BUUCTF MISC刷题笔记(三)

原创

z.volcano 于 2021-05-07 18:05:21 发布 2476 收藏 6

分类专栏： # buuoj # 刷题

本文链接：https://blog.csdn.net/weixin_45696568/article/details/116421340

buuoj 同时被 2 个专栏收录

7 篇文章 1 订阅

订阅专栏

刷题

8 篇文章 0 订阅

订阅专栏

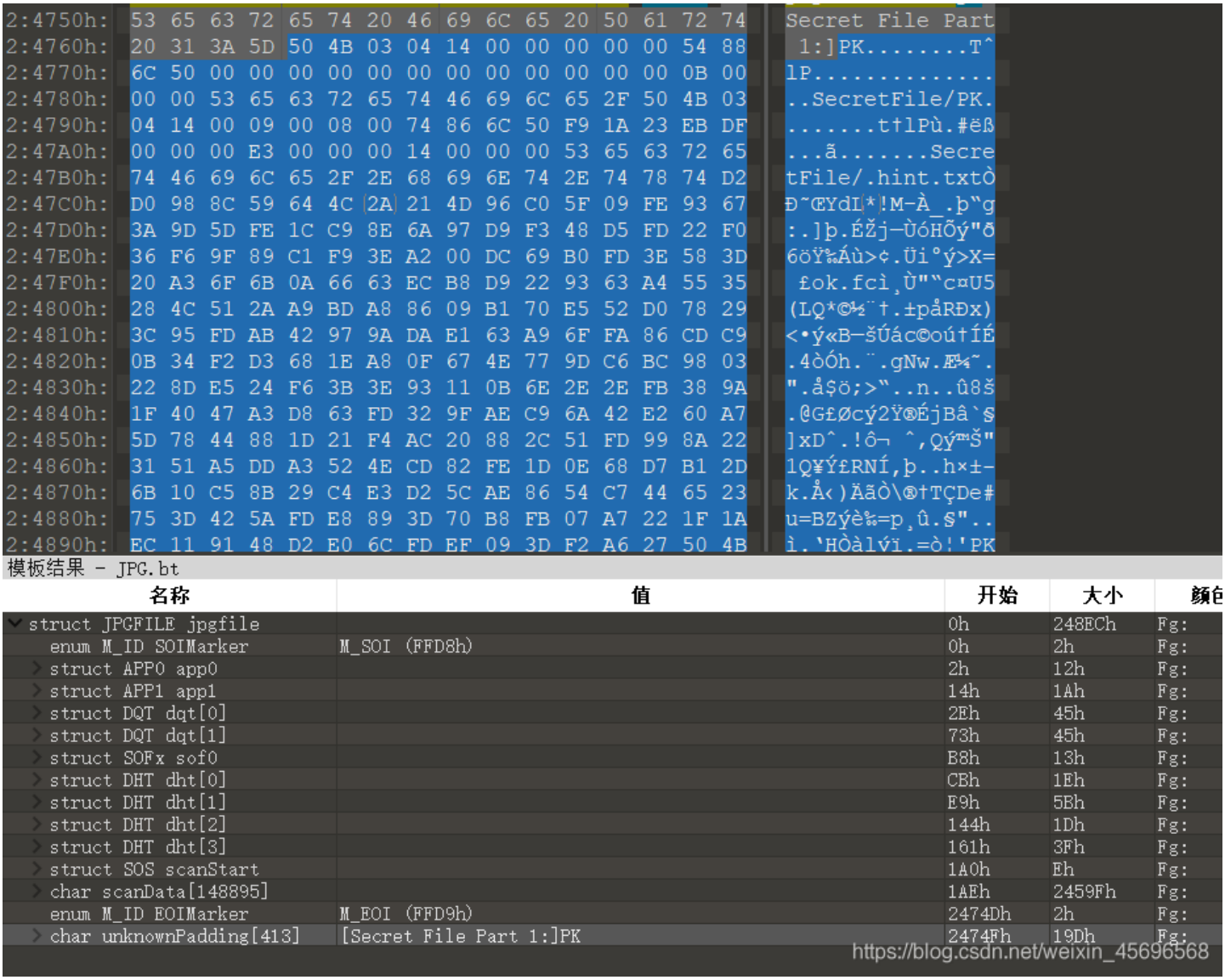
BUUOJ

- Misc
  - [MRCTF2020]pyFlag
  - Business Planning Group
  - [ACTF新生赛2020]剑龙
  - [GWCTF2019]huyao
  - [UTCTF2020]File Carving
  - [GUET-CTF2019]soul sipse
  - [watevrCTF 2019]Evil Cuteness
  - [UTCTF2020]sstv
  - [UTCTF2020]spectogram
  - 我爱Linux

Misc

[MRCTF2020]pyFlag

三个图片尾部都有额外数据

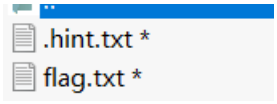


2:4750h:	53 65 63 72 65 74 20 46 69 6C 65 20 50 61 72 74	Secret File Part
2:4760h:	20 31 3A 5D 50 4B 03 04 14 00 00 00 00 00 54 88	1:]PK.....T^
2:4770h:	6C 50 00 00 00 00 00 00 00 00 00 00 00 00 0B 00	lP.....
2:4780h:	00 00 53 65 63 72 65 74 46 69 6C 65 2F 50 4B 03	..SecretFile/PK.
2:4790h:	04 14 00 09 00 08 00 74 86 6C 50 F9 1A 23 EB DF	.....t†lPù.#ëß
2:47A0h:	00 00 00 E3 00 00 00 14 00 00 00 53 65 63 72 65	...ã.....Secre
2:47B0h:	74 46 69 6C 65 2F 2E 68 69 6E 74 2E 74 78 74 D2	tFile/.hint.txtÒ
2:47C0h:	D0 98 8C 59 64 4C 2A 21 4D 96 C0 5F 09 FE 93 67	Ð˜ŒYdL*!M-À_.þ"g
2:47D0h:	3A 9D 5D FE 1C C9 8E 6A 97 D9 F3 48 D5 FD 22 F0	:.]þ.ÉŽj—ÙóHÕý"ð
2:47E0h:	36 F6 9F 89 C1 F9 3E A2 00 DC 69 B0 FD 3E 58 3D	6öŸ‰Áù>¢.Üi°ý>X=
2:47F0h:	20 A3 6F 6B 0A 66 63 EC B8 D9 22 93 63 A4 55 35	£ok.fcì¸Ù"“c¤U5
2:4800h:	28 4C 51 2A A9 BD A8 86 09 B1 70 E5 52 D0 78 29	(LQ*©½¨†.±påRÐx)
2:4810h:	3C 95 FD AB 42 97 9A DA E1 63 A9 6F FA 86 CD C9	<•ý«B—šÚác©oú†ÍÉ
2:4820h:	0B 34 F2 D3 68 1E A8 0F 67 4E 77 9D C6 BC 98 03	.4òÓh.¨.gNw.Æ¼˜.
2:4830h:	22 8D E5 24 F6 3B 3E 93 11 0B 6E 2E 2E FB 38 9A	".å$ö;>“..n..û8š
2:4840h:	1F 40 47 A3 D8 63 FD 32 9F AE C9 6A 42 E2 60 A7	.@G£Øcý2Ÿ®ÉjBâ`§
2:4850h:	5D 78 44 88 1D 21 F4 AC 20 88 2C 51 FD 99 8A 22	]xDˆ.!ô¬ ˆ,Qý™Š"
2:4860h:	31 51 A5 DD A3 52 4E CD 82 FE 1D 0E 68 D7 B1 2D	1Q¥Ý£RNÍ‚þ..h×±-
2:4870h:	6B 10 C5 8B 29 C4 E3 D2 5C AE 86 54 C7 44 65 23	k.Å‹)ÄãÒ\®†TÇDe#
2:4880h:	75 3D 42 5A FD E8 89 3D 70 B8 FB 07 A7 22 1F 1A	u=BZýè‰=p¸û.§"..
2:4890h:	EC 11 91 48 D2 E0 6C FD EF 09 3D F2 A6 27 50 4B	ì.‘HÒàlýï.=ò¦'PK

