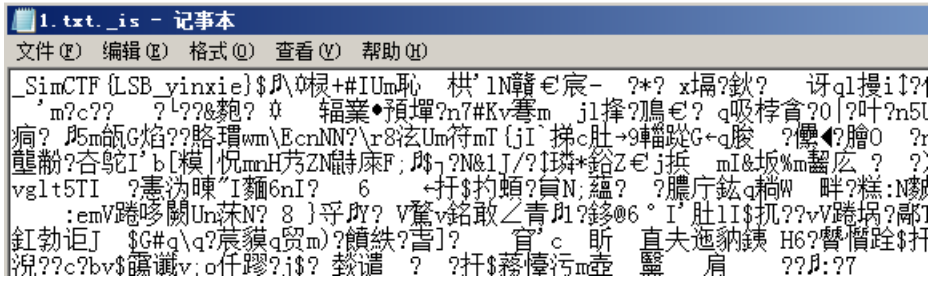


继续选择路径~





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

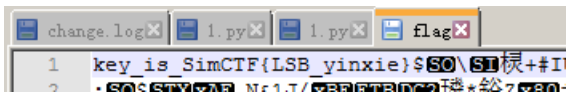
附上脚本python2:

```
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()
```

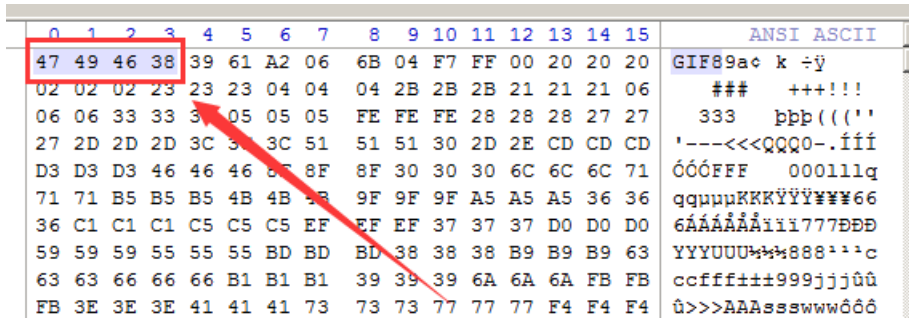
结果:



好菜啊!!!

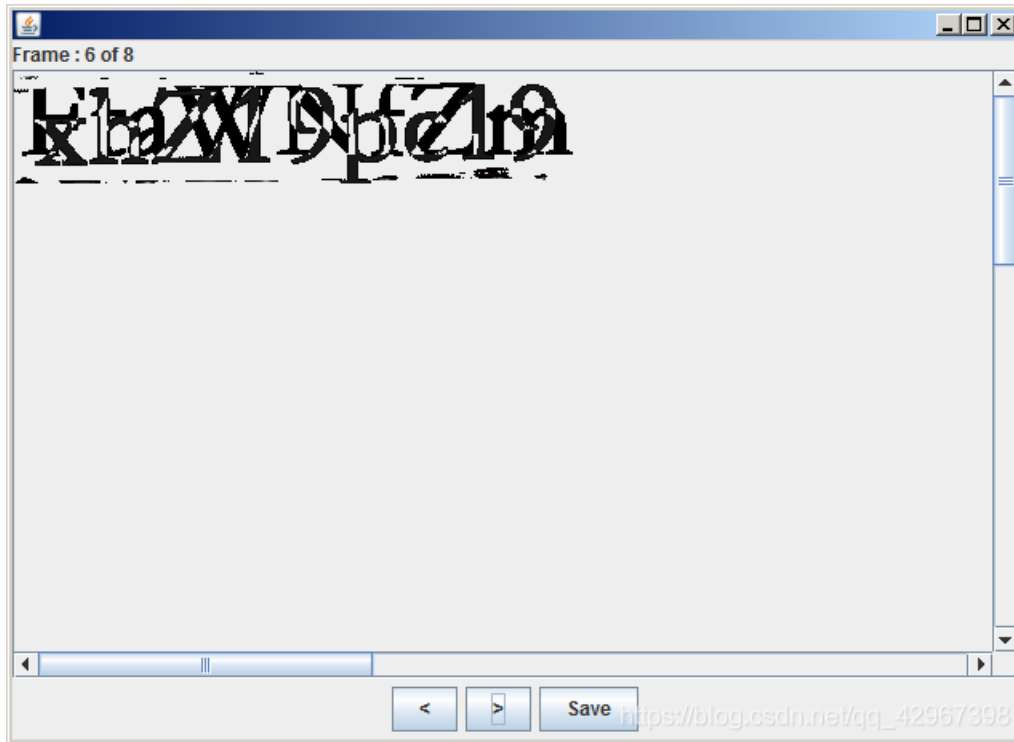
真是动图

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：`Y2F0Y2hfdGhIX2R5bmFtaWNfZmxhZ19pc19xdWl0ZV9zaW1wbGU=`

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，，，， At10ISCC4\_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：At10ISCC421ZLAPL

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: 1000x562, 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 = 3780x3780 pixels per meter (96.0x96.0 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

[https://blog.csdn.net/qq\\_42967398](https://blog.csdn.net/qq_42967398)

