

2021涅普冬令营_wp_(二)

原创

[Nebula1805](#)  已于 2022-04-07 21:57:54 修改  1889  收藏 3

分类专栏: [涅普冬令营学习笔记](#) 文章标签: [反编译](#) [字符串](#) [python](#) [信息安全](#) [cmd](#)

于 2021-02-24 21:16:08 首次发布

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/Nebula1805/article/details/113943879>

版权



[涅普冬令营学习笔记](#) 专栏收录该内容

7 篇文章 0 订阅

订阅专栏

[NepNep Winter-CAMP](#)

1.GIF图片隐写



提示此文件为gif图片，用010editor打开，发现文件头不对，

```
00h: 39 61 A2 06 6B 04 F7 FF 00 20 20 20 02 02 02 23 9aÇ.k.÷ÿ. ...#
010h: 23 23 04 04 04 2B 2B 2B 21 21 21 06 06 06 33 33 ##...+++!!!...33
020h: 33 05 05 05 FE FE FE 28 28 28 27 27 27 2D 2D 2D 3...bbb((((''--
```

应为 47 49 46 38 ,添加文件头，保存，得到GIF图片，GIF图片中有“password is ……”格式闪过



用stegslope工具打开，analyse->frame browser,查看每一帧，frame 3-8:





字母重叠的图片，再用stegsolve工具单独打开，切换通道查看，



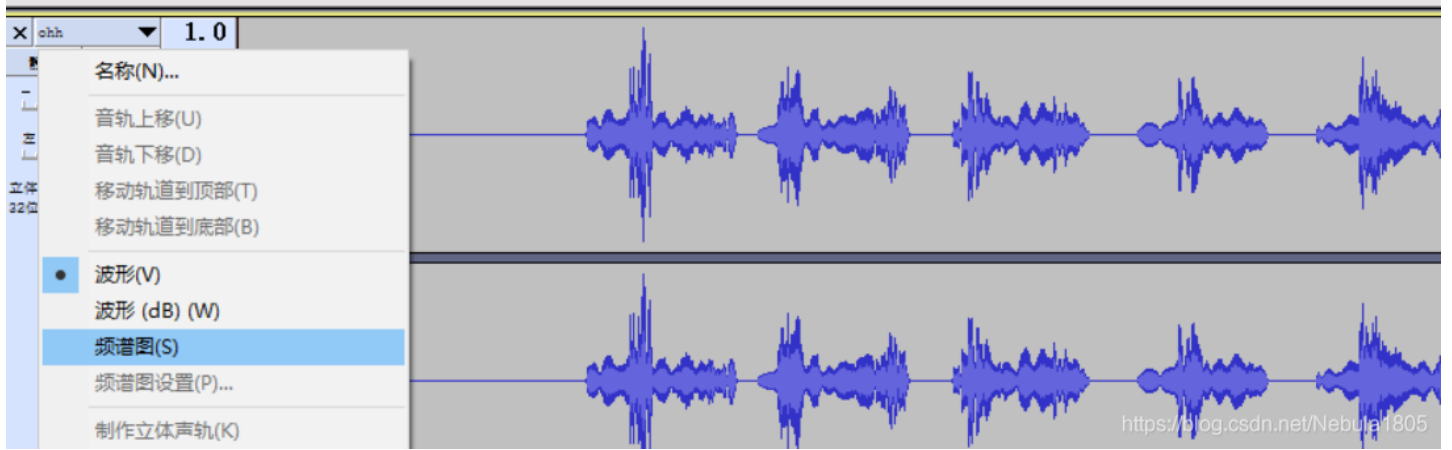
得到一些字母和数字：`Y2F0Y2hfd Gh1X2R5bm FtaWNfZm xhZ19pc19 xdW10ZV9z aW1wbGU=`，

观察特点，为base64编码，解码得 `catch_the_dynamic_flag_is_qumte_simple`，则flag为：

`flag{catch_the_dynamic_flag_is_qumte_simple}`，提交，发现flag错误，翻译下flag内容，wt? 将qumte换成quite。

2. 音频频谱隐写

得到一个ohh.wav文件，使用Audacity工具打开，Audacity工具使用
点击倒三角标识，切换频谱图，找到flag

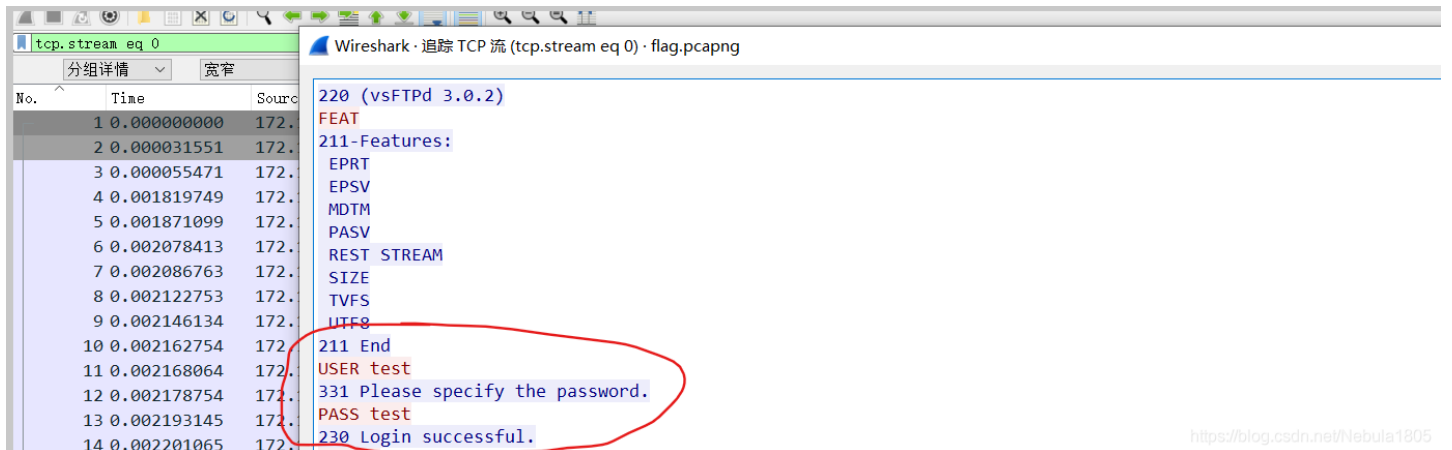


按住ctrl，滚动鼠标滑轮，放大图片,查看flag: `fbctf{This_1s_a_message}`

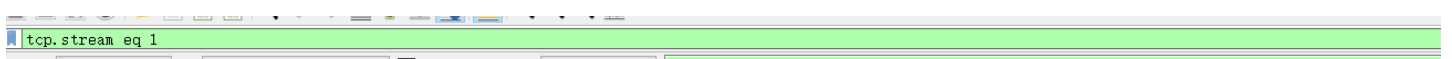


3.流量分析(一)