模板结果 - JPG.bt

名称	值	开始	大小	颜色
struct JPGFILE jpgfile		0h	248ECh	Fg:
enum M_ID SOIMarker	M_SOI (FFD8h)	0h	2h	Fg:
struct APP0 app0		2h	12h	Fg:
struct APP1 app1		14h	1Ah	Fg:
struct DQT dqt[0]		2Eh	45h	Fg:
struct DQT dqt[1]		73h	45h	Fg:
struct SOFx sof0		B8h	13h	Fg:
struct DHT dht[0]		CBh	1Eh	Fg:
struct DHT dht[1]		E9h	5Bh	Fg:
struct DHT dht[2]		144h	1Dh	Fg:
struct DHT dht[3]		161h	3Fh	Fg:
struct SOS scanStart		1A0h	Eh	Fg:
char scanData[148895]		1AEh	2459Fh	Fg:
enum M_ID EOIMarker	M_EOI (FFD9h)	2474Dh	2h	Fg:
char unknownPadding[413]	[Secret File Part 1:]PK	2474Fh	19Dh	Fg:

按顺序拼在一起，另存为zip文件，里面有俩txt，爆破得到密码`1234`



hint.txt：

> 我用各种baseXX编码把flag套娃加密了，你应该也有看出来。
> 但我只用了一些常用的base编码哦，毕竟我的智力水平你也知道…像什么base36base58听都没听过
> 提示：0x10,0x20,0x30,0x55

flag.txt：

> G&eOhGcq(ZG(t2*H8M3dG&wXiGcq(ZG&wXyG(jtG&eOdGcq+aG(t5oG(jqG&eleGcq+aG)6Q<G(jrG&eOdH9<5qG&eLvG(jsG&nRdH9<8rG%++qG%_
> _eG&eleGc+|cG(t5oG(jsG&eOlH9<8rH8C_qH9<8oG&eOhGc+_bG&eLvH9<8sG&eLgGcz?cG&3|sH8M3cG&eOtG%_?
> aG(t5oG(jtG&wXxGcq+aH8V6sH9<8rG&eOhH9<5qG(<E-H8M3eG&wXiGcq(ZG)6Q<G(jtG&eOtG%+
> <aG&wagG%__cG&eleGcq+aG&M9uH8V6cG&eOlH9<8rG(<HrG(jqG&eLcH9<8sG&wUwGek2)

根据hint可以知道可能有base16、32、48、85

先用Python进行base85解码一次

```
>>> import base64
>>> n='''G&e0hGcq(ZG(t2*H8M3dG&wXiGcq(ZG&wXyG(j~tG&e0dGcq+aG(t5oG(j~qG&eIeGcq+aG
)6Q<G(j~rG&e0dH9<5qG&eLvG(j~sG&nRdH9<8rG%++qG%__eG&eIeGc+|cG(t5oG(j~sG&e01H9<8rH
8C_qH9<8oG&e0hGc+_bG&eLvH9<8sG&eLgGcz?cG&3|sH8M3cG&e0tG%_?aG(t5oG(j~tG&wXxGcq+aH
8V6sH9<8rG&e0hH9<5qG(<E-H8M3eG&wXiGcq(ZG)6Q<G(j~tG&e0tG%+<aG&wagG%__cG&eIeGcq+aG
&M9uH8V6cG&e01H9<8rG(<HrG(j~qG&eLcH9<8sG&wUwGek2)'''
>>> base64.b85decode(n)
b'475532444B4E525549453244494E4A57475132544B514A54473432544F4E4A5547515A44474D4A
5648415A54414E4257473434544B514A5647595A54514D5A5147553444474D5A5547453355434E52
54475A42444B514A57494D3254534D5A5447555A444D4E5256494532444F4E4A57475A4154495242
5547343254454E534447595A544D524A5447415A55493D3D3D'
>>>
```


再base16解码一次

> GU2DKNRUIE2DINJWGQ2TKQJTG42TONJUGQZDGMJVHAZTANBWG44TKQJVGYZTQMZQGU4DGMZUGE3UCNRTGZBDKQJWIM2TS
> MZTGUZDMNRVIE2DONJWGZATIRBUG42TENSDGYZTMRJTGAZUI===

然后base32

> 54564A4456455A3757544231583046795A5638305833417A636B5A6C593352665A47566A4D47526C636E303D

再16

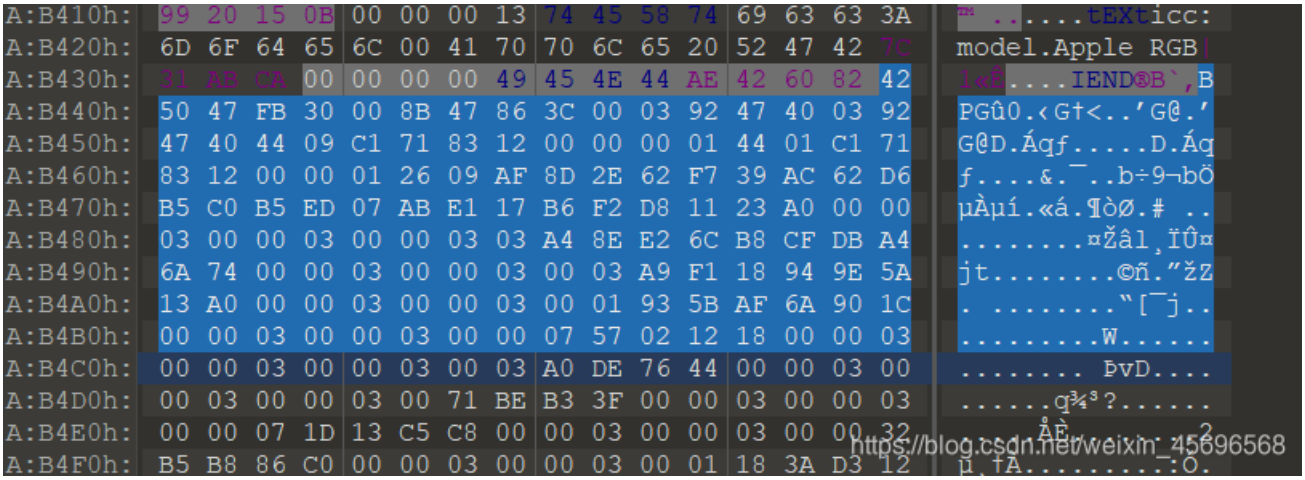
> TVJDVEZ7WTB1X0FyZV80X3AzckZlY3RfZGVjMGRlcn0=

最后来一下base64

> MRCTF{Y0u_Are_4_p3rFect_dec0der}

Business Planning Group

图片结尾有点东西，这个其实是bpg文件，misc入门的misc3考过



提取出来保存为bpg文件，不过windows下不能直接查看，参考上面那篇博客，下载工具查看

图片里有字符串

> YnNpZGVzX2RlbGhpe0JQR19pNV9iM3R0M3JfN2g0bl9KUEd9Cg==

base64解码得到`bsides_delhi{BPG_i5_b3tt3r_7h4n_JPG}`

[ACTF新生赛2020]剑龙

hint.zip	65,503	65,478	WinRAR ZIP 压缩...	2020/1/28 0:51	76C77770
O_O	2,206	1,252	文件	2019/12/2 19:...	79B2BD1D

先看hint.zip的内容，里面有两个文件

hh.jpg	74,380	64,781	JPG 文件	2020/1/16 22:...	4CCB99BE
pwd.txt	1,974	450	文本文档	2020/1/28 0:18	B74E1613

看一下txt

```
ﾟωﾟ/= / ｀m´) ﾉ ~┻━┻   //*´▽｀*/ ['_']; o=(ﾟｰﾟ)  =_=3; c=(ﾟΘﾟ) =(ﾟｰﾟ)-(ﾟｰﾟ); (ﾟДﾟ) =(ﾟΘﾟ)= (o^_^o)/ (o^_^o);(ﾟДﾟ)={ﾟΘﾟ: '_' ,ﾟωﾟﾉ : ((ﾟωﾟﾉ==3) +'_') [ﾟΘﾟ] ,ﾟｰﾟﾉ :(ﾟωﾟﾉ+ '_')[o^_^o -(ﾟΘﾟ)] ,ﾟДﾟ/:((ﾟｰﾟ==3) +'_')[ﾟｰﾟ] }; (ﾟДﾟ) [ﾟΘﾟ] =((ﾟωﾟﾉ==3) +'_') [c^_^o];(ﾟДﾟ) ['c'] = ((ﾟДﾟ)+'_') [ (ﾟｰﾟ)+(ﾟｰﾟ)-(ﾟΘﾟ) ];(ﾟДﾟ) ['o'] = ((ﾟДﾟ)+'_') [ﾟΘﾟ];(ﾟoﾟ)=(ﾟДﾟ) ['c']+(ﾟДﾟ) ['o']+(ﾟωﾟﾉ +'_')[ﾟΘﾟ]+ ((ﾟωﾟﾉ==3) +'_') [ﾟｰﾟ] + ((ﾟДﾟ) +'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ ((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [(ﾟｰﾟ) - (ﾟΘﾟ)]+(ﾟДﾟ) ['c']+((ﾟДﾟ)+'_') [(ﾟｰﾟ)+(ﾟｰﾟ)]+ (ﾟДﾟ) ['o']+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ];(ﾟДﾟ) ['_'] =o^_^o) [ﾟoﾟ] [ﾟoﾟ];(ﾟεﾟ)=((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟДﾟ) .ﾟДﾟ/+((ﾟДﾟ)+'_') [(ﾟｰﾟ) + (ﾟｰﾟ)]+((ﾟｰﾟ==3) +'_') [o^_^o -ﾟΘﾟ]+((ﾟｰﾟ==3) +'_') [ﾟΘﾟ]+ (ﾟωﾟﾉ +'_') [ﾟΘﾟ]; (ﾟｰﾟ)+=(ﾟΘﾟ); (ﾟДﾟ)[ﾟεﾟ]='\\'; (ﾟДﾟ)[ﾟoﾟ]=(ﾟωﾟﾉ +'_')[c^_^o];(ﾟДﾟ) [ﾟoﾟ]='\"';(ﾟДﾟ) ['_'] ( (ﾟДﾟ) ['_'] (ﾟεﾟ)+(ﾟДﾟ)[ﾟoﾟ]+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((o^_^o) +(o^_^o))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟｰﾟ)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ (ﾟｰﾟ)+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (o^_^o))+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟΘﾟ)+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ ((ﾟｰﾟ) + (ﾟΘﾟ))+ (ﾟДﾟ)[ﾟεﾟ]+((o^_^o) +(o^_^o))+ (o^_^o)+ (ﾟДﾟ)[ﾟεﾟ]+(ﾟｰﾟ)+ (ﾟΘﾟ)+ (ﾟДﾟ)[ﾟoﾟ]) (ﾟΘﾟ)) ('_');
```


到在线网址解密一手，得到`welcom3!`

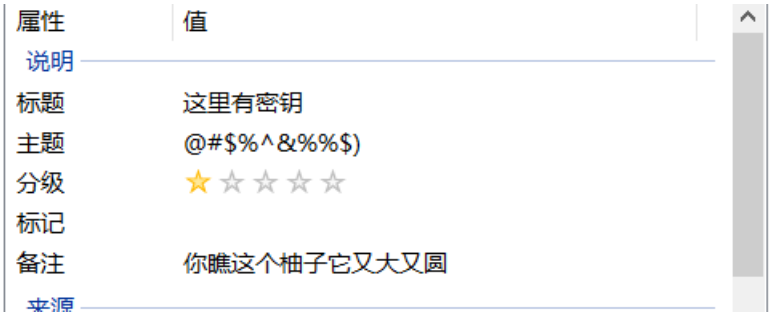
得到的这个应该是密码，说明hh.jpg应该是某种隐写，测试发现是`steghide`

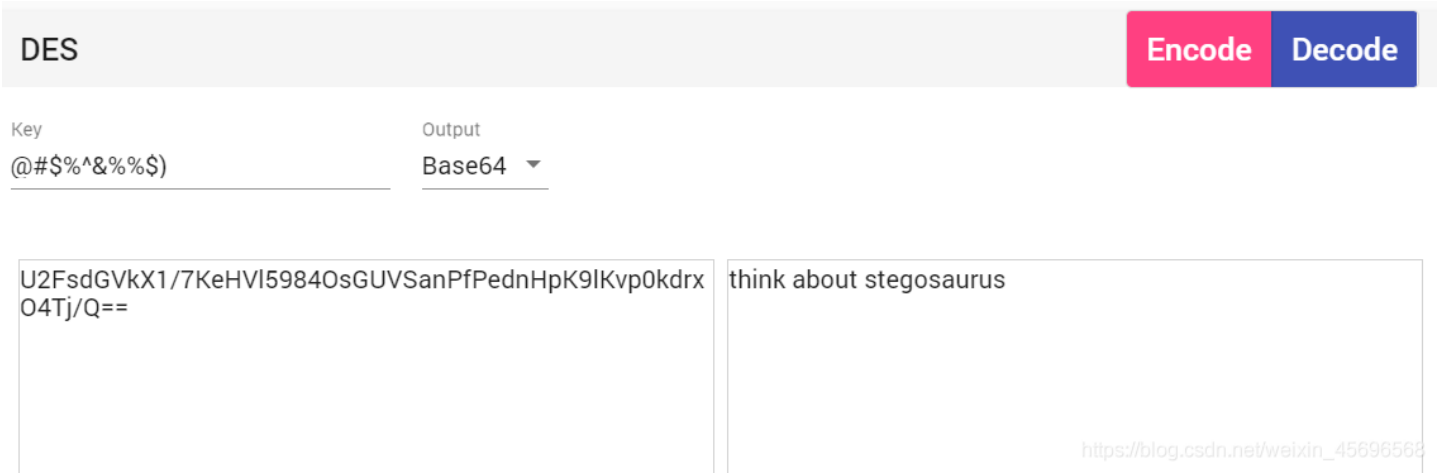
```
┌──(root💀kali)-[/home/volcano/桌面]
└─# steghide extract -sf hh.jpg -p welcom3!
wrote extracted data to "secret.txt".
```

解出的内容如下：

> 想要flag吗？解出我的密文吧~
> U2FsdGVkX1/7KeHVl5984OsGUVSanPfPednHpK9lKvp0kdrxO4Tj/Q==

U2F开头，应该是`AES`或者`DES`这种，不过一般需要密码，再去找一下密码的线索，在图片的exif信息中





这里解出来让找一下`stegosaurus`，在github上找到https://github.com/AngelKitty/stegosaurus

这个工具我以前做题的时候用过，详细介绍可以看这篇博客

题目中给的O_O其实是一个`pyc`文件，改回后缀后使用这个工具解一下就行

```
C:\Users\      \Desktop\stegosaurus隐写>python stegosaurus.py -x 0_0.pyc
Extracted payload: flag{3teg0Sauru3_!1}
```

[GWCTF2019]huyao

给了两张一样的图片，应该是盲水印

盲水印的话我遇到过三种，具体在另一篇博客介绍过

这里试了一下发现是`频域盲水印`，上脚本

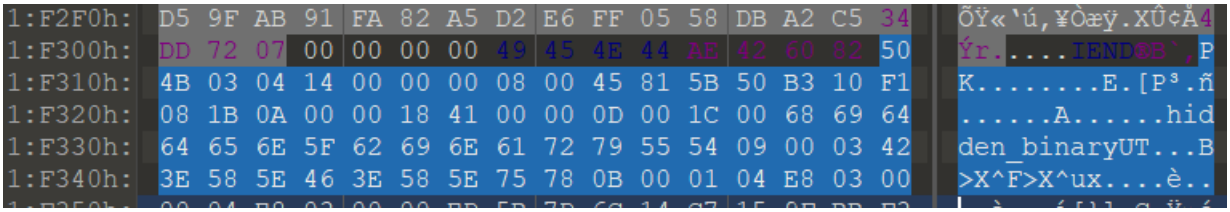
```
# coding=utf-8
import cv2
import numpy as np
import random
import os
from argparse import ArgumentParser
ALPHA = 5
def build_parser():
    parser = ArgumentParser()
    parser.add_argument('--original', dest='ori', required=True)
    parser.add_argument('--image', dest='img', required=True)
    parser.add_argument('--result', dest='res', required=True)
    parser.add_argument('--alpha', dest='alpha', default=ALPHA)
    return parser
def main():
    parser = build_parser()
    options = parser.parse_args()
    ori = options.ori
    img = options.img
    res = options.res
    alpha = options.alpha
    if not os.path.isfile(ori):
        parser.error("original image %s does not exist." % ori)
    if not os.path.isfile(img):
        parser.error("image %s does not exist." % img)
    decode(ori, img, res, alpha)
def decode(ori_path, img_path, res_path, alpha):
    ori = cv2.imread(ori_path)
    img = cv2.imread(img_path)
    ori_f = np.fft.fft2(ori)
    img_f = np.fft.fft2(img)
    height, width = ori.shape[0], ori.shape[1]
    watermark = (ori_f - img_f) / alpha
    watermark = np.real(watermark)
    res = np.zeros(watermark.shape)
    random.seed(height + width)
    x = range(height / 2)
    y = range(width)
    random.shuffle(x)
    random.shuffle(y)
    for i in range(height / 2):
        for j in range(width):
            res[x[i]][y[j]] = watermark[i][j]
    cv2.imwrite(res_path, res, [int(cv2.IMWRITE_JPEG_QUALITY), 100])
if __name__ == '__main__':
    main()
```

命令：`python BlindWaterMarkplus.py --original 1.png --image 2.png --result res.png`

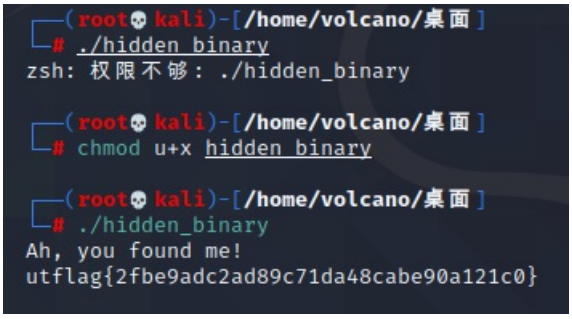
拿到flag{BWM_1s_c00l}

[UTCTF2020]File Carving

尾部有额外数据，提取出来另存为zip文件



解压后拿到文件`hidden_binary`，看了一下发现是ELF文件，果断放进kali里运行一下



这里遇到一个小问题，但是顺利解决了，同时学到一个命令

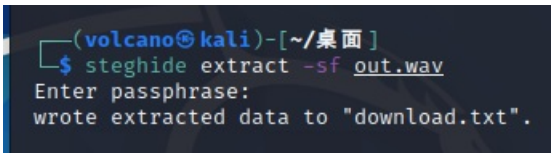
```
chmod u+x somefile 只授予这个文件的所属者执行的权限
```

[GUET-CTF2019]soul sipse

下载得到out.wav，拿Audacity分析下，没啥东西

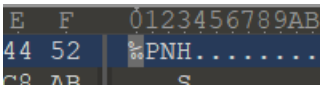
那应该就是隐写了，一般情况下是`mp3stego`，不过这题居然是`steghide`，属实少见

这里没有密码，解出一个txt



> https://share.weiyun.com/5wVTIN3

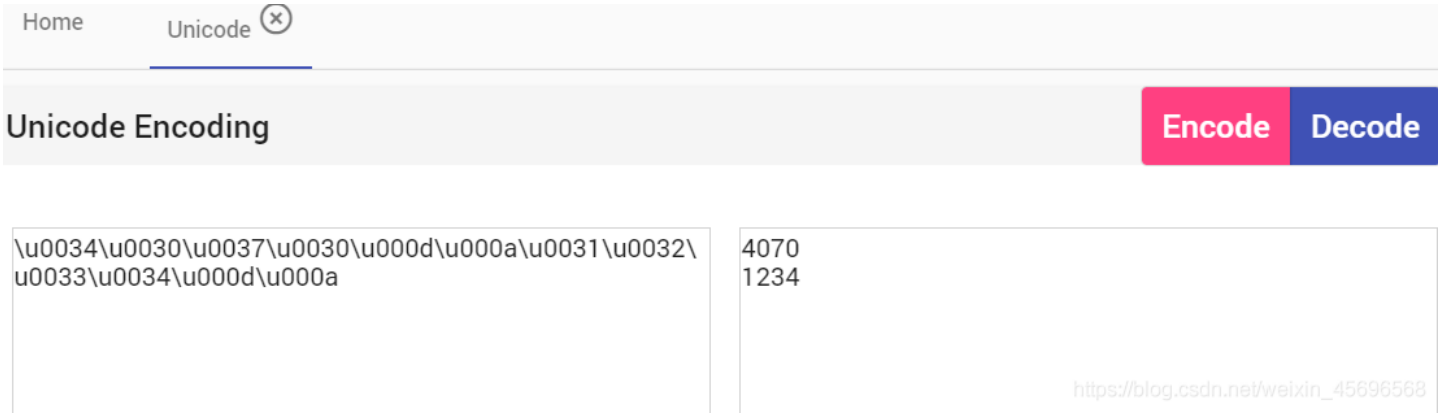
把文件下载下来，显示异常，原来是文件头有问题



修复好文件头，图片内容如下

\u0034\u0030\u0037\u0030\u000d\u000a\u0031\u0032\u0033\u0034\u000d\u000a

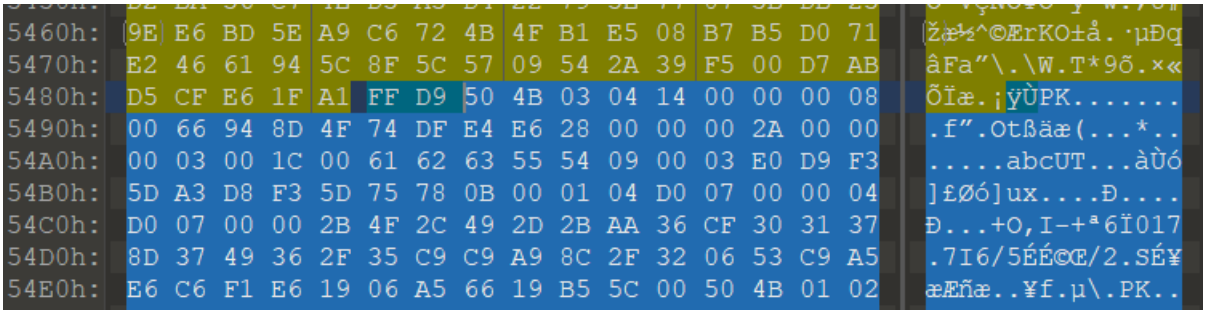
unicode解码一下