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为什么不放到未读的45027里面？？？这138个数据有问题！

winhex走一波：

无信息

07/24 11:36:59

07/24 11:37:09

A 0

六进制  
decimal  
6=832

1  
1

可用

KB 空余

Temp

01421216	9C 31 3E CD 73 9A 13 D0 1F 0B A0 F3 B4 68 12 A0	αl>îsš Đ ó'h
01421232	4F 97 E6 D1 36 CF C6 74 7E 16 A6 B2 E8 F3 96 9A	C-æÑôïÅt~ !'èó-š
01421248	20 A0 2E 05 FA 44 C3 FF 2E DO 69 5B 0A A0 97 FF	. úDÃÿ.Đi[ -ÿ
01421264	17 40 AF C3 48 6B A5 00 3A 4F ED 26 05 FA 54 63	@-ÅHk¥ :Oi& úTc
01421280	95 00 FA 54 0D 21 BD BA 02 FF 00 01 E7 98 5E 68	• út !%° ÿ? ç^^h
01421296	95 8F CD 00 00 00 8A 49 44 41 54 78 9C 5D 91 01	• í ŠIDATxα]`
01421312	12 80 40 08 02 BF 04 FF FF 5C 75 29 4B 55 37 73	€@ ç ÿÿ\u)KU7s
01421328	8A 21 A2 7D 1E 49 CF D1 7D B3 93 7A 92 E7 E6 03	Š!c} IiÑ)'z'çæ
01421344	88 0A 6D 48 51 00 90 1F B0 41 01 53 35 0D E8 31	^ mHQ °A S5 èl
01421360	12 EA 2D 51 C5 4C E2 E5 85 B1 5A 2F C7 8E 88 72	è-QÅLåã...±Z/ÇŽ~r
01421376	F5 1C 6F C1 88 18 82 F9 3D 37 2D EF 78 E6 65 B0	ð oÁ^ ,ù=7-ixæ°
01421392	C3 6C 52 96 22 A0 A4 55 88 13 88 33 A1 70 A2 07	ÄlR-" «U^ ^3;pç
01421408	1D DC D1 82 19 DB 8C 0D 46 5D 8B 69 89 71 96 45	ÜÑ, ÚÇ F <ixq-E
01421424	ED 9C 11 C3 6A E3 AB DA EF CF C0 AC F0 23 E7 7C	iα Åjã«ÜiïÄ-ð#ç
01421440	17 C7 89 76 67 D9 CF A5 A8 00 00 00 00 49 45 4E	ÇkvgÜi¥" IEN
01421456	44 AE 42 60 82	D&B` ,

CRC值

https://blog.csdn.net/qq\_42967398

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:

```
111111100010000110111111110000010111001011010000011011101010000000010111011011101001000000001011101101101101110
110100101110110000010101011011010000011111111010101010101111110000000101110111000000010100110000010100111011
01111010101001000011100000000001010000000100100110100010011100111101110011110000111011111000110010100011001110
00010101000110100011110101100000101000101100000110111011001000010111111010000000110101001000111
101111110111000011010110111000001000011001100011110101110100011010011111000010111010110001110100111010011101001
00111011011000110000010110001101000111111101010110110110111111011011000111010110001110100111010011101001
```

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 = "1111111000100001101111111100000101110010110100000110111010100000000101110110111010010000000010111011011101
01110110100101110110000010101011011011010000011111111010101010111111100000000101110111000000001101001100000101001
1101101111010101001000111000000000010100000001001001101000100111001111011100111100001110111110001100101000110
01110000101010001101000111101011000001010001011000001101110110010000111001110010000101111110100000001101010010
00111101111111011100001101011011100000100001100110001111010111010001101001111010001111010111010001110100111010111
01001001110110110001100000101100011010001100011111110110110111011011"

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

```

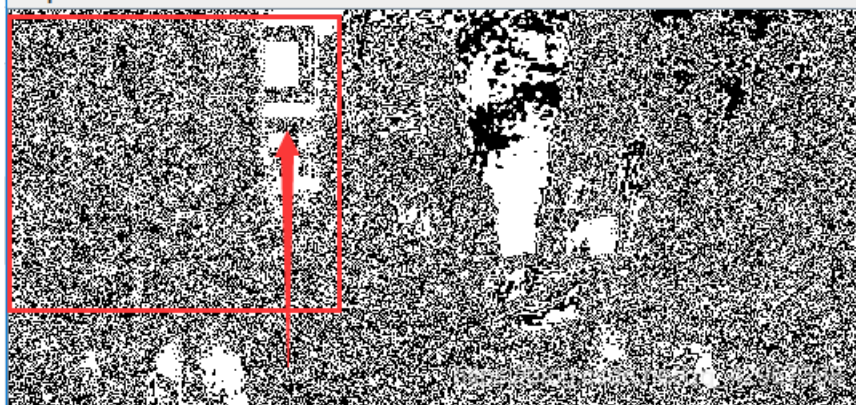
得到一张贼小的图片，，，，，放到扫码可得~~



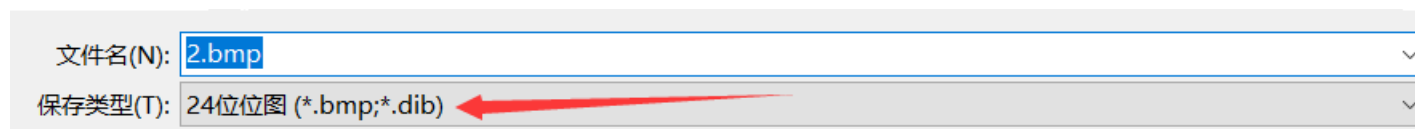
最低位的亲吻



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:

