攻防世界MISC刷题1-50

原创

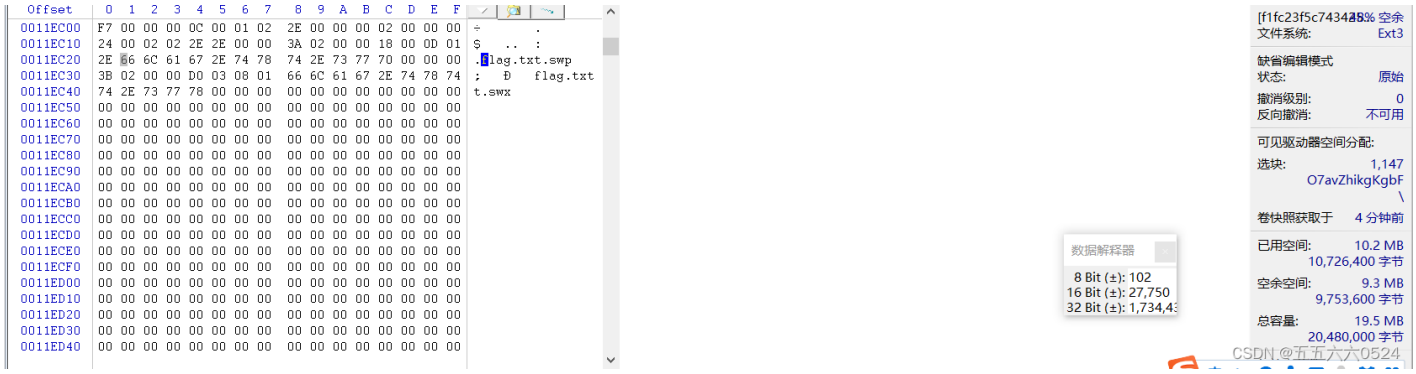
五五六六0524 已于 2022-03-02 20:56:25 修改 3043 收藏 3
分类专栏： CTF积累及刷题 文章标签： 安全
于 2022-01-11 16:52:03 首次发布

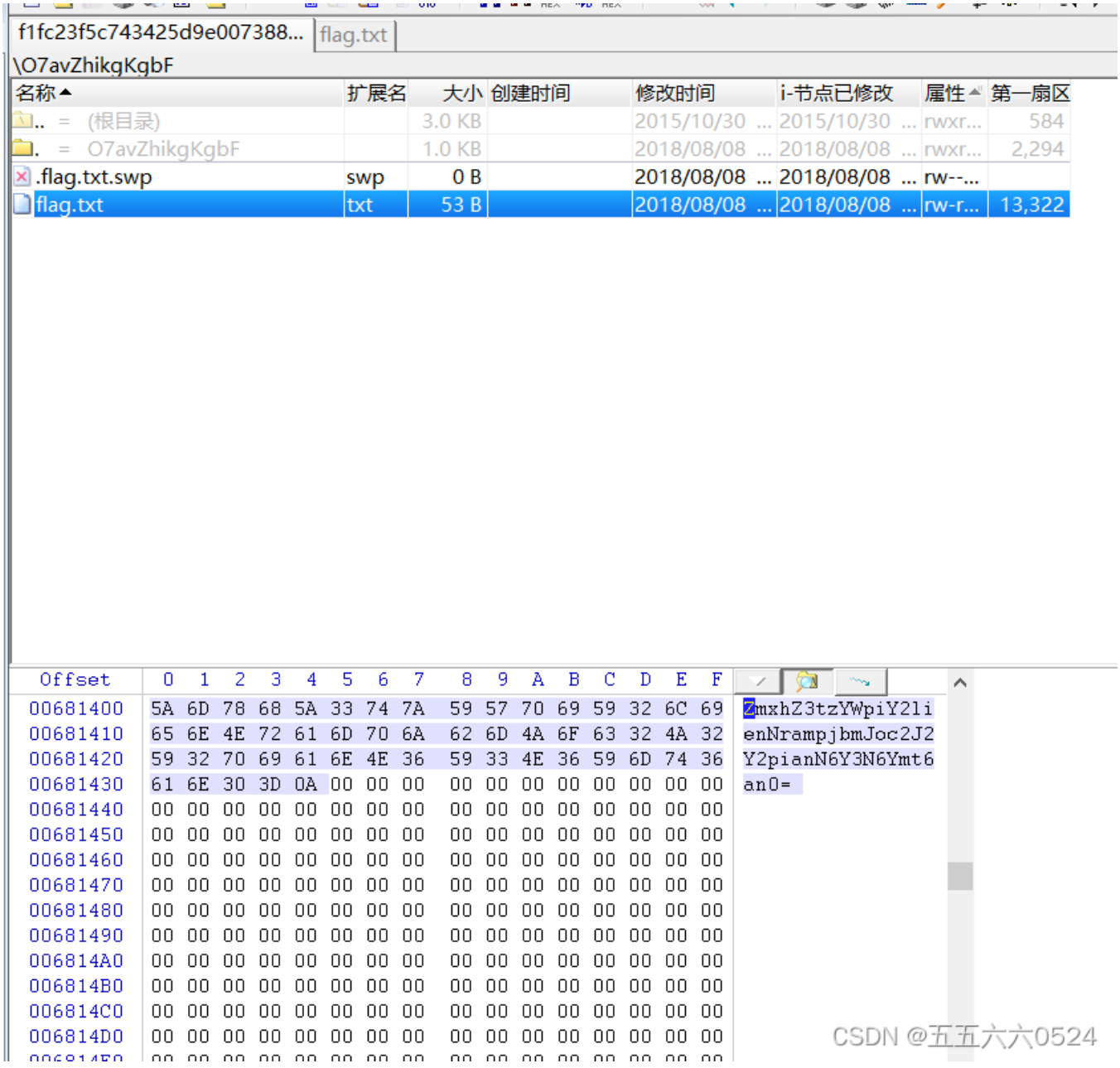

CTF积累及刷题 专栏收录该内容
6 篇文章 0 订阅
订阅专栏

目录

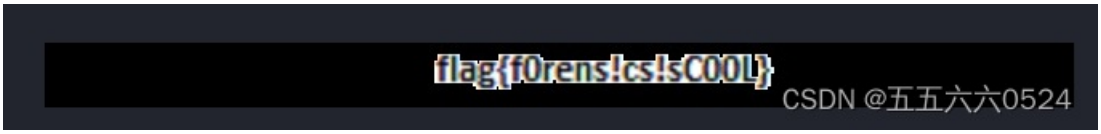
1、ext3

用winhex打开，搜索flag能查到，这个是linux磁盘，专业工具-将镜像转为磁盘，再次搜索flag，在最右边能看到文件位置，打开后就能直接找到flag，base64解密一下即可，flag{sajbcibzskjjcnbhsbvcjbjszcszbkzj}