直接查找flag无果，发现这是FTP的流量包,追踪一下tcp流，用户test，密码test



则flag信息可能存在于ftp传输数据中，搜索ftp-data，追踪一下tcp流，发现曾执行郭过ls命令，flag可能存在于txt或者png文件中



No.	Time	Source	Destination	Protocol	Length	Info
53	0.437534412	172.17.0.1	172.17.0.2	TCP	74	36626 → 21108 [SYN] Seq=0 Win=29200 Len=0 MSS=1460
54	0.437553362	172.17.0.2	172.17.0.1	TCP	74	21108 → 36626 [ACK] Seq=1 Ack=0 Win=0 Len=0
55	0.437575942	172.17.0.1	172.17.0.2	TCP	66	36626 → 21106 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=616959586 TSecr=2980003694
58	0.438140641	172.17.0.2	172.17.0.1	FTP	264	Sep 19 07:52 .
59	0.438165181	172.17.0.1	172.17.0.2	FTP	264	Sep 19 07:52 ..
60	0.438191912	172.17.0.2	172.17.0.1	FTP	41	Sep 19 07:52 flag.txt
61	0.438604788	172.17.0.1	172.17.0.2	FTP	1133535	Sep 19 07:51 universe.png

搜索flag.txt, 并逐个追踪tcp流, 发现可疑字符串, 猜测是base64编码, 拿去解码

No.	Time	Source	Destination	Protocol	Length	Info
417	46.665039278	172.17.0.1	172.17.0.2	TCP	74	21106 → 48566 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=616959586 TSecr=0
418	46.665059039	172.17.0.2	172.17.0.1	TCP	74	48566 → 21106 [ACK] Seq=1 Ack=0 Win=0 Len=0
419	46.665081669	172.17.0.1	172.17.0.2	TCP	66	21106 → 48566 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=616959586 TSecr=2980003694
422	46.665564614	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
423	46.665578145	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
424	46.665600535	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
425	46.666062180	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)

假的。。。。。

```
flag{This is fake flag hahaha}
```

再搜索一下universe.png文件, 追踪一下tcp流,

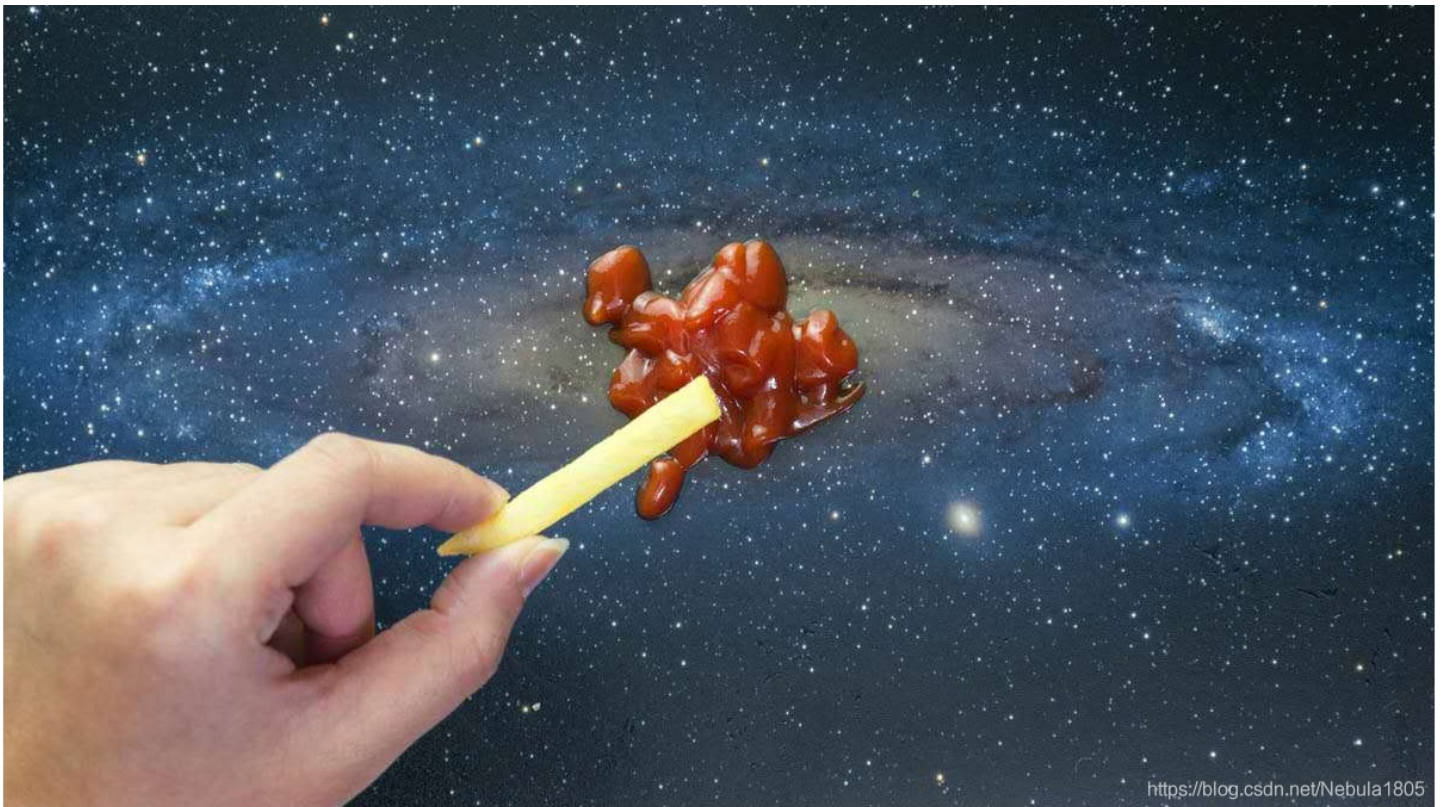
```
RETR /universe.png
150 Opening BINARY mode data connection for /universe.png (1133535 bytes).
226 Transfer complete.
PASV
227 Entering Passive Mode (172,17,0,2,82,115).
RETR /universe.png
150 Opening BINARY mode data connection for /universe.png (1133535 bytes).
426 Failure writing network stream.
```

猜测可能想将flag信息藏在图片中

No.	Time	Source	Destination	Protocol	Length	Info
83	34.641978404	172.17.0.1	172.17.0.2	TCP	74	48566 → 21106 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=616959586 TSecr=0
84	34.641999014	172.17.0.2	172.17.0.1	TCP	74	21106 → 48566 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2980003694
85	34.642022504	172.17.0.1	172.17.0.2	TCP	66	48566 → 21106 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=616959586 TSecr=2980003694
88	34.642500140	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
89	34.642514480	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
90	34.642525771	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
91	34.642535181	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
92	34.642540101	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
93	34.642546511	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
94	34.642559571	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
95	34.642567911	172.17.0.1	172.17.0.2	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)
96	34.642572981	172.17.0.2	172.17.0.1	FTP	7306	FTP Data: 7240 bytes (PASV) (RETR /universe.png)

发现传输过png文件, 复制原始数据, 用010editor打开, 保存为png文件,





使用了各种方法，无果，转向流量分析，查找ftp-data，搜索其它可能传输的数据，

Wireshark · 追踪 TCP 流 (tcp.stream eq 12) · flag.pcapng

```
220 (vsFTPD 3.0.2)
USER test
331 Please specify the password.
PASS test
230 Login successful.
TYPE I
200 Switching to Binary mode.
OPTS UTF8 ON
200 Always in UTF8 mode.
PASV
227 Entering Passive Mode (172,17,0,2,82,114).
RETR /universe.png
150 Opening BINARY mode data connection for /universe.png (1133535 bytes).
226 Transfer complete.
CWD /new_universe.png
550 Failed to change directory.
SIZE /new_universe.png
550 Could not get file size.
PASV
227 Entering Passive Mode (172,17,0,2,82,111).
STOR /new_universe.png
150 Ok to send data.
226 Transfer complete.
SITE CHMOD 0777 /new_universe.png
200 SITE CHMOD command ok.
CWD /
250 Directory successfully changed.
PASV
227 Entering Passive Mode (172,17,0,2,82,117).
LIST -a
150 Here comes the directory listing.
```

150 Here comes the directory listing.

226 Directory send OK.

PASV

227 Entering Passive Mode (172,17,0,2,82,115).

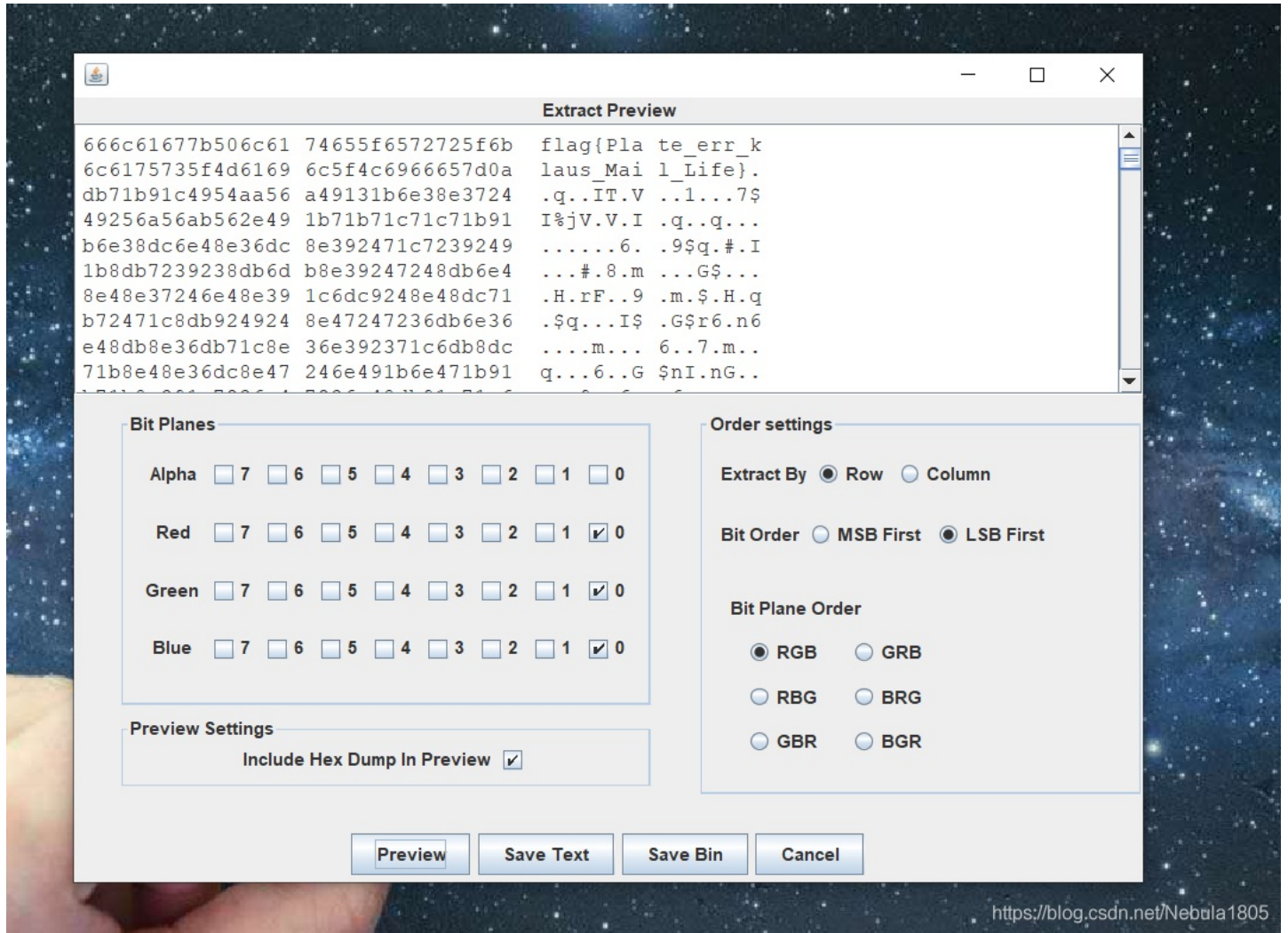
RETR /flag.txt

150 Opening BINARY mode data connection for /flag.txt (41 bytes).

226 Transfer complete.

<https://blog.csdn.net/Nebula1805>

发现又保存了一个png文件,再追踪一下tcp流,得到该文件的原始数据,同上操作,得到一张看上去与前一张没区别的图片,试着使用stegsolve工具



<https://blog.csdn.net/Nebula1805>

发现flag: flag{Plate_err_klaus_Mail_Life}

4.zip口令爆破

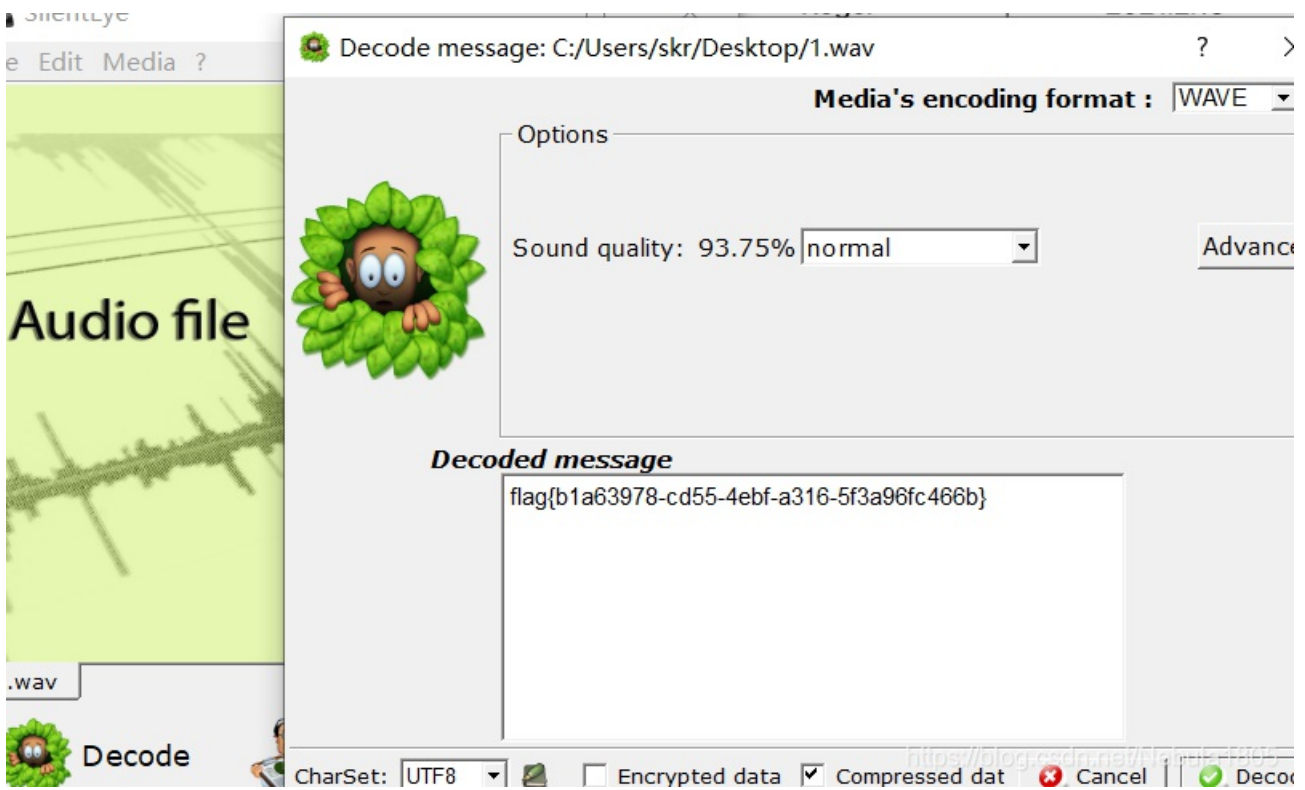
使用zip口令爆破工具，得到解压密码 1658967，得到flag

Advanced Archive Password Recovery 统计信息:	
总计口令	683,705
总计时间	49s 65ms
平均速度(口令/秒)	13,934
这个文件的口令	1658967
十六进制口令	31 36 35 38 39 36 37

保存... <https://blog.csdn.net/Nebula1805> 确定

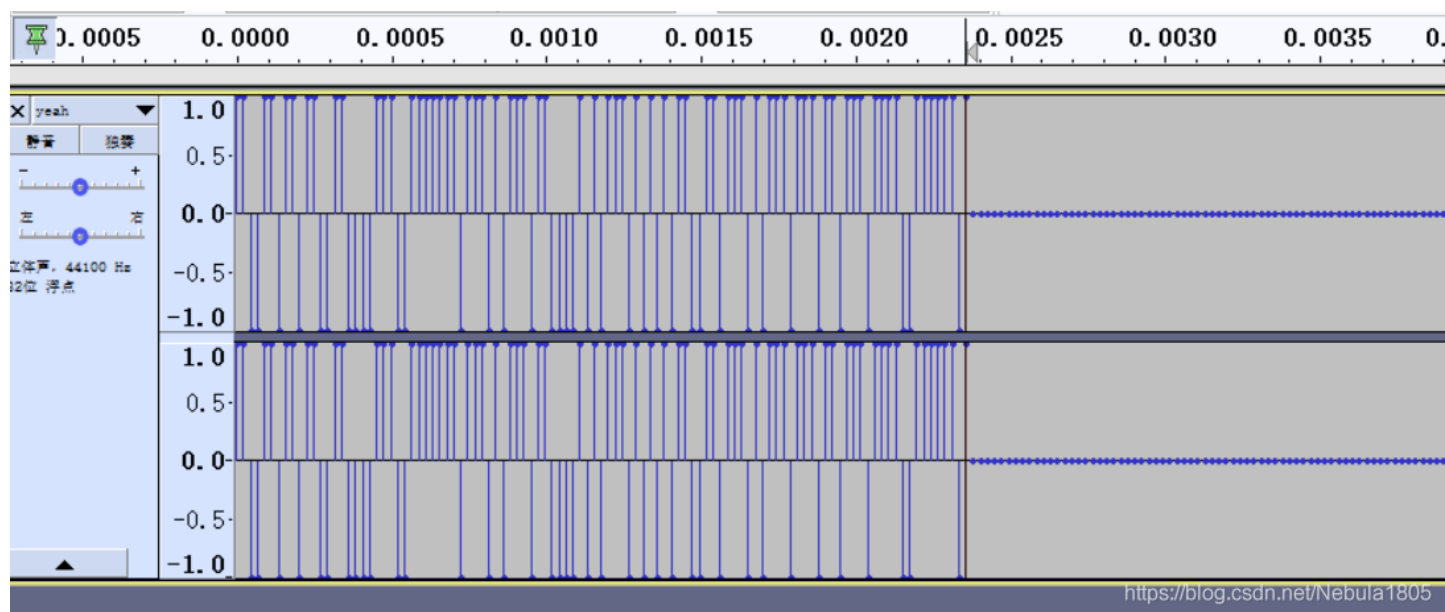
5.LSB音频隐写

使用Silenteye 工具，decode，得到flag



6.音频波形隐写

用Audacity工具打开，发现音频开头又一段可疑的波形图，猜测是高位代表1，低位代表0，然后二进制转ASCII码



则表示的字符串

为: `11001101101100110000111001111111011101011101100001010111010101011001101110101110111011101110111011101`

总共是105位，应该是每7位转，

ASCII ⇌ 进制 进制转换 (常用) 进制转换 (任意) ●

文本 `flag{WOW*funny}`

清空

二进制 `01100110 01101100 01100001 01100111 01111011 01010111 00110000 01010111 00101010 01100110 01110101 01101110 01101110 01111001 01111101`

<https://blog.csdn.net/Nebula1805>

得到flag

7.pyc反编译

- [在线网站反编译](#)
- [使用命令反编译](#)

反编译得到: `#python2`

```
#!/usr/bin/env python
# visit http://tool.lu/pyc/ for more information
print 'Your input1 is your flag~'
l = len(input1)
code = []
for i in range(l):
    num = ((ord(input1[i]) + i) % 128 + 128) % 128
    code += chr(num)

for i in range(l - 1):
    code[i] = chr(ord(code[i]) ^ ord(code[i + 1]))

print code
code = [
    '\x0b',
    '\x0e',
    '\t',
    '\x15',
    '0',
    '4',
    '\x01',
    '\x06',
    '\x14',
    '4',
    ',',
    '\x1b',
    'U',
    '?',
    'o',
    '6',
    '*',
    ':',
    '\x01',
    'D',
    ';',
    '%',
    '\x13']
```

re得

```

code = [
    '\x1f',
    '\x12',
    '\x1d',
    '(',
    '0',
    '4',
    '\x01',
    '\x06',
    '\x14',
    '4',
    ',',
    '\x1b',
    'U',
    '?',
    'o',
    '6',
    '*',
    ':',
    '\x01',
    'D',
    ';',
    '%',
    '\x13']
flag = ''
for i in range(len(code) - 2, -1, -1):
    code[i] = chr(ord(code[i]) ^ ord(code[i + 1]))
for i in range(len(code)):
    code[i] = chr((ord(code[i]) - i) % 128)
    flag += code[i]
print flag

```

运行得flag

```
GWHT{Just_Re_1s_Ha66y!}
```

提交，flag错误，wt? 多次尝试，将 `GWHT` 换成 `flag` 即可

8.MP3 隐写

附件无法播放，用010打开，

```
00000000: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 60  ÿøÿà..JFIF.....`
00000008: 00 60 00 00 FF DB 00 43 00 08 06 06 07 06 05 08  `...ÿÛ.C.....
0000000F: 07 07 07 09 09 08 0A 0C 14 0D 0C 0B 0B 0C 19 12  .....
```

发现是jpg文件头，修改扩展名，得到一张图片



葫芦小金刚

像不像解不开题目的你！

怕你们做不出来，留了条线索给你们，能知道看图片，说明你们还是挺细心的。

Tips: 葫芦小金刚的英文名称就是他唱的歌中的密码噢！
(去除空格,有大小写区分)

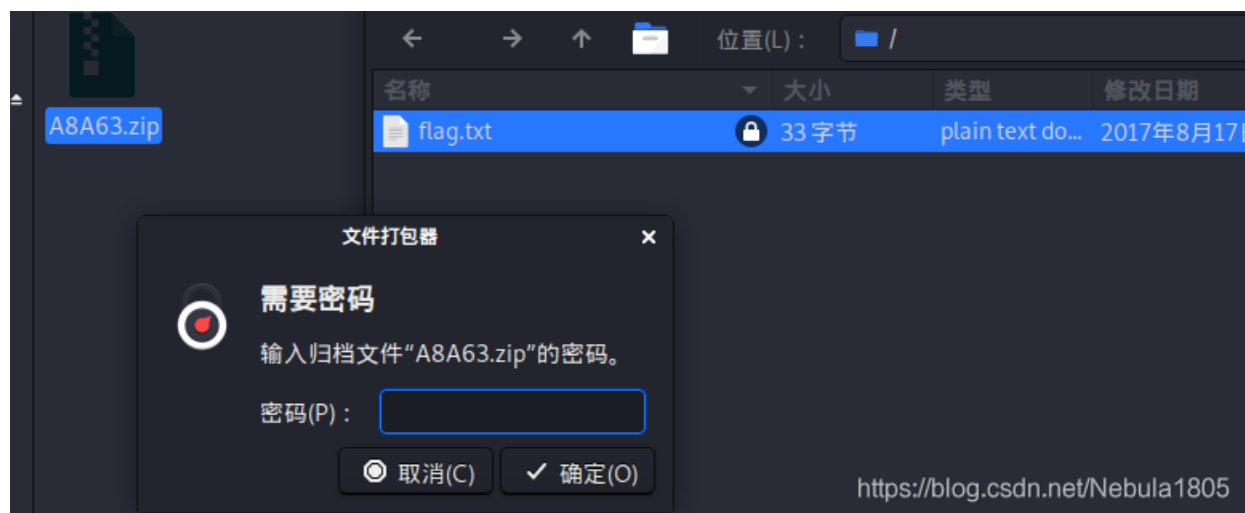
<https://blog.csdn.net/Nebula1805>

并且在末尾发现藏有zip文件，复制到kali，使用binwalk命令，得到一个加密的压缩包

```
8A60h: 00 00 00 50 4B 03 04 14 00 01 08 08 00 C2 BD 11 ...PK.....Â½.
8A70h: 4B 57 EA 42 3B 2B 00 00 00 21 00 00 00 08 00 00 KwêB;+...!.....
8A80h: 00 66 6C 61 67 2E 74 78 74 EB ED 0E 48 2B A5 C4 .flag.txtéí.H+¥Ä
8A90h: C3 35 28 70 90 13 34 A0 12 CC 76 06 BD BF 31 63 Å5(p..4 .Ìv.½ç1c
8AA0h: 01 83 C1 93 B4 E8 1D 16 4B E8 6E 9E 8C 97 C9 2F .fA''`è..KènžŒ-É/
8AB0h: 4D 8A 48 C3 50 4B 01 02 3F 00 14 00 01 08 08 00 MŠHĀPK..?.....
8AC0h: C2 BD 11 4B 57 EA 42 3B 2B 00 00 00 21 00 00 00 Â½.KwêB;+...!...
8AD0h: 08 00 24 00 00 00 00 00 00 00 20 00 00 00 00 00 ..$.
8AE0h: 00 00 66 6C 61 67 2E 74 78 74 0A 00 20 00 00 00 ..flag.txt...
8AF0h: 00 00 01 00 18 00 96 E6 7D EE 6F 17 D3 01 09 AF .....-æ}îo.Ó..
8B00h: 68 DA 6F 17 D3 01 09 AF 68 DA 6F 17 D3 01 50 4B hÚo.Ó..hÚo.Ó.PK
8B10h: 05 06 00 00 00 00 01 00 01 00 5A 00 00 00 51 00 .....Z...Q.
8B20h: 00 00 00 00 .....
```

<https://blog.csdn.net/Nebula1805>

在这里插入图片描述






然后根据图片提示使用Mp3stego工具，下载，用法见该工具文件中的readme.txt，将得到的图片放入Decode.exe所在的文件夹中，当前目录输入cmd回车，输入命令