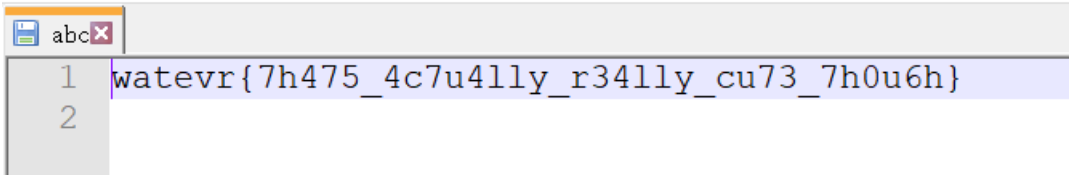
两个数加起来就是flag：`flag{5304}`

[watevrCTF 2019]Evil Cuteness

尾部有额外数据，另存为zip



解压得到无后缀文件abc



就这…

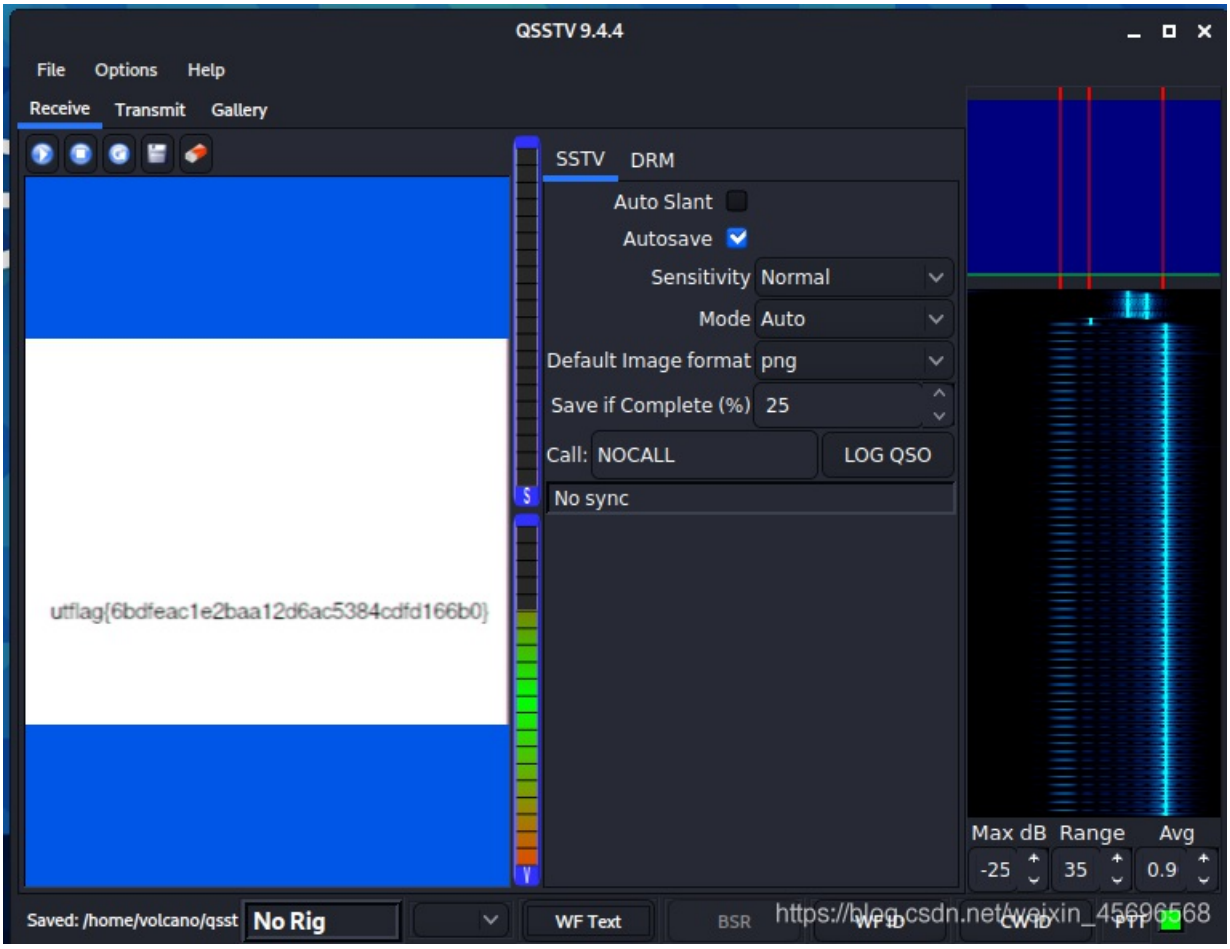
[UTCTF2020]sstv

又是一个wav，百度一下题目名sstv，发现这玩意有点神奇啊

这里用到工具`QSSTV`

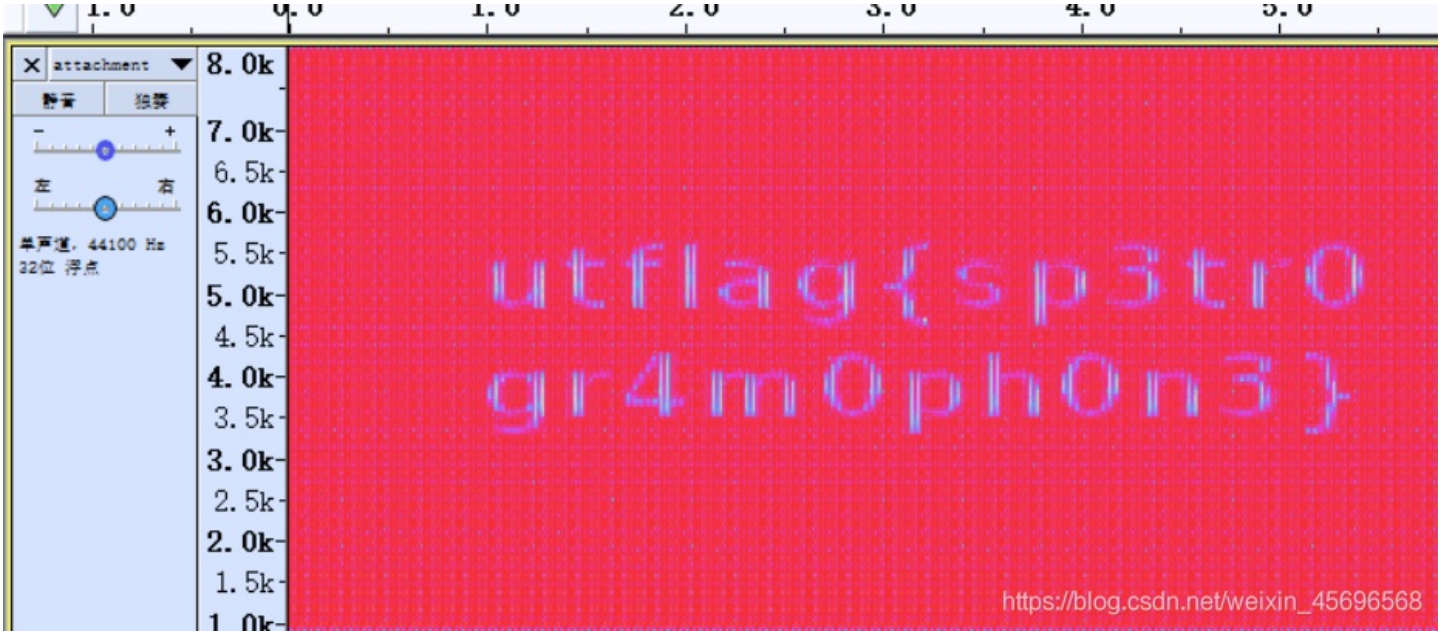
在linux下安装：`apt install qsstv`

安装完打开`qsstv`，操作一波即可，关于qsstv的具体使用参考这篇文章



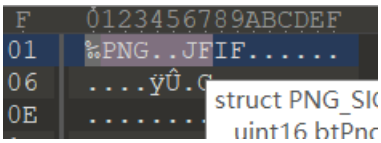
[UTCTF2020]spectogram

直接看频谱图...