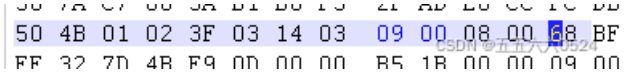


比较奇怪的是，把这个linux拖进kali后，foremost一下，能得到一张照片，里面也是flag，但是提交进去不对，不太懂这是干什么



2、base64stego

压缩包加密，爆破一下发现不行，怀疑是伪加密，winhex打开后果然



txt里面是一串base64，解密之后看起来像说明书，肯定不对，在网上找了个base64的脚本，得到答案

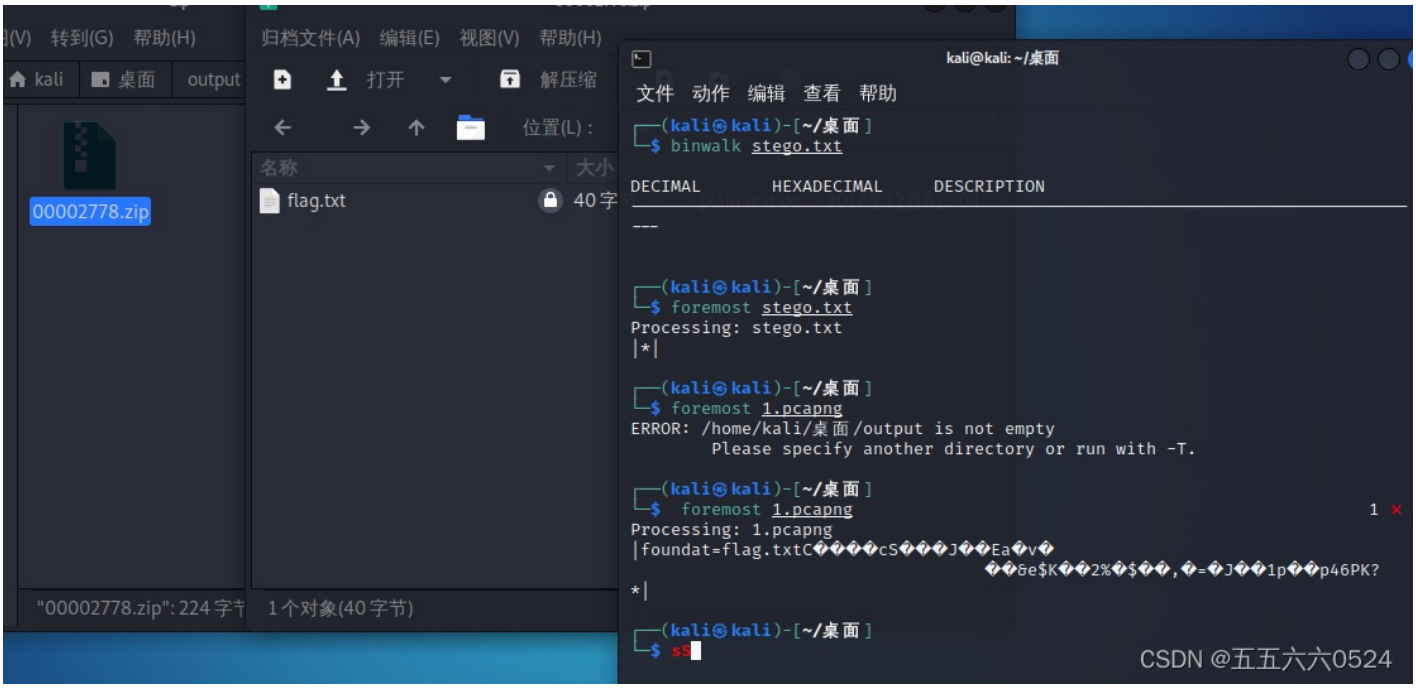
CSDN @五五六六0524

第四行的路径记得要改成绝对路径，flag{Base_sixty_four_point_five}

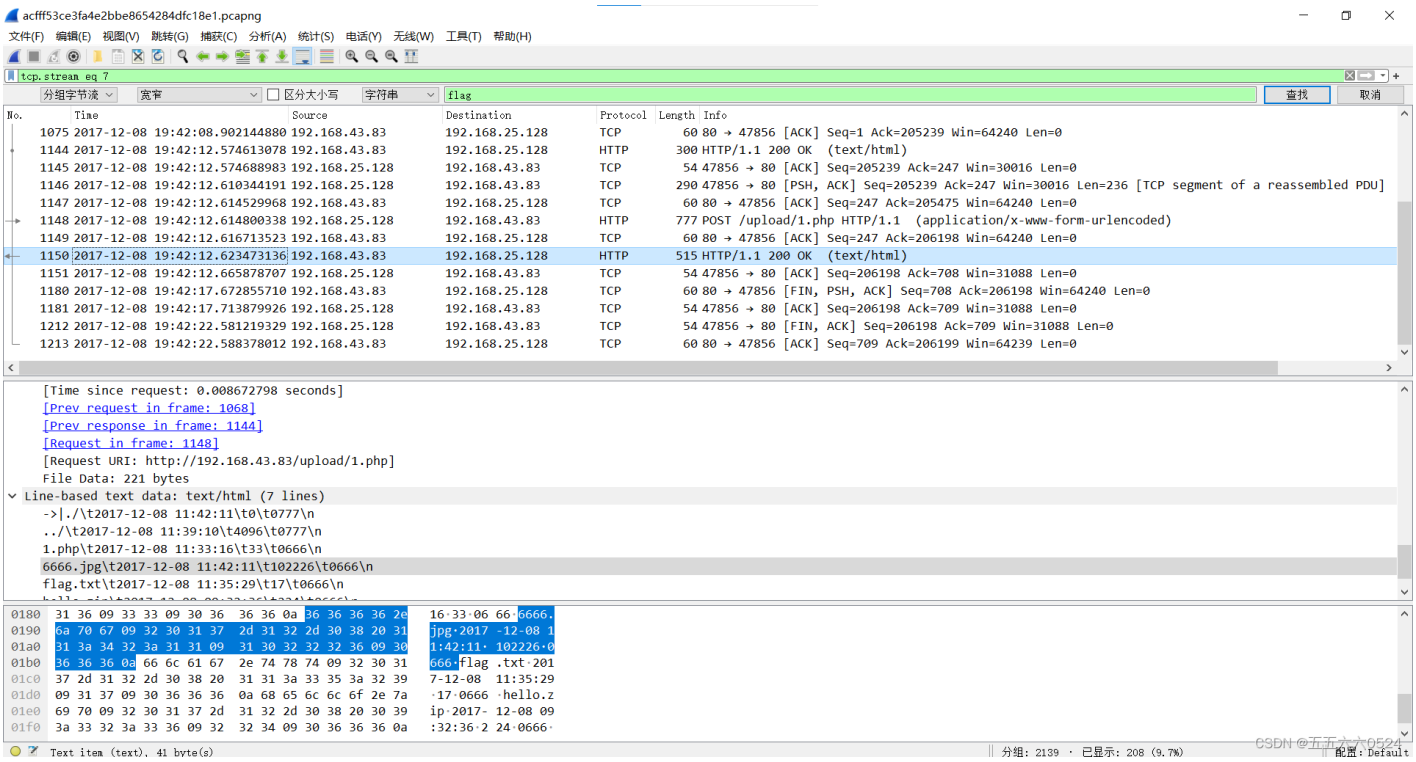
```
import base64
bin_str=''
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
with open('stego.txt','r') as f:
 for line in f.readlines():
  stegb64="".join(line.split())
  rowb64="".join(str(base64.b64encode(base64.b64decode(stegb64)),'utf-8').split())
  offset=abs(b64chars.index(stegb64.replace('','')[-1])-b64chars.index(rowb64.replace('=', '')[-1]))
  equalnum=line.count('=')
  if equalnum:
   bin_str += bin(offset)[2:].zfill(equalnum * 2)
 print(''.join([chr(int(bin_str[i:i + 8], 2)) for i in range(0,len(bin_str),8)]))
```

3、功夫再高也怕菜刀

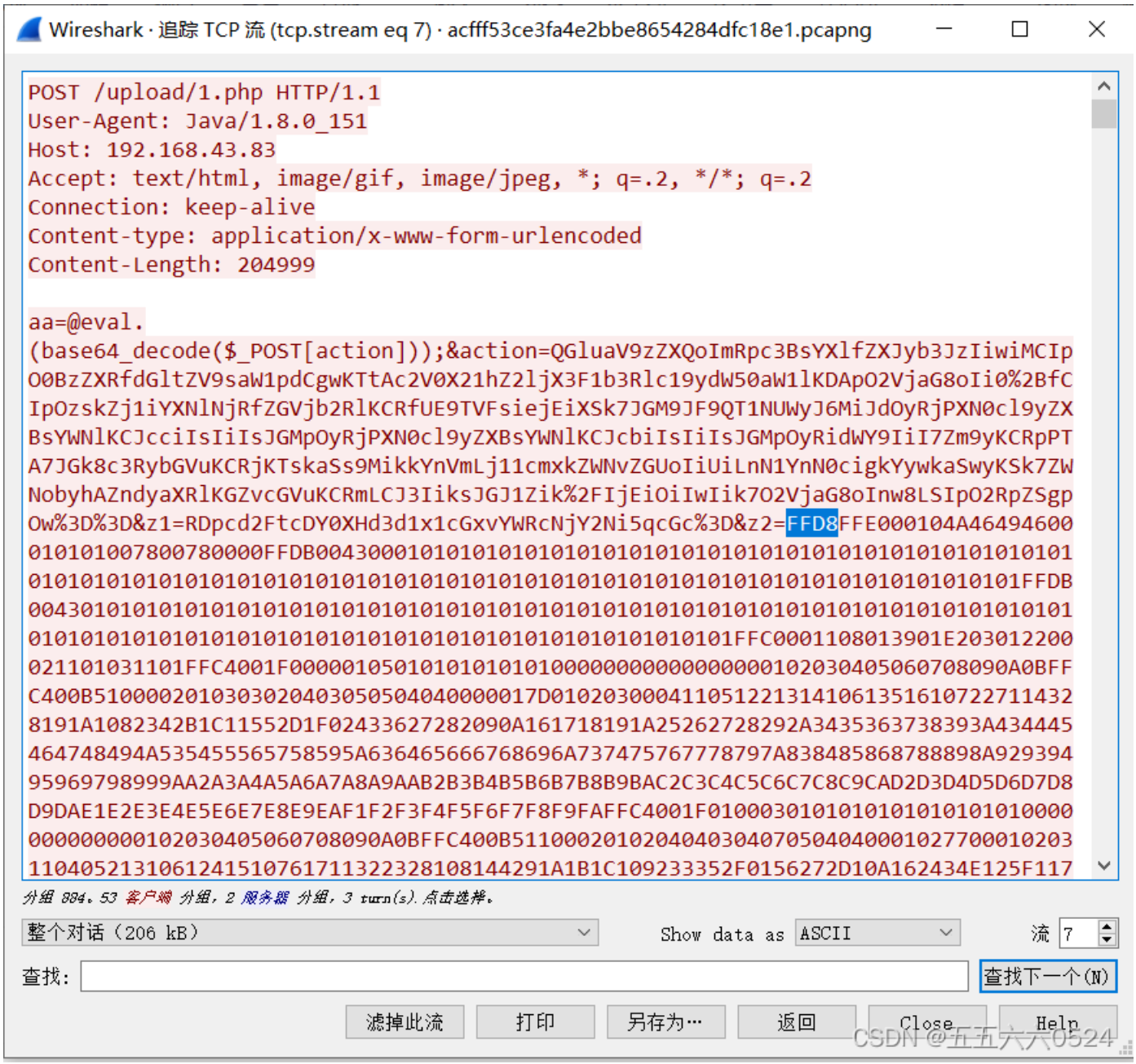
pcapng格式，一看就是用wireshark，把它拖到kali里foremost一下，得到压缩包，最终是一个加密的flag.txt