谷歌翻译：**Gourd Little King Kong**，没用。。。

在网上查了查，试了一下，发现是 **Gourd Small Diamond**

```
E:\Stalker\My_Tools\misc\音频隐写\MP3Stego_1_1_18\MP3Stego>Decode.exe -X -P GourdSmallDiamond 1.mp3
MP3StegoEncoder 1.1.17
See README file for copyright info
Input file = '1.mp3' output file = '1.mp3.pcm'
Will attempt to extract hidden information. Output: 1.mp3.txt
the bit stream file 1.mp3 is a BINARY file
HDR: s=FFF, id=1, l=3, ep=off, br=9, sf=0, pd=1, pr=0, m=0, js=0, c=0, o=0, e=0
alg.=MPEG-1, layer=III, tot bitrate=128, sfrq=44.1
mode=stereo, sblim=32, jsbd=32, ch=2
[Frame 1563]Frame cannot be located
Input stream may be empty
Avg slots/frame = 441.804; b/smp = 3.07; br = 135.302 kbps
Decoding of "1.mp3" is finished
The decoded PCM output file name is "1.mp3.pcm"
https://blog.csdn.net/Nebula1805
```

解压密码

 1.mp3	2021/2/24 10:22	MP3 文件	675 KB
 1.mp3.pcm	2021/2/24 11:28	PCM 文件	7,038 KB
 1.mp3.txt	2021/2/24 11:28	文本文档	1 KB

```
1.mp3.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
解压密码:j7v@8@8QUWG0FWU^
```

解压，得到flag

```
flag.txt 33 字节 plain text do... 2017年8月17日 2
/home/kali/.cache/fr-sa401e/flag.txt - Mousepad
文件(F) 编辑(E) 搜索(S) 视图(V) 文档(D) 帮助(H)
MSTSEC{MSTSEC_DINGANN_KEY_IS_GSD}
```

输入flag，提示错误，经过多次尝试，wt?，梅开二度。。。flag为 **f1ag{MSTSEC_DINGANN_KEY_IS_GSD}**

9.Affine_task

附件py:

```

from string import digits, ascii_lowercase
from secret import numbers, A, B

assert min([i in digits for i in numbers])

flag = "flag{".join([ascii_lowercase[int(i)] for i in numbers])+"}"

assert numbers == "".join([str(ascii_lowercase.find(i)) for i in flag[5:-1]])
Ciphertext = ""
for i in flag:
    if i not in ascii_lowercase:
        Ciphertext += i
    else:
        Ciphertext += ascii_lowercase[(ascii_lowercase.find(i)*A+B) % 26]
print("Ciphertext =", Ciphertext)
# Ciphertext = vjsg{dckvzksr}

```

exp:

```

from Crypto.Util.number import *
from string import ascii_lowercase
table = ascii_lowercase
Ciphertext = "vjsg{dckvzksr}"
MOD = len(table)

def crack():
    for a in range(MOD):
        for b in range(MOD):
            if (a*table.find("f")+b) % MOD == table.find(Ciphertext[0]):
                if (a*table.find("l")+b) % MOD == table.find(Ciphertext[1]):
                    if (a*table.find("a")+b) % MOD == table.find(Ciphertext[2]):
                        if (a*table.find("g")+b) % MOD == table.find(Ciphertext[3]):
                            print("a, b = {}, {}".format(a, b))
                            return (a, b)

flag = ""
A, B = crack()
for i in Ciphertext:
    if i not in table:
        flag += i
    else:
        flag += table[inverse(A, MOD)*(table.find(i)-B) % MOD]
print(flag)
print("".join([str(ascii_lowercase.find(i)) for i in flag[5:-1]]))

```

10.明文攻击

附件为一个zip压缩包，里面是加密的两个文件，flag.doc和readme.txt
明文攻击介绍

明文攻击

81 Points, 23 Solves

well,it plays an important role in flag.

有额外分

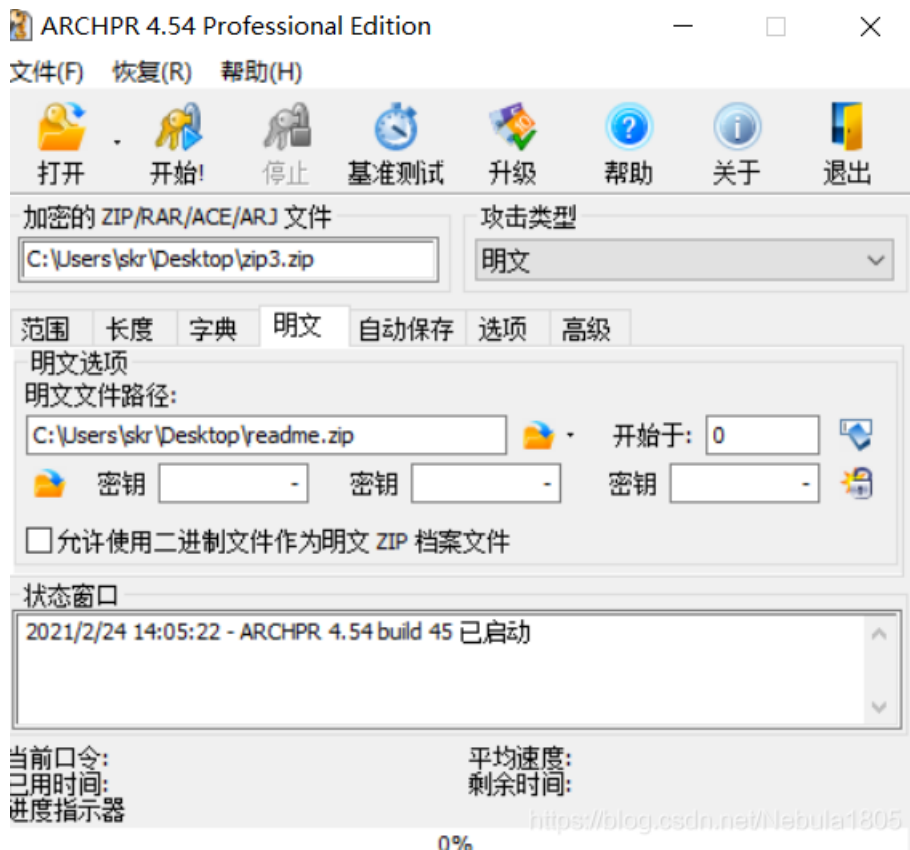
根据提示，再创建一个readme.txt文件，将提示内容粘贴保存，保证加密的和创建的readme.txt文件的CRC32值和文件大小相同，然后以zip压缩

名称	大小	压缩后大小	修改时间	创建时间	访问时间	属性	加密	注释	CRC	算法
flag.doc	18 944	6 481	2020-01-1...	2020-01-1...	2020-01-1...	A	+		BD63F5DE	ZipCryp
readme.txt	40	52	2020-01-1...	2020-01-1...	2020-01-1...	A	+		<u>105F46D0</u>	ZipCryp