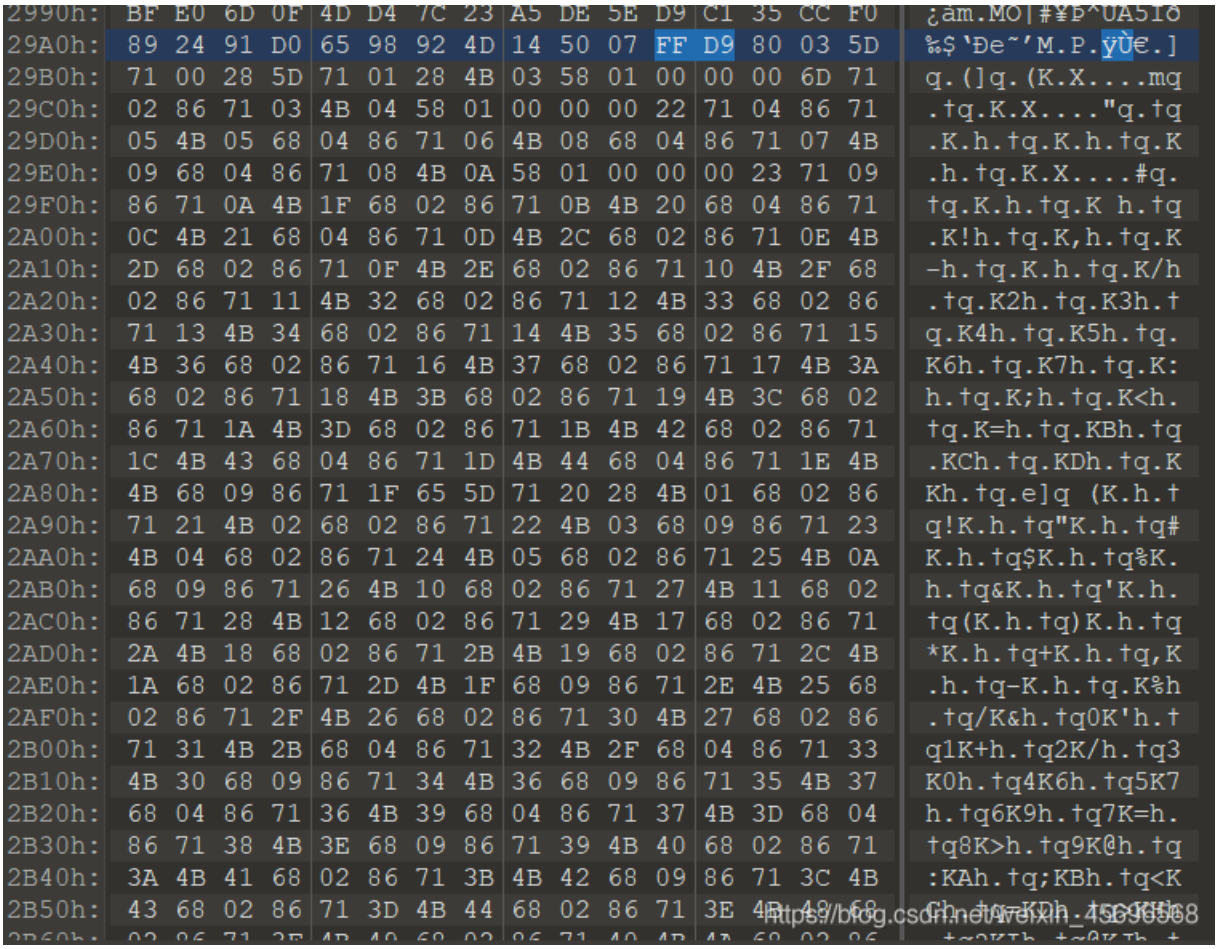


我爱Linux

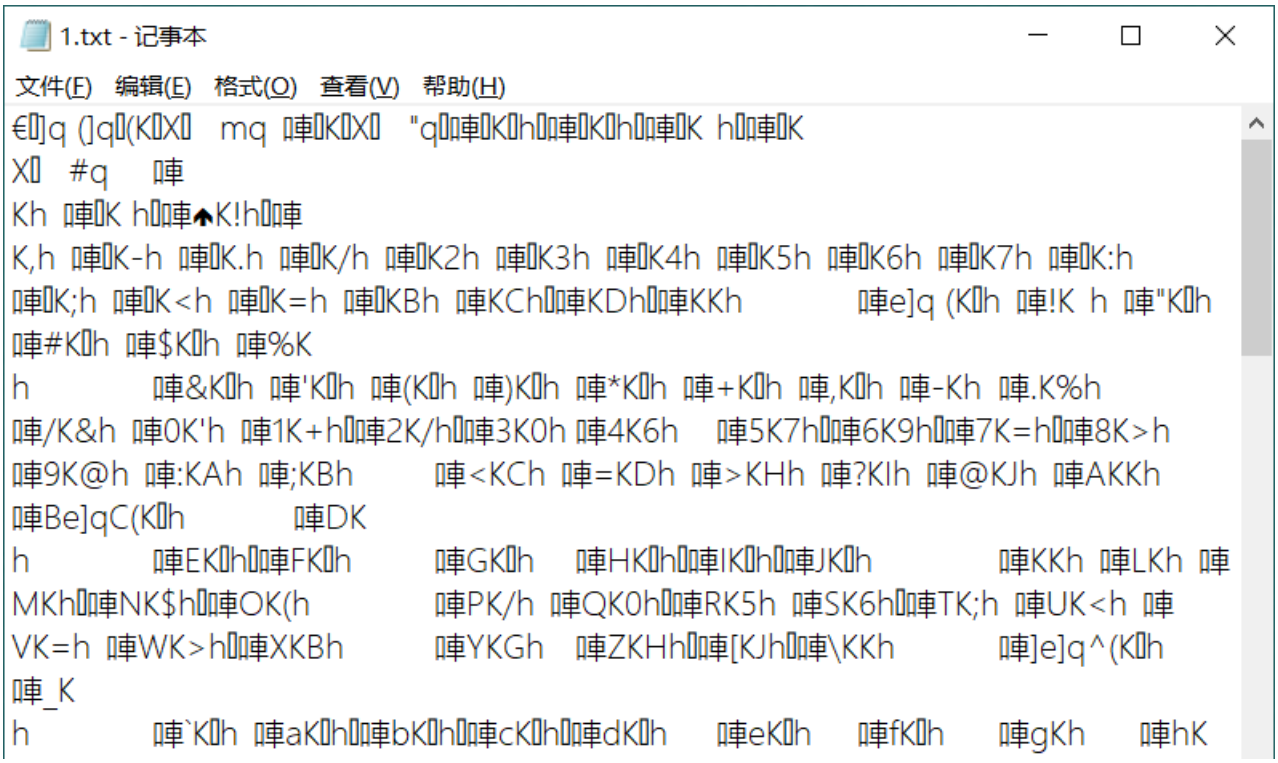
下载的图片打不开，发现是png的文件头，jpg的文件尾

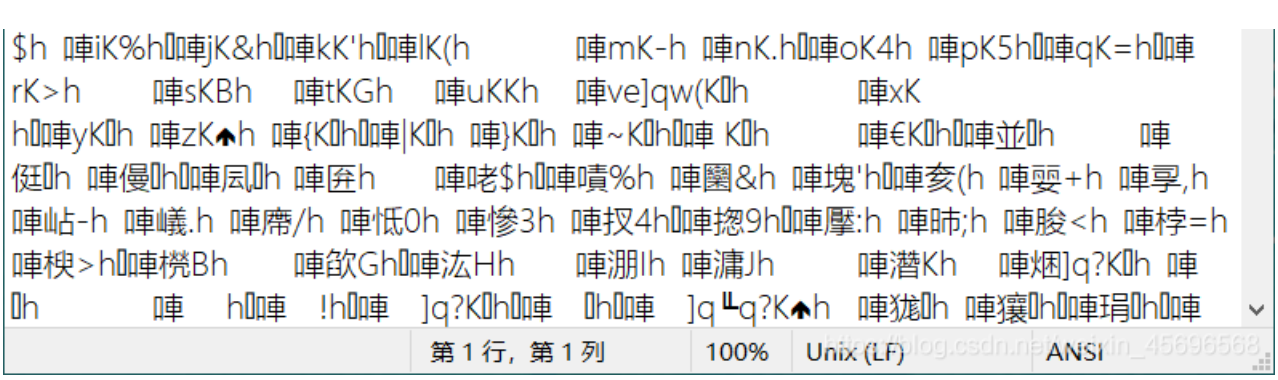


定位到jpg尾部之后，发现后面的数据比较奇怪，看了大佬的wp得知这种数据格式是Python的序列化文件



把这些数据提取出来，保存在1.txt中

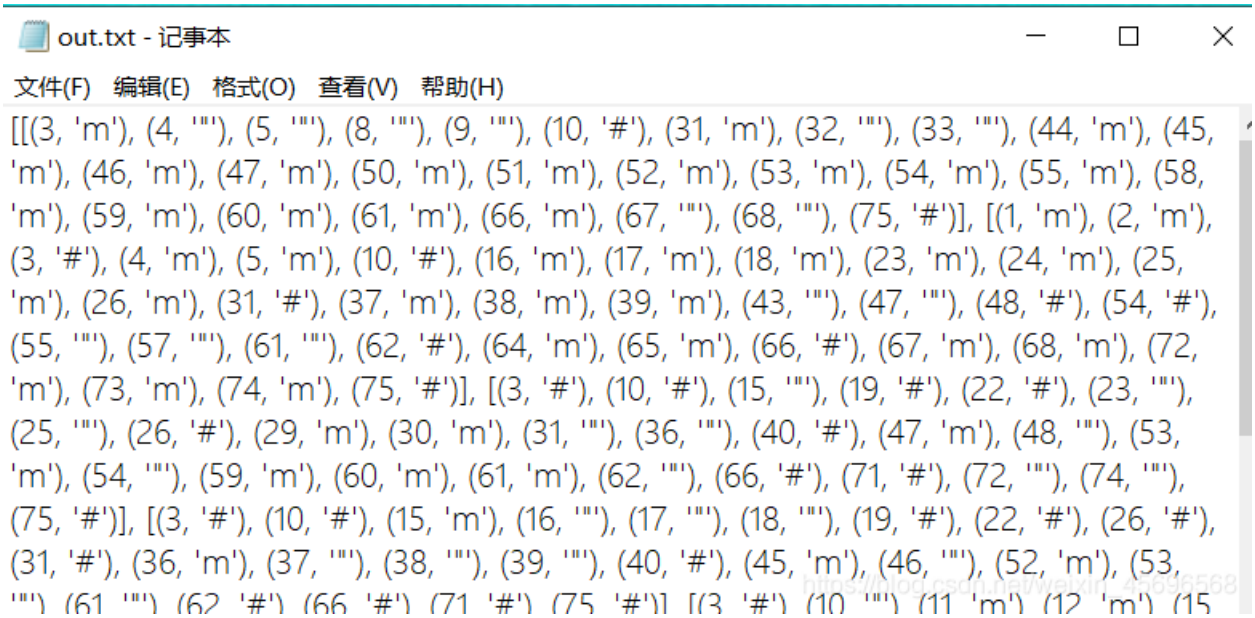




找到一篇Python反序列化的文章：https://jingyan.baidu.com/article/0bc808fcb7f91e5bd585b97d.html，跟着步骤写脚本

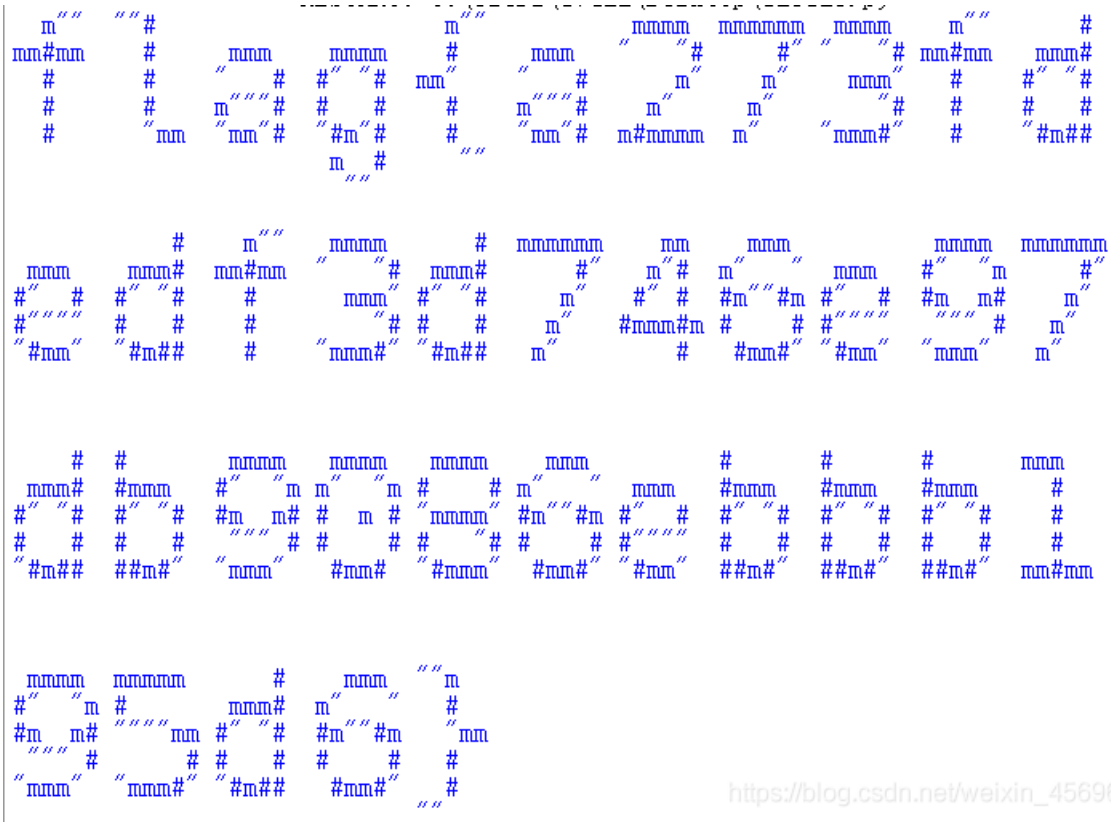
```
import pickle
f=open("1.txt","rb")
result=pickle.load(f)
f.close()
f1=open("out.txt","w")
f1.write(str(result))
f1.close()
```

得到一些坐标：



写一手脚本给它画出来：

```
f=open("out.txt","r")
txt=eval(f.read())
for i in range(len(txt)):
    s=""
    for j in range(1,77):
        n=0
        for k in txt[i]:
            if k[0] == j:
                n=1
                break
        if n:
            s+=k[1]
        else:
            s+=" "
    print(s)
f.close()
```




```
flag{a273fdedf3d746e97db9086ebbb195d6}
```