在wireshark中搜索flag.txt，只有1150最奇葩，有一个666.jpg



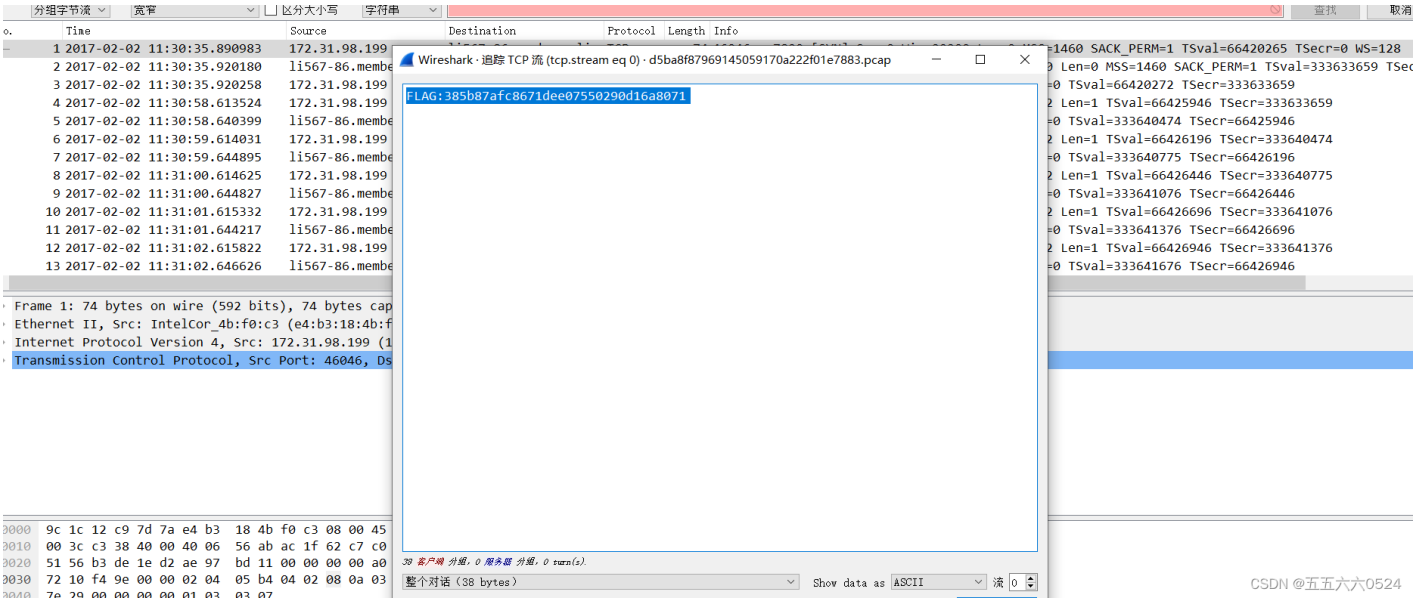
然后追踪TCP流，发现有FFD8 FFD9，把整个复制下来



打开winhex，编辑-剪切板数据-粘贴至新文件-ASCII HEX-保存为jpg就行了，得到flag.txt的解压密码，Th1s_1s_p4sswd_!!!，flag{3OpWdJ-JP6FzK-koCMAK-VkfWBq-75Un2z}

4、easycap

这一题简单的有点不可思议，随便一行追踪一下TCP流就行了，不用改为中括号，改了就不对了，FLAG:385b87afc8671dee07550290d16a8071

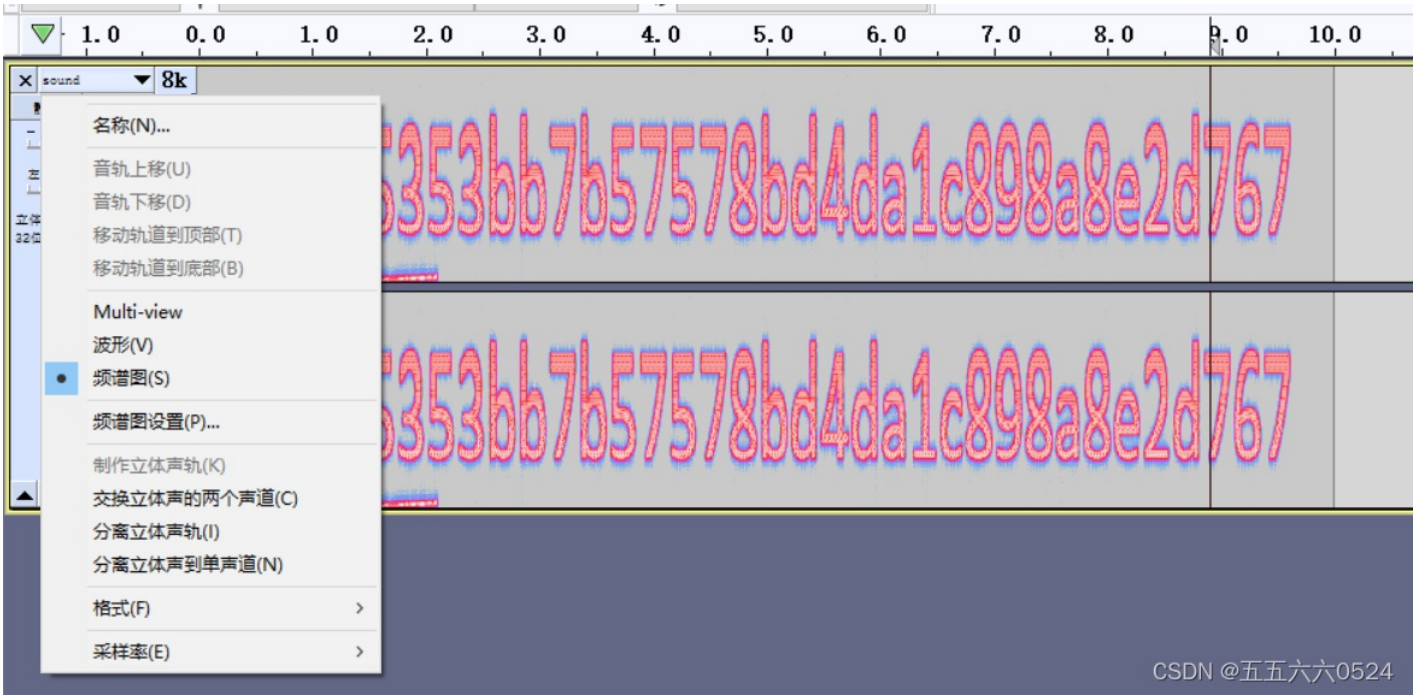


5、reverseMe

这一题也很离谱，旋转一下就行了，华为的提取文字真香，直接出flag{4f7548f93c7bef1dc6a0542cf04e796e}

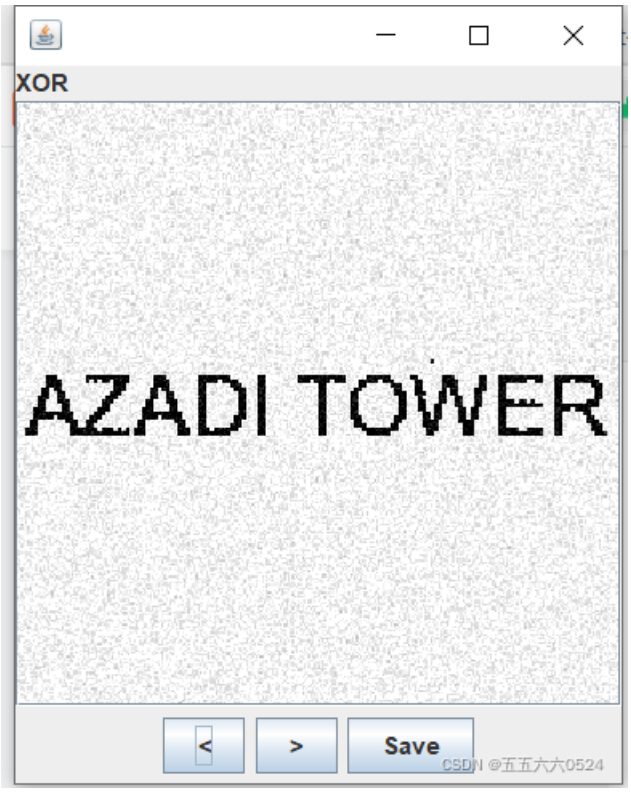
6、Hear-with-your-Eyes

Bandzip能直接解压，得到一段音频wav，在Audacity中打开，看不出有什么不一样，根据题意我还以为是摩斯密码，但是形状对不上，点那个小三角，切换到频道图，flag就出来了，奇葩的是这个只要提交后面的字符串，全输进去还不对



7、What-is-this

打开之后是两张图片，根据名称猜想肯定是合并，用copy命令不行，得到的图片没有，binwalk一下感觉有隐藏，但是提取不出来，百度了一下，需要用到stegsolve的图片合并功能，先打开第一张图，然后Analyse-Image Combiner得到flag，AZADI TOWER



8、normal-png

打开是图片，没看出来啥，百度了一下发现他的高度被改了，03改成04就行了，
flag{B8B68DD7007B1E406F3DF624440D31E0}

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
00000010	00	00	02	6C	00	00	03	6B	08	06	00	00	00	36	B4	F5
00000020	FD	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B
00000030	13	01	00	9A	9C	18	00	00	0A	4D	69	43	43	50	50	68
00000040	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66

flag{B8B68DD7007B1E406F3DF624440D31E0}

如何判断图片的高宽度被修改了？

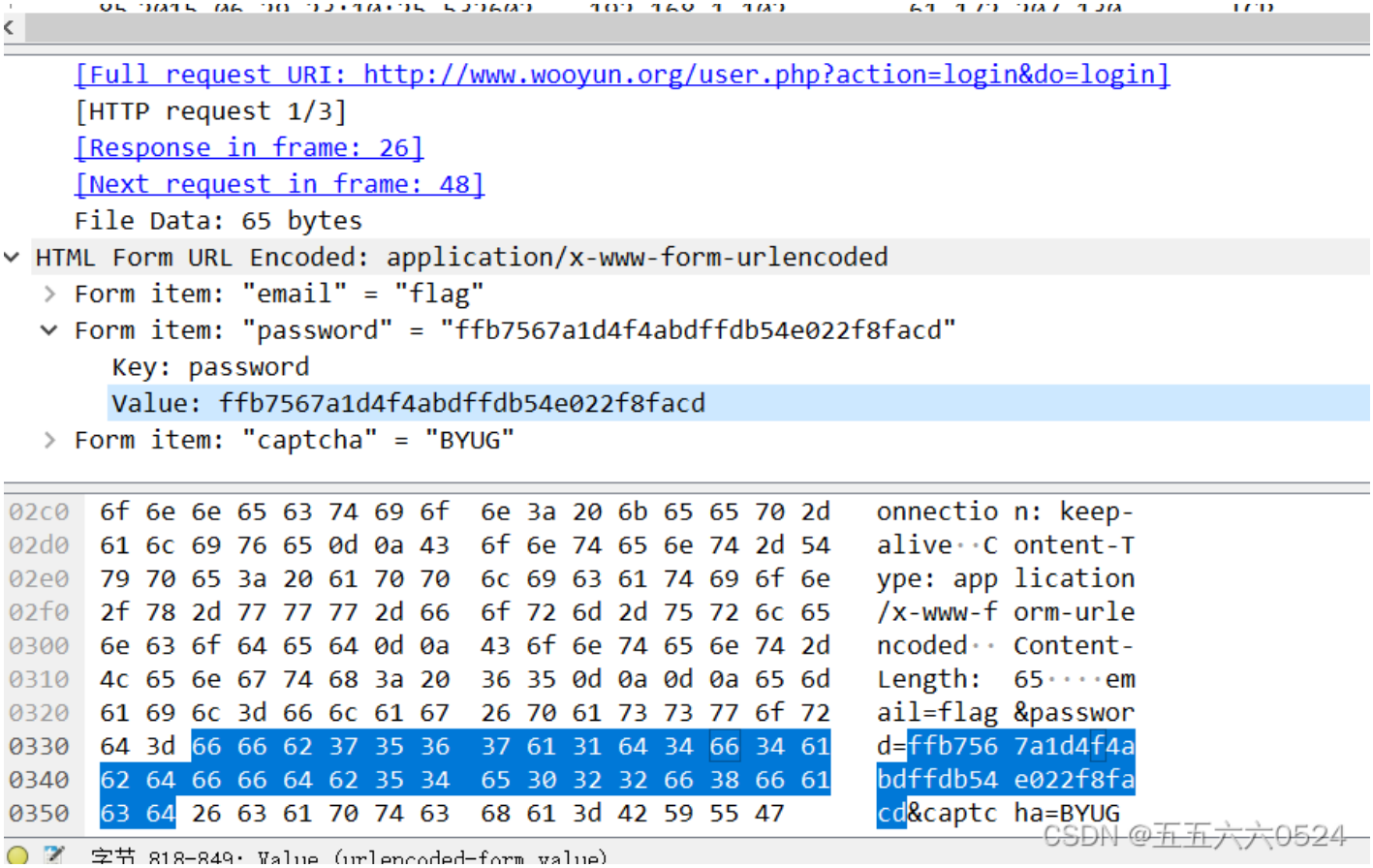
9、something in image

在linux里解压，然后直接搜索，strings badimages | grep -i Flag（-i不分大小写），得到Flag{yc4pl0fvjs2k1t7T}

10、wireshark-1

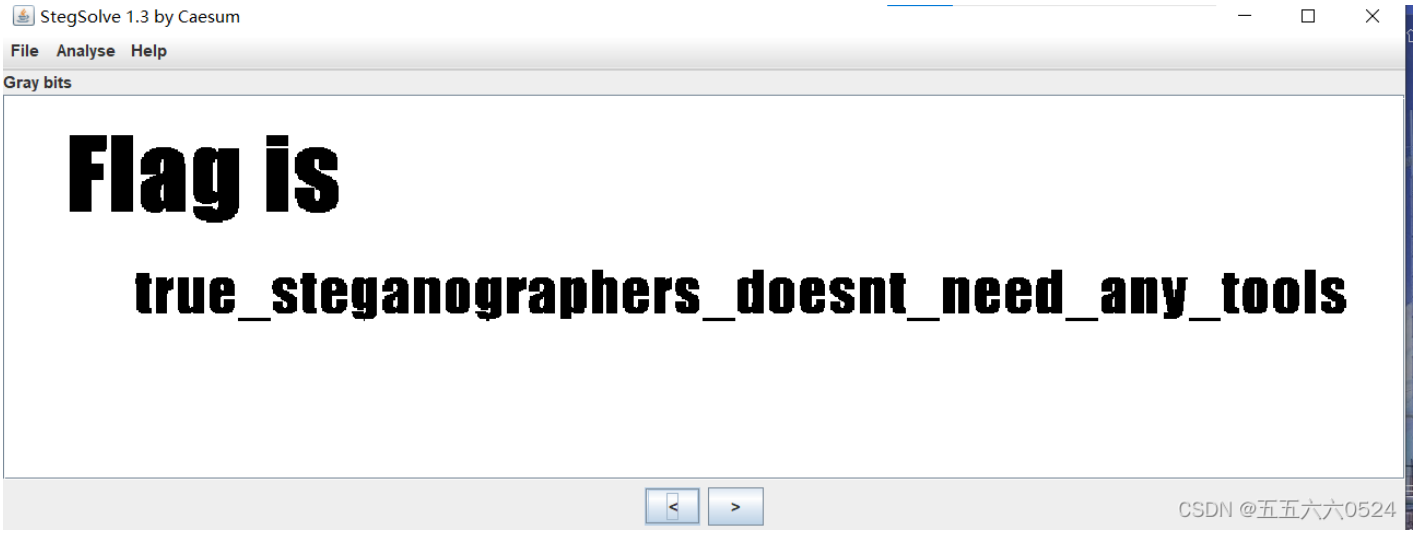
> 题目：黑客通过wireshark抓到管理员登陆网站的一段流量包（管理员的密码即是答案)。 flag提交形式为flag{XXXX}

搜password一搜就出来了



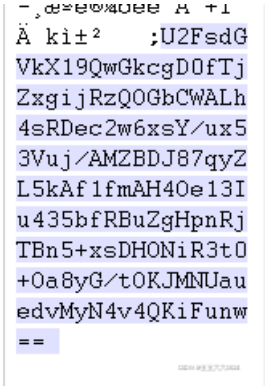
11、 pure_color

foremost什么都试了，没想到这玩意居然直接点一下就出来了，
flag{true_steganographers_doesnt_need_any_tools}太离谱啦



12、Aesop_secret

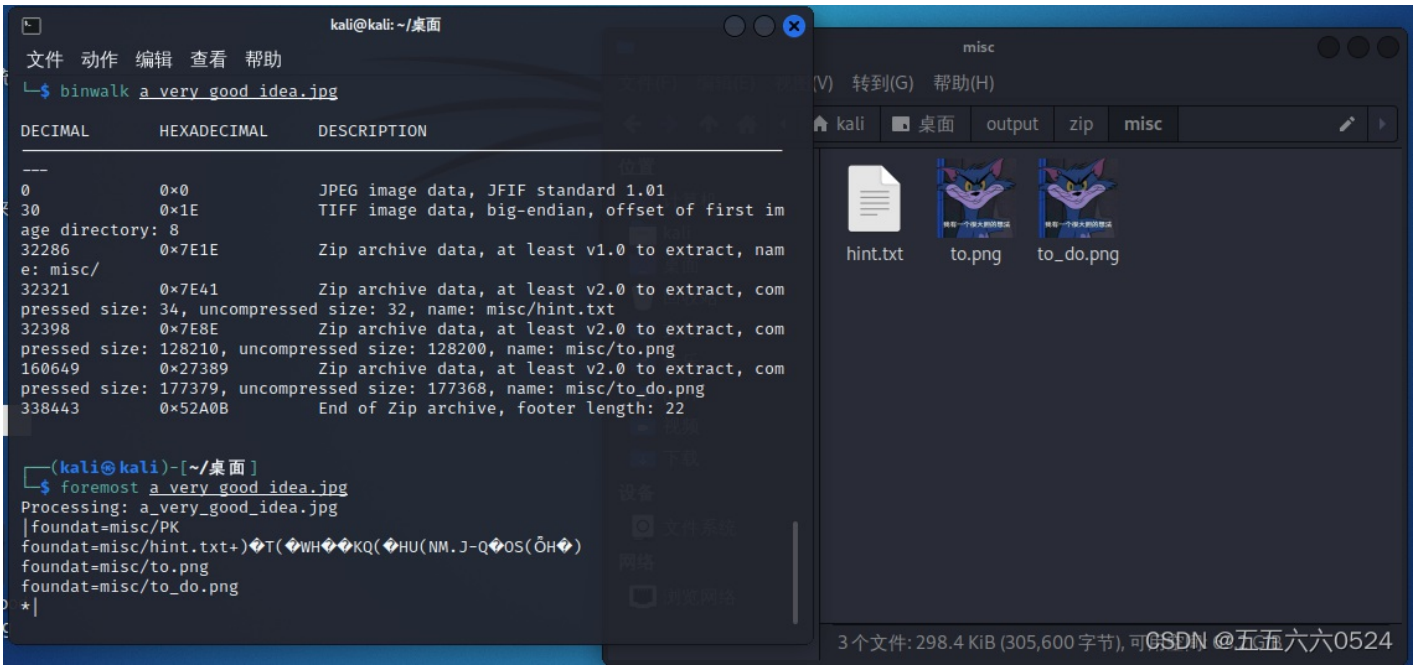
看的WP攻防世界misc进阶区Aesop_secret_gongjingege的博客-CSDN博下载附件，解压后，打开是一张gif动图，因为没有下载ps啥的，直接在线分解的gif动图分解：https://tu.sioe.cn/gj/fenjie/差不离应该是ISCC到这啥也没有了，用winhex打开，看一下信息在最后发现了一串字符串，以为是b64，结果没解出来看了wp，才知道aes加密（第一次知道。。。），而ISCC就是密钥（其实题目也有提示aes operation secret）解密两次得到flag这个题题目就是提示，没看出来，蓝瘦了，铁子2020.8.17 公瑾..
https://blog.csdn.net/gongjingege/article/details/108059130先把动图分解GIF动态图片分解，多帧动态图分解成多张静态图片_图片工具网页版，得到一个ISCC，winhex打开动图，最后的一部分看起来不太对劲



根据题目提示是AES解密，密钥就是ISCC，解两次得flag{DugUpADiamondADeepDarkMine}

13、a_good_idea

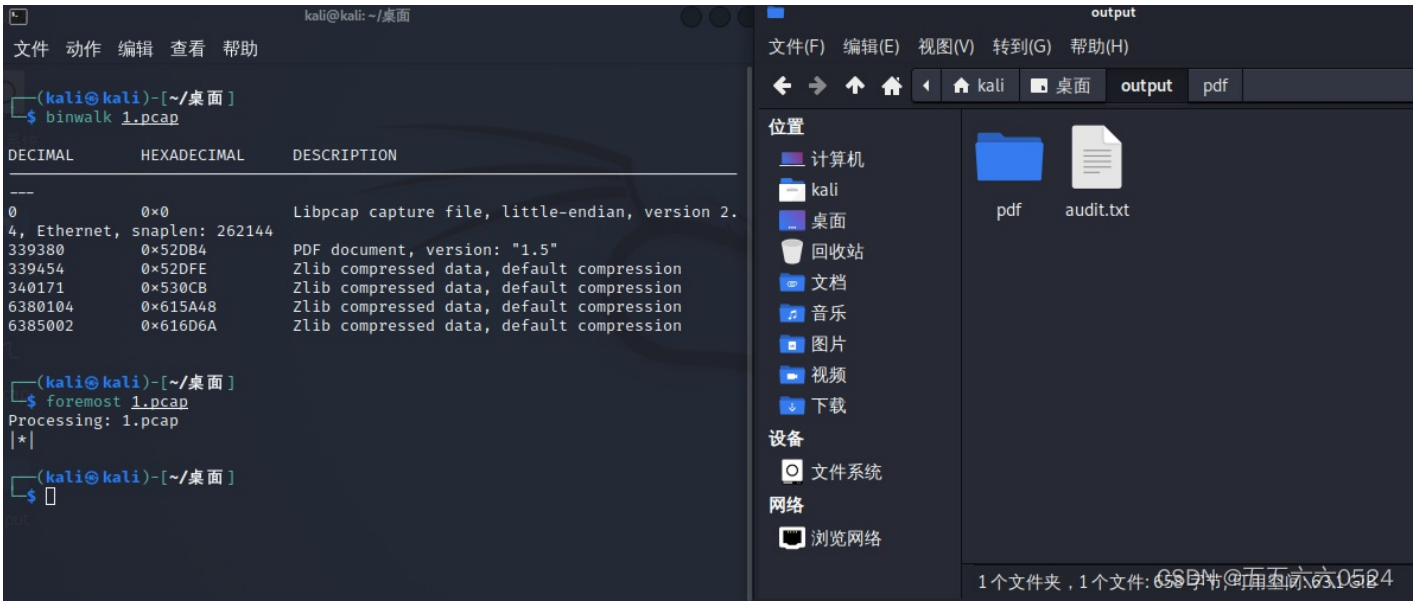
得到一张图，binwalk发现里面有东西，foremost提取出来，hint.txt里面有提示“try to find the secret of pixels”，从像素中寻找秘密，根据两张图片的名称，猜想将这两张合起来



用stegsolve的图片合并功能，先打开第一张图，然后Analyse-Image Combiner，点两下箭头直接得到二维码，得NCTF{m1sc_1s_very_funny!!!}

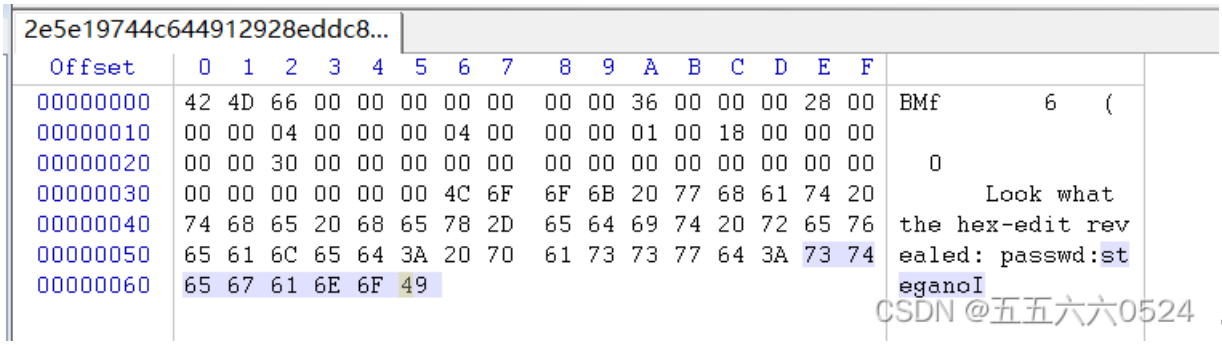
14、simple_transfer

pcap格式，用wireshark找半天没有找到什么有用的东西，最后binwalk了一下，发现里面有东西，foremost提取出来一个pdf文档，直接出flag，HITB{b3d0e380e9c39352c667307d010775ca}



15、Training-Stegano-1

用winhex打开，直接出，steganol



16、2017_Dating_in_Singapore

这一题的脑洞真的大的离谱，完全想不到日历还可以这么玩，详见这位大佬的博客【XCTF 攻防世界】杂项 misc 高手进阶区 2017_Dating_in_Singapore_Kal1的博客-CSDN博客

17 、can_has_stdio?

下载下来的misc50，不知道文件类型，winhex打开，发现全是+><组成，很特殊

```
00000210  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000220  20 20 20 20 20 20 20 20  0A 20 20 20 20 20 20 20
00000230  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000240  20 20 20 20 20 20 20 20  20 20 20 20 2B 2B 2B 3E                +++>
00000250  2B 2B 20 20 20 20 20 20  20 20 20 20 20 20 20 20  ++
00000260  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000270  20 20 20 20 20 20 20 0A  20 20 20 20 20 20 20 20
00000280  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000290  20 20 20 20 20 20 20 20  20 20 20 2B 2B 3E 2B 2B              ++>++
000002A0  2B 2B 20 20 20 20 20 20  20 20 20 20 20 20 20 20  ++
000002B0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000002C0  20 20 20 20 20 20 0A 20  20 20 20 20 20 20 20 20
000002D0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000002E0  20 20 20 20 20 20 20 20  20 2B 3E 2B 2B 2B 2B 2B           +>+++++
000002F0  2B 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  +
00000300  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000310  20 20 20 20 20 0A 20 20  20 20 20 20 20 20 20 20
00000320  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000330  20 20 20 20 20 20 20 20  3E 2B 2B 2B 2B 2B 2B 2B          >+++++++
00000340  3E 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20  >
00000350  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000360  20 20 20 20 0A 20 20 20  20 20 20 20 20 20 20 20
00000370  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000380  20 20 20 20 20 20 2B 2B  2B 2B 2B 2B 2B 2B 3E 2B        ++++++++>+
00000390  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000003A0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000003B0  20 20 20 0A 20 20 20 20  20 20 20 20 20 20 20 20
000003C0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000003D0  20 20 20 20 20 2B 2B 2B  2B 2B 2B 2B 2B 3E 2B 2B       ++++++++>++
000003E0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
000003F0  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000400  20 20 0A 20 20 20 20 20  20 20 20 20 20 20 20 20
00000410  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000420  20 20 20 2B 2B 2B 2B 2B  2B 2B 2B 3E 2B 2B 2B 20     ++++++++>+++
00000430  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000440  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000450  20 0A 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000460  20 20 20 20 20 20 20 20  20 20 20 20 20 20 20 20
00000470  20 20 2B 2B 2B 2B 2B 2B  2B 2B 3E 2B 2B 2B 2B 20    ++++++++>++++
```


没见过这样的加密，看了别人的wp才知道这是Brainfuck加密，复制下来的图形刚好还是一个五角星，解密得flag{esolangs_for_fun_and_profit}

18、János-the-Ripper

下载下来得misc100，winhex打开发现隐藏了一个flag.txt，加后缀.zip，解压需要密码，爆破一下，密码是fish，解压出flag{ev3n::y0u::bru7us?!}

misc50 misc100

```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
00000000  50 4B 03 04 14 00 03 00  08 00 0E A2 77 44 44 D4  PK          ¢wDDÔ
00000010  88 77 27 00 00 00 19 00  00 00 08 00 00 00 66 6C  ▮w'             fl
00000020  61 67 2E 74 78 74 00 10  01 4B 93 FF 03 EE 9C FA  ag.txt    K▮ÿ î▮ú
00000030  D3 12 83 A1 57 88 57 8C  BF 41 AA 41 87 16 F6 85  Ó ▮¡W▮W▮¿AªA▮ ö▮
00000040  FE 40 02 DA 73 CA 1F AC  16 97 89 44 3A 50 4B 01  þ@ ÚsÊ ¬ ▮▮D:PK
00000050  02 14 00 14 00 03 00 08  00 0E A2 77 44 44 D4 88            ¢wDDÔ▮
00000060  77 27 00 00 00 19 00 00  00 08 00 00 00 00 00 00  w'
00000070  00 01 00 20 00 00 00 00  00 00 00 66 6C 61 67 2E             flag.
00000080  74 78 74 50 4B 05 06 00  00 00 00 01 00 01 00 36  txtPK          6
00000090  00 00 00 4D 00 00 00 00  00                        M
```

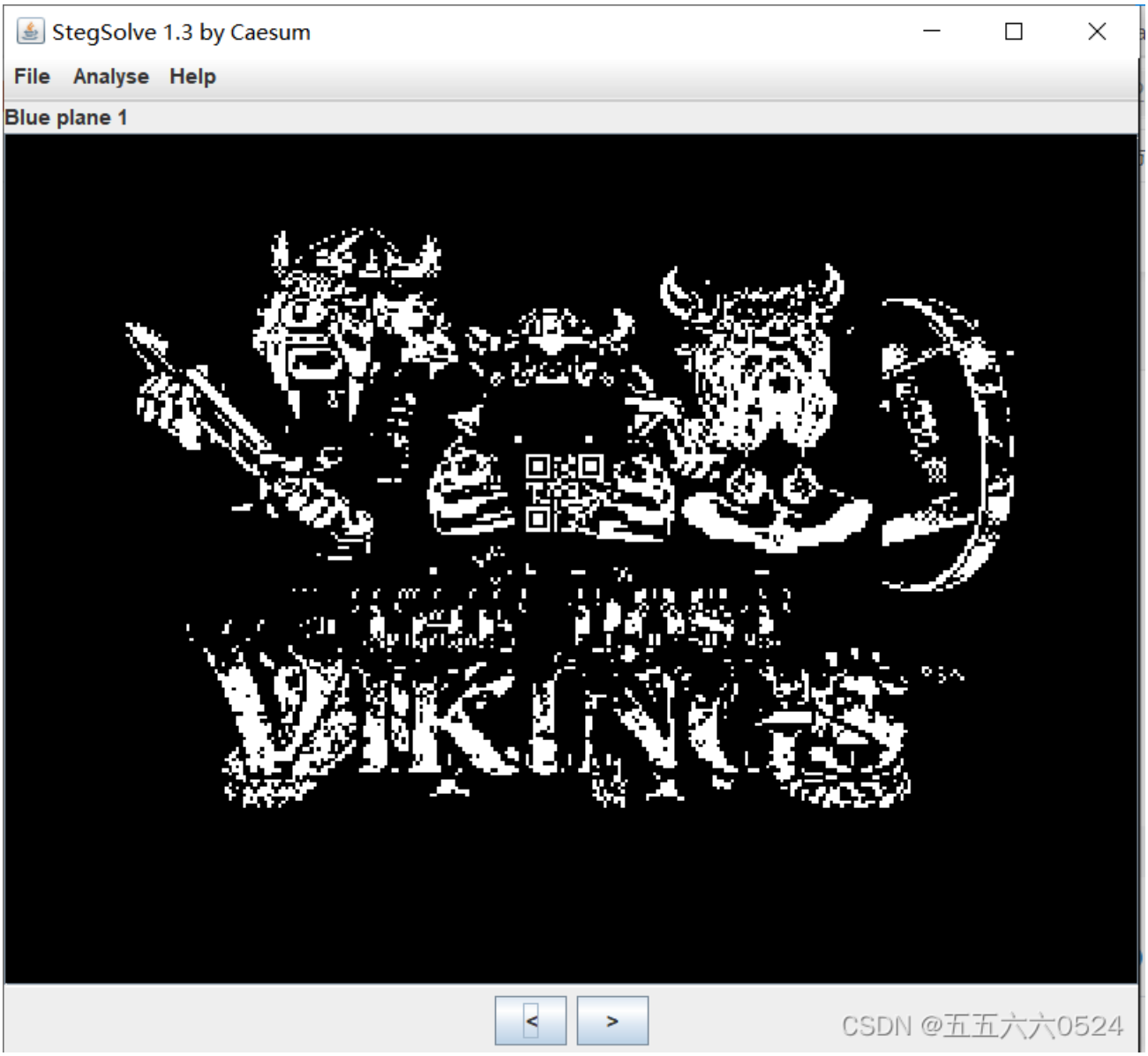

19、Erik-Baleog-and-Olaf

stego100用winhex打开发现是png格式，改后缀

stego100

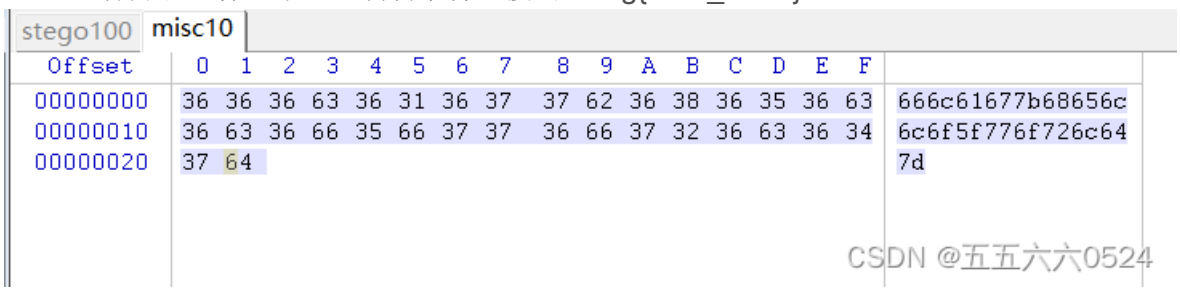
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000010	00	00	02	80	00	00	01	E0	08	02	00	00	00	BA	B3	4B	▌ à º³K
00000020	B3	00	00	68	81	49	44	41	54	78	DA	ED	BD	6F	72	DB	³ h IDATxÚí½orÛ
00000030	4C	BA	E5	09	57	F4	26	FC	C1	DD	60	F5	2E	1C	F1	C6	Lºå Wô&üÁÝ`õ. ñÆ
00000040	0C	B5	0D	45	79	5A	56	EF	62	14	8E	09	B7	EE	2E	46	µ EyZVïb ▌ ·î.F
00000050	56	B7	2B	D4	BB	B8	62	77	BC	11	DE	C5	14	71	EF	FB	V·+Ô»¸bw¼ ÞÅ qïû
00000060	41	BB	A8	9A	62	9E	84	70	12	4F	3E	C9	04	41	0A	12	A»¨šb▌▌p O>É A
00000070	75	7E	1F	68	38	09	24	12	09	50	91	07	CF	BF	A6	11	u~ h8 $ P‘ Ï¿¦
00000080	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B ▌! B ▌! B ▌! B
00000090	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	▌! B ▌! B ▌! B

得到一张图片，根据文件名，肯定是要用stegslove的，点几下箭头出现一个二维码，如果不仔细看还真不好看出来，直接扫扫不出来，还得p一下，把右下角的方块补齐，flag{#justdiffit}



20、Test-flag-please-ignore

winhex打开长这样，十六进制转字符直接出，flag{hello_world}



21、hit-the-core

放到kali里，binwalk没有，strings +文件名，查看文件内的字符串，四位一个提取大写字母，ALEXCTF{K33P_7H3_g00D_w0rk_up}

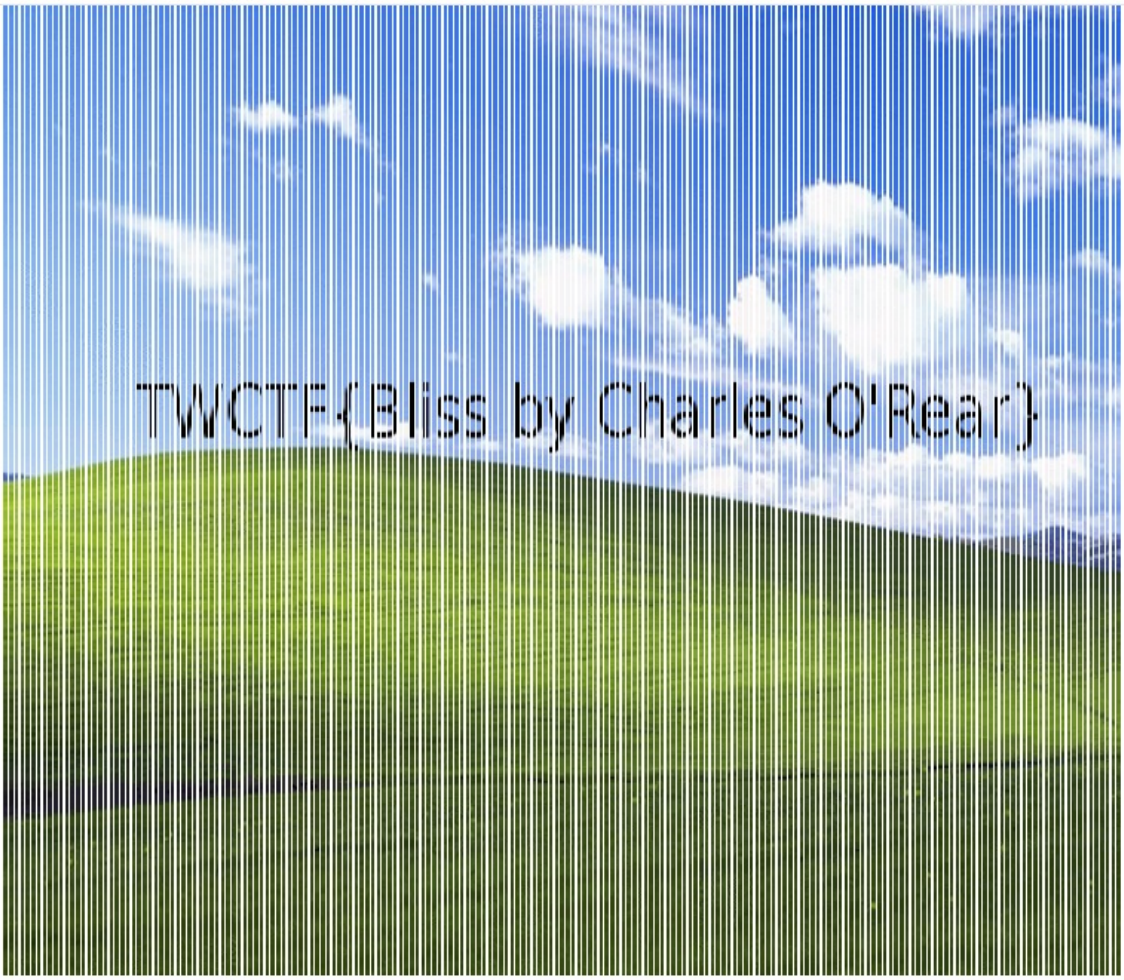
22、快乐游戏题

下载下来在我的电脑上显示是恶意软件，没敢运行，应该就是游戏通关得flag，看了一眼wp，√

UNCTF{c783910550de39816d1de0f103b0ae32}

23、glance-50

动图分解一下就行了，TWCTF{Bliss by Charles O'Rear}



2×600px（201张）

题目是一张动漫图，总感觉高度不对，04改成05，得到一个StRe1izia

Ditf.pcapng | e02c9de40be145dba6baa80e...