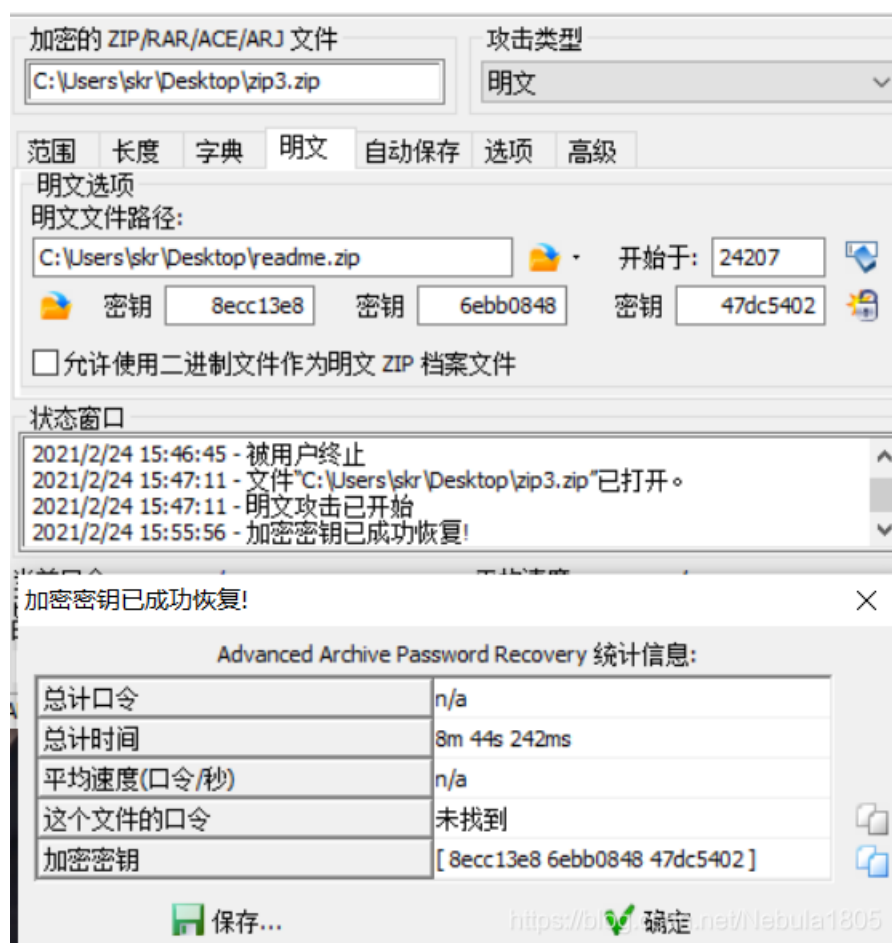


名称	大小	压缩后大小	修改时间	创建时间	访问时间	属性	加密	注释	CRC	算法
readme.txt	40	40	2021-02-2...	2021-02-2...	2021-02-2...	A	-		<u>105F46D0</u>	Store

然后使用ARCHPR工具进行明文攻击，



点击开始，等待。。。。。



点击下方保存，得到一个 `zip3_decrypted.zip` 文件，里面即是已解密的文件，
打开flag.doc,是小说片段，其中藏在flag

当初的少年，自信而且潜力无可估量，不知让得多少少女对其春心荡漾，当然，这也包括以前的萧媚。←

flag{plain_text_is_so_cute}←

然而天才的道路，貌似总是曲折的，三年之前，这名声望达到巅峰的天才少年，却是突兀的接受到了有生以来最残酷的打击，不仅辛辛苦苦修炼十数载方才凝聚的斗之气旋，一夜之间，化为乌有，而且体内的斗之气，也是随着时间的流逝，变得诡异的越来越少。←

11.python脚本使用(一)

附件为一张图片，名为misc.jpg



根据提示，应该用python脚本解决图片隐写

使用binwalk命令分析一下，发现zlib压缩

```
(kali㉿kali)-[~/桌面/555]
└─$ binwalk -e misc.png
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 1000 x 562, 8-bit/color RGBA, non-interlaced
91	0x5B	Zlib compressed data, compressed
1421307	0x15AFFB	Zlib compressed data, default compression

并在生成的extracted文件夹中发现一堆总共625位二进制字符串，emm...25*25=625，



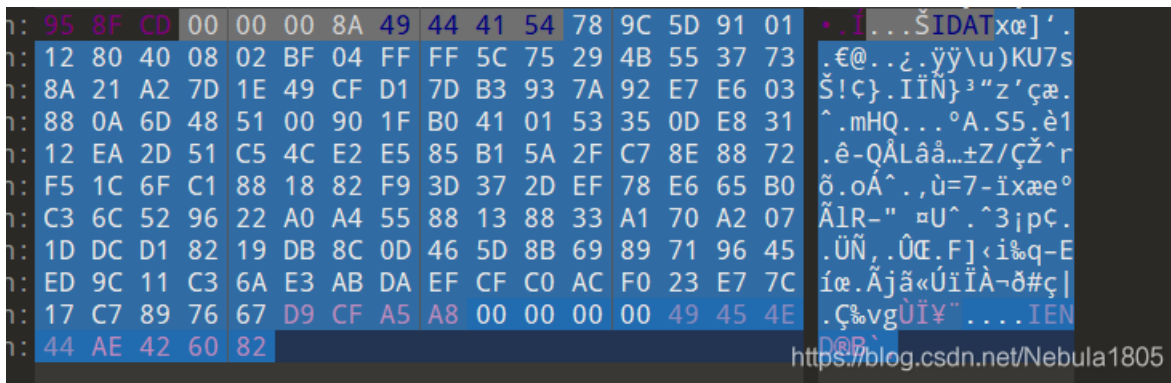
将这些二进制字符串使用python脚本转图片

```
from PIL import Image
MAX = 25
pic = Image.new("RGB", (MAX, MAX))
str = "得到的二进制数字"
i=0
for y in range (0,MAX):
    for x in range (0,MAX):
        if(str[i] == '1'):
            pic.putpixel([x,y],(0, 0, 0))
        else:
            pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
pic.save("flag.png")
```

得到一张二维码，解码得flag



或者用010找到zlib压缩部分，文件头 78 9C



使用zlib解压脚本:


```
import zlib
s = '''
78 9C 5D 91 01 12 80 40 08 02 BF 04 FF FF 5C 75
29 4B 55 37 73 8A 21 A2 7D 1E 49 CF D1 7D B3 93
7A 92 E7 E6 03 88 0A 6D 48 51 00 90 1F B0 41 01
53 35 0D E8 31 12 EA 2D 51 C5 4C E2 E5 85 B1 5A
2F C7 8E 88 72 F5 1C 6F C1 88 18 82 F9 3D 37 2D
EF 78 E6 65 B0 C3 6C 52 96 22 A0 A4 55 88 13 88
B3 A1 70 A2 07 1D DC D1 82 19 DB 8C 0D 46 5D 8B
69 89 71 96 45 ED 9C 11 C3 6A E3 AB DA EF CF C0
AC F0 23 E7 7C 17 C7 89 76 67 D9 CF A5 A8 00 00
00 00 49 45 4E 44 AE 42 60 82