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000010	00	00	03	9E	00	00	05	4C	08	02	00	00	00	38	16	5A	L 8 Z
00000020	34	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	4 pHYs
00000030	13	01	00	9A	9C	18	00	00	06	D4	69	54	58	74	58	4D	ÔiTXtXM

StRe1izia

把原图放进kali里binwalk一下，发现有东西，foremost提取出来一个rar，爆破解不开，把上面那个字符串扔进去，解出来一个pcapng，剩下的不太会，根据wp，搜索png，得到一串字符ZmxhZ3tPel80bmRfSGlyMF9sb3YzX0ZvcjN2ZXJ9，

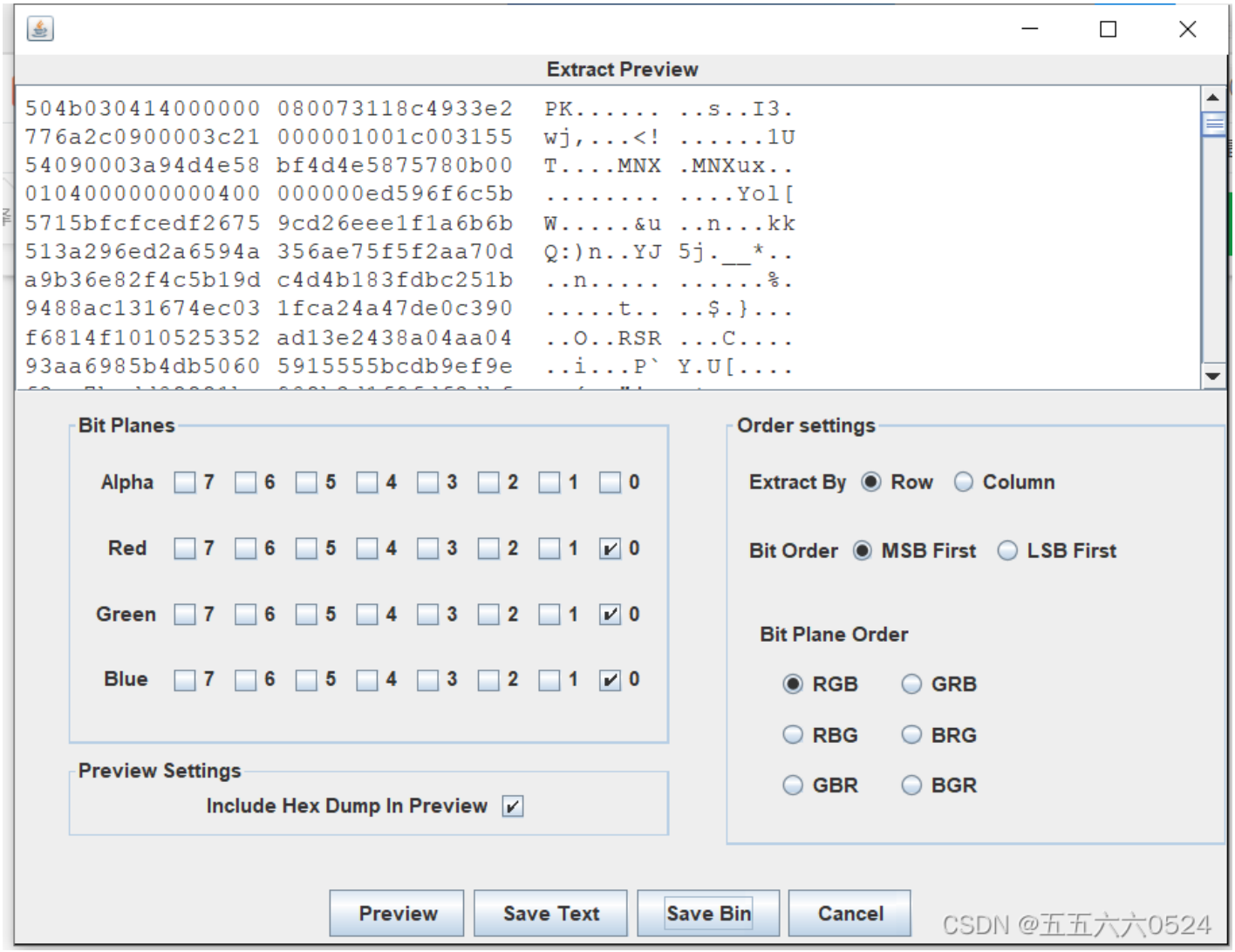
```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
       <img src="/kiss.png" />
  ZmxhZ3tPel80bmRfSGlyMF9sb3YzX0ZvcjN2ZXJ9
  </body>
</html>

GET /kiss.png HTTP/1.1
```


Zmxh的开头明显就是flag的base64密文，base64转一下就行了，flag{Oz_4nd_Hir0_lov3_For3ver}

24、misc_pic_again

winhex打开没有发现什么，binwalk尝试无果，扔stegslove里面，箭头看了一圈无果，RGB改为0，发现压缩包的文件头，savebin为压缩包，解压后winhex，根据题目搜索得到，hctf{scxdc3tok3yb0ard4g41n~~~}

```
000005B0  41 5E 41 5F C3 66 66 2E  0F 1F 84 00 00 00 00 00  A^A_Aff.  ▮
000005C0  F3 C3 00 00 48 83 EC 08  48 83 C4 08 C3 00 00 00  óÃ  H▮ì H▮Ä Ã
000005D0  01 00 02 00 00 00 00 00  68 63 74 66 7B 73 63 78          hctf{scx
000005E0  64 63 33 74 6F 6B 33 79  62 30 61 72 64 34 67 34  dc3tok3yb0ard4g4
000005F0  31 6E 7E 7E 7E 7D 00 00  00 00 01 1B 03 3B 30 00  1n~~~}        ;0
00000600  00 00 05 00 00 00 04 FE  FF FF 7C 00 00 00 44 FE          þÿÿ|   Dþ
00000610  FF FF 4C 00 00 00 31 FF  FF FF A4 00 00 00 54 FF  ÿÿL   1ÿÿÿ¤   Tÿ
```

25、Banmabanma

这个也是相当坑，zsteg +文件名，出来一大堆标红的东西，没想明白是啥玩意，搜了一下wp，这个题居然是条形码，Barcode Reader. Free Online Web ApplicationRead Code39, Code128, PDF417, DataMatrix, QR, and other barcodes from TIF, PDF and other image documents https://online-barcode-reader.inliteresearch.com/扫一下，没有想到，flag{TENSHINE}

Free Online Barcode Reader

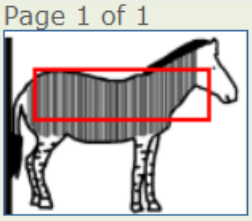
To get such results using ClearImage SDK use TBR Code 103.

If your **business** application needs barcode recognition capabilities,
email your technical questions to support@inliteresearch.com
email your sales inquiries to sales@inliteresearch.com

File: 斑马斑马.png New File
Pages: 1 **Barcodes:** 1

Barcode: 1 of 1 **Type:** Code39
Length: 16 **Rotation:** none
Module: 1.6pix **Rectangle:** {X=71,Y=93,Width=410,Height=119}