'''
s = s.replace(' ', '').replace('\n', '')
b = bytes.fromhex(s)
flag = zlib.decompress(b)
print(flag)
```

同样得到一堆二进制字符串

12.数字水印隐写

用010打开，点击模板尾，发现还藏有无文件头的png文件

```

Col: F7 72 50 53 32 03 3E DA 93 3E 27 56 78 F6 1A 0F 1F P52 >0 > Vx0 ..
D0h: AA FA FB F5 FB F5 17 BD FE 1F 11 DE 1C DD B2 61 a úóóóó. %p . . P. Ý? a
E0h: 1D 89 00 00 00 00 49 45 4E 44 AE 42 60 82 49 48 % . . . . IEND@B` , IH
F0h: 44 52 00 00 02 F9 00 00 01 E4 08 02 00 00 00 2B DR . . . . ã . . . . +
00h: 7E A7 F7 00 00 20 00 49 44 41 54 78 01 8C E1 69 ~ $ ÷ . . . . IDATx. @á i
10h: 92 6D 09 9A 58 D7 ED FD 9D EB EF F9 6B 22 B2 A9 ' m. šX×íý. ëiùk"²@
20h: 8E 34 40 92 99 7E 68 0C 32 02 23 D0 00 08 CE 81 Ž4@'™~h.2.#Đ. . Î .
30h: 34 89 E4 3C 34 25 0D 84 BF 64 26 50 44 A1 AA 22 4%ã<4% . . . d&PDj a"
40h: 33 E2 75 DE DC F3 6D 5E F7 88 A8 CA 4C 80 34 AE 3âuPÛóm^÷~"ÉL€4@
50h: E5 7F FB 7F FF 77 71 82 4A 8D 9E 21 E2 C2 6E C7 à. ù. ýwq, J. Ž! âÂñÇ
60h: B4 D0 39 CC 29 47 BB 98 C0 CC 9C A4 16 0A 35 43 'ð9İ)G»~ÀÏe# . . 5C
70h: A3 CE 14 5A A9 D4 CE 82 2C 1C 16 20 C4 01 F1 62 fÎ.Z@0Î, . . . Ä.ñb

```

结果 - PNG.bt

名称	值	开始	大小	颜色	注释
uct PNG_SIGNATURE...		0h	8h	Fg: Bg:	
uct PNG_CHUNK chu... IHDR (Critical,...		8h	19h	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		21h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		1002Dh	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		20039h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		30045h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		40051h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		5005Dh	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		60069h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		70075h	1000Ch	Fg: Bg:	
uct PNG_CHUNK chu... IDAT (Critical,...		80081h	6561h	Fg: Bg:	
uct PNG_CHUNK chu... IEND (Critical,...		865E2h	Ch	Fg: Bg:	

<https://blog.csdn.net/Nebula1805>

可以直接用010新建文件，将隐藏的png文件数据复制到新建文件中，并添加上png文件头，**89 50 4E 47 0D 0A 1A 0A 00 00 00 0D**，保存

```

L$ zsteg -a half.png
[?] 522453 bytes of extra data after image end (IEND), offset = 0x865ee
extradata:0
00000000: 49 48 44 52 00 00 02 f9 00 00 01 e4 08 02 00 00 | IHDR..... |
00000010: 00 2b 7e a7 f7 00 00 20 00 49 44 41 54 78 01 8c | .+~... .IDATx.. |
00000020: e1 69 92 6d 09 9a 58 d7 ed fd 9d eb ef f9 6b 22 | .i.m..X.....k" |
00000030: b2 a9 8e 34 40 92 99 7e 68 0c 32 02 23 d0 00 08 | ...4@~h.2.#... |
00000040: ce 81 34 89 e4 3c 34 25 0d 84 bf 64 26 50 44 a1 | ..4.<4%...d5PD. |
00000050: aa 22 33 e2 75 de dc f3 6d 5e f7 88 a8 ca 4c 80 | ."3.u...m^....L. |
00000060: 34 ae e5 7f fb 7f ff 77 71 82 4a 8d 9e 21 e2 c2 | 4.....wq.J..!.. |
00000070: 6e c7 b4 d0 39 cc 29 47 bb 98 c0 cc 9c a4 16 0a | n...9.)G..... |
00000080: 35 43 a3 ce 14 5a a9 d4 ce 82 2c 1c 16 20 c4 01 | 5C...Z..... |
00000090: f1 62 e6 d8 7d 86 41 88 63 e6 ee f2 e6 fe 78 33 | .b..}.A.c.....x3 |
000000a0: ce 23 4f cf 8f 4f cf e7 35 82 81 32 43 5e 04 c1 | .#0..0..5..2C^.. |
000000b0: f0 2f 84 40 e5 67 a5 f2 2f e2 4f 78 03 81 11 89 | ./.@.g../.0x.... |
000000c0: fc 05 11 81 4a 1d 7e 25 62 2d c8 ab 40 5e 78 43 | ....J.~%b-..@^xC |
000000d0: a1 06 91 10 01 87 73 02 46 52 ce 00 0d 26 b5 20 | .....s.FR...6. |
000000e0: 7f 4e 44 fe 99 5a 19 c9 8d f1 aa c4 44 60 49 31 | ND..Z.....D'I1 |
000000f0: 38 34 83 81 58 b2 38 f5 30 5c 96 18 d4 61 22 42 | 84.9x8.0\...aB |

```

<https://blog.csdn.net/Nebula1805>

```

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABC
: 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....I
: 00 00 02 F9 00 00 01 E4 08 02 00 00 00 2B 7E A7 ...ù...ä.....
: F7 00 00 20 00 49 44 41 54 78 01 8C E1 69 92 6D ÷...IDATx. @á i
: 09 9A 58 D7 ED FD 9D EB EF F9 6B 22 B2 A9 8E 34 .šX×íý.ëiùk"²@
: 40 92 99 7E 68 0C 32 02 23 D0 00 08 CE 81 Ž4@'™~h.2.#Đ. . Î .


```



<https://blog.csdn.net/Nebula1805>

得到一张与原图一样的图片，猜测是双图隐写

使用BlindWaterMark工具，将两张图片复制到该工具下，并在当前目录cmd回车，输入命令 `python3 bwmforpy3.py decode half.png 123.png 333.png`，然后什么事也没发生，不知道是什么原因，难道是python库的版本不对???

 requirements.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
opencv-python==4.2.0.34
matplotlib==2.1.1
```

无果。。。

这题解题思路可参见[南京大学：数字水印隐写writeup](#)

13.流量分析(二)-hard?????

14.流量分析(三)?????

15.内存取证(一)

解题思路见[内存取证](#)

16.python脚本使用(二)?????

17.流量分析-hard???????

18.内存取证(二)???????