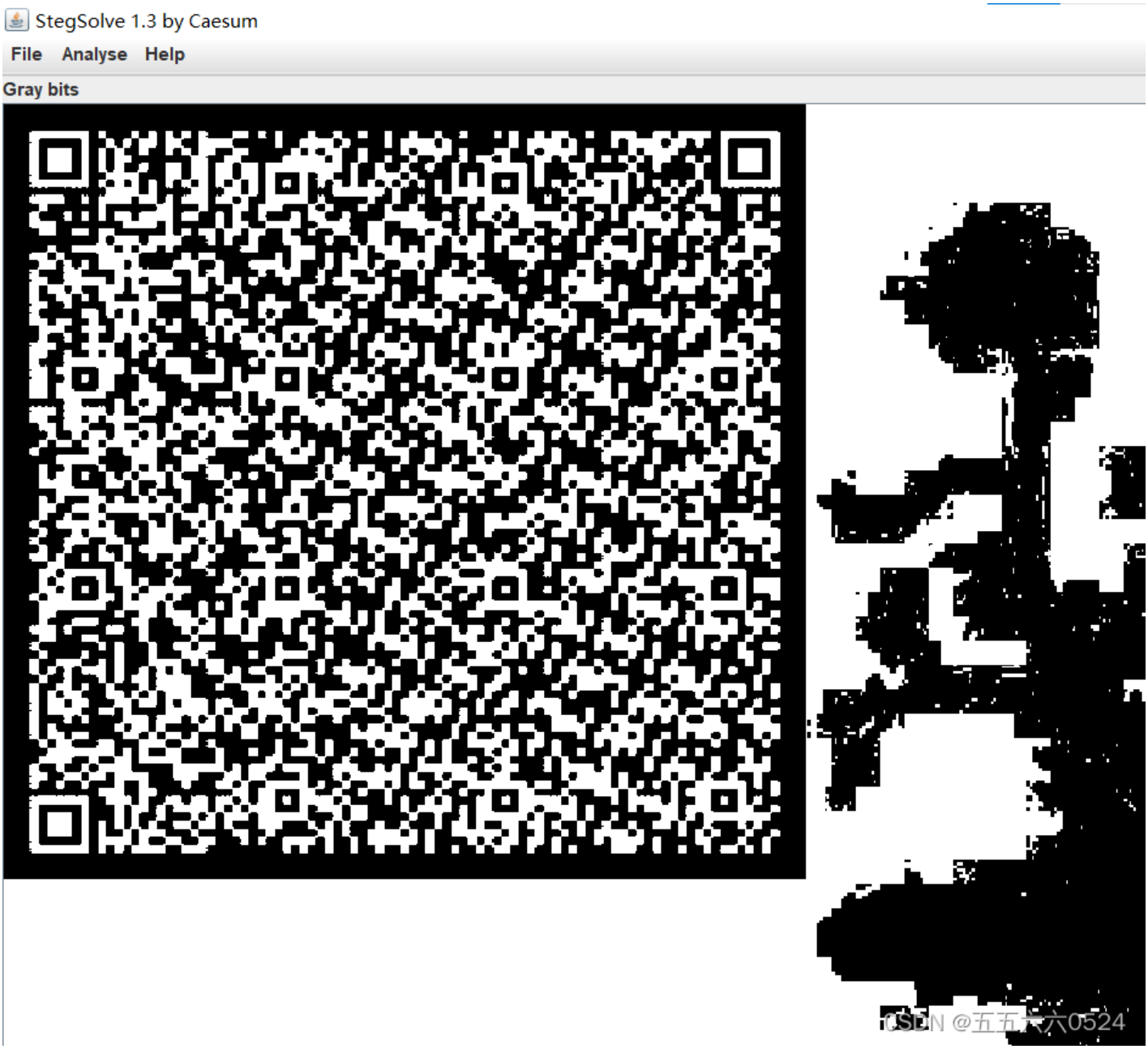
Page 1 of 1



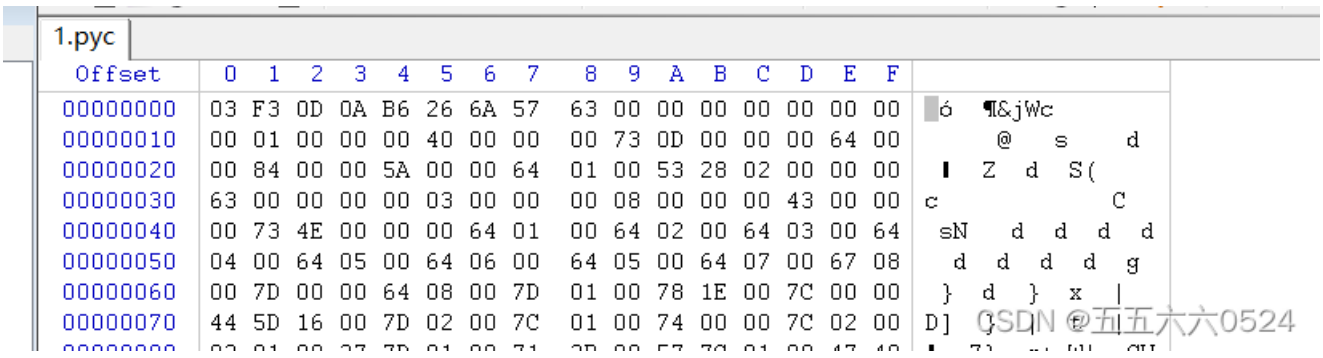
FLAG IS TENSHINE

26、stage1

还是熟悉的图片，stegslove得到，扫一下出来一堆十六进制的数



用winhex保存一下没看出来啥，看wp说文件头03F30D0A，这个是.pyc文件



需要进行反编译，可以用python反编译 - 在线工具，直接就能看到代码，也可以用Easy Python Decompiler软件，反编译之后得到*.pyc_dis文件，我没有下载文本编译器，拖进kali里就能直接打开看到了，长这样

改一下把数值扔进去就是

```
str = [
 65,
        108,
        112,
        104,
        97,
        76,
        97,
        98]
flag = '03F30D0AB6266A57630000000000000000001000000400000007300000064000084000005A000064010053280200000063
for i in str:
    flag += chr(i)

print(flag)
```

直接出AlphaLab，不用加flag

27、Miscellaneous-200

像素点python成图，引自攻防世界-进阶区-Miscellaneous-200_shadowland_L的博客-CSDN博客

```
from ast import literal_eval as make_tuple
from PIL import Image
f = open(r'.txt', 'r')
corl = [make_tuple(line) for line in f.readlines()]
f.close()
img0 = Image.new('RGB', (270, 270), '#ffffff')
k=0
for i in range(246):
   for j in range(246):
      img0.putpixel ([i , j], corl[k])
      k=k+1
img0.save("result.png")
```

28、red_green

zsteg隐写，发现不对，提取该通道图片，直接出flag{134699ac9d6ac98b}

```
┌──(kali㉿kali)-[~/桌面]
└─$ zsteg 1.png                                                    1 ×
b1,r,lsb,xy         .. file: JPEG image data, JFIF standard 1.01, aspect rati
o, density 100×100, segment length 16, Exif Standard: [TIFF image data, big-e
ndian, direntries=10, manufacturer=Canon, model=Canon EOS DIGITAL REBEL XTi,
orientation=upper-left, xresolution=168, yresolution=176, resolutionunit=2, s
oftware=Adobe Photoshop CS2 Windows, datetime=2010:02:03 08:]
b1,bgr,lsb,xy       .. text: "E\"QE\"QI$"

┌──(kali㉿kali)-[~/桌面]
└─$ zsteg -e b1,r,lsb,xy 1.png -> out.png

┌──(kali㉿kali)-[~/桌面]
└─$ s
```


> zsteg使用，引自隐写工具zsteg安装+使用教程_Amherstieae的博客-CSDN博客_zsteg
>
> 1.查看lsb数据
>
> zsteg xxx.bmp
> zsteg xxx.png
> zsteg -a (文件名)　#查看各个通道的lsb
>
> 2.检测zlib
>
> #-b的位数是从1开始的
> zsteg zlib.bmp -b 1 -o xy -v
>
> 3.提取该通道图片
>
> zsteg -e b8,a,lsb,xy 文件.png -> out.png

29、 Recover-Deleted-File

引自Linux用extundelete恢复磁盘文件-攻防世界Recover-Deleted-File_半岛铁盒的博客-CSDN博客

linux删除文件时其实删的是文件名，数据还是存储在硬盘中的，恢复用extundelete命令，

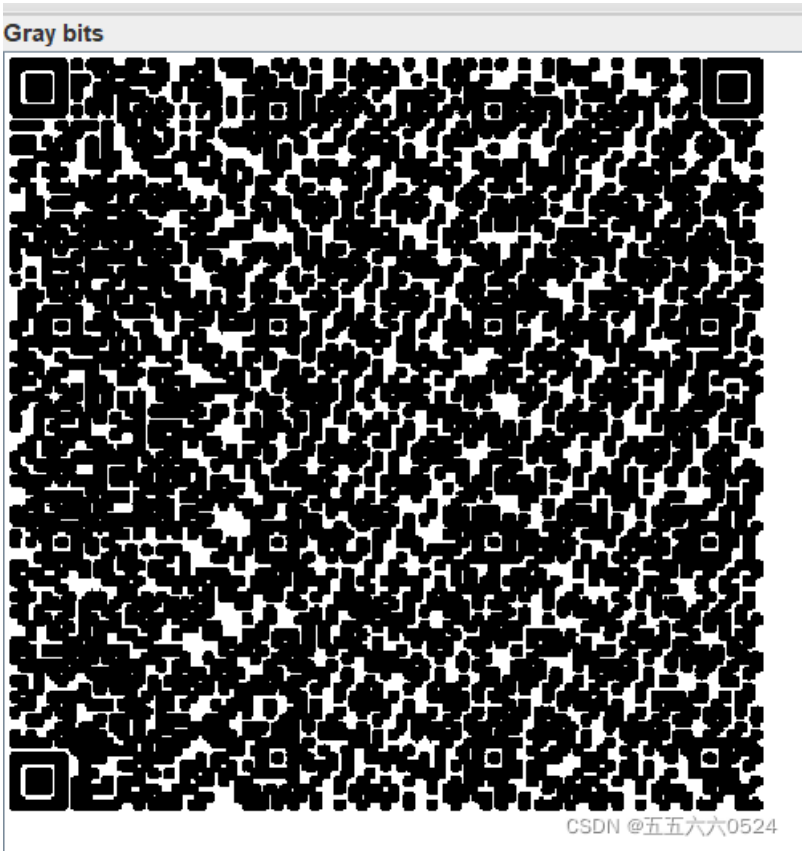
```
安装：sudo apt-get install extundelete
```

```
应用：extundelete 文件名 --restore-all
```

题目是一个.gz的文件，扔进kali里解压得到disk-image，恢复文件“extundelete disk-image --restore-all”，得到一个文件夹，里面有一个flag文件，运行一下就能得到flag，在运行flag时，用“./ +文件名”命令，但是显示权限不够，网上搜了一下，“chmod 777 +文件名”可以解决这个问题， flag是de6838252f95d3b9e803b28df33b4baa

30、适合作为桌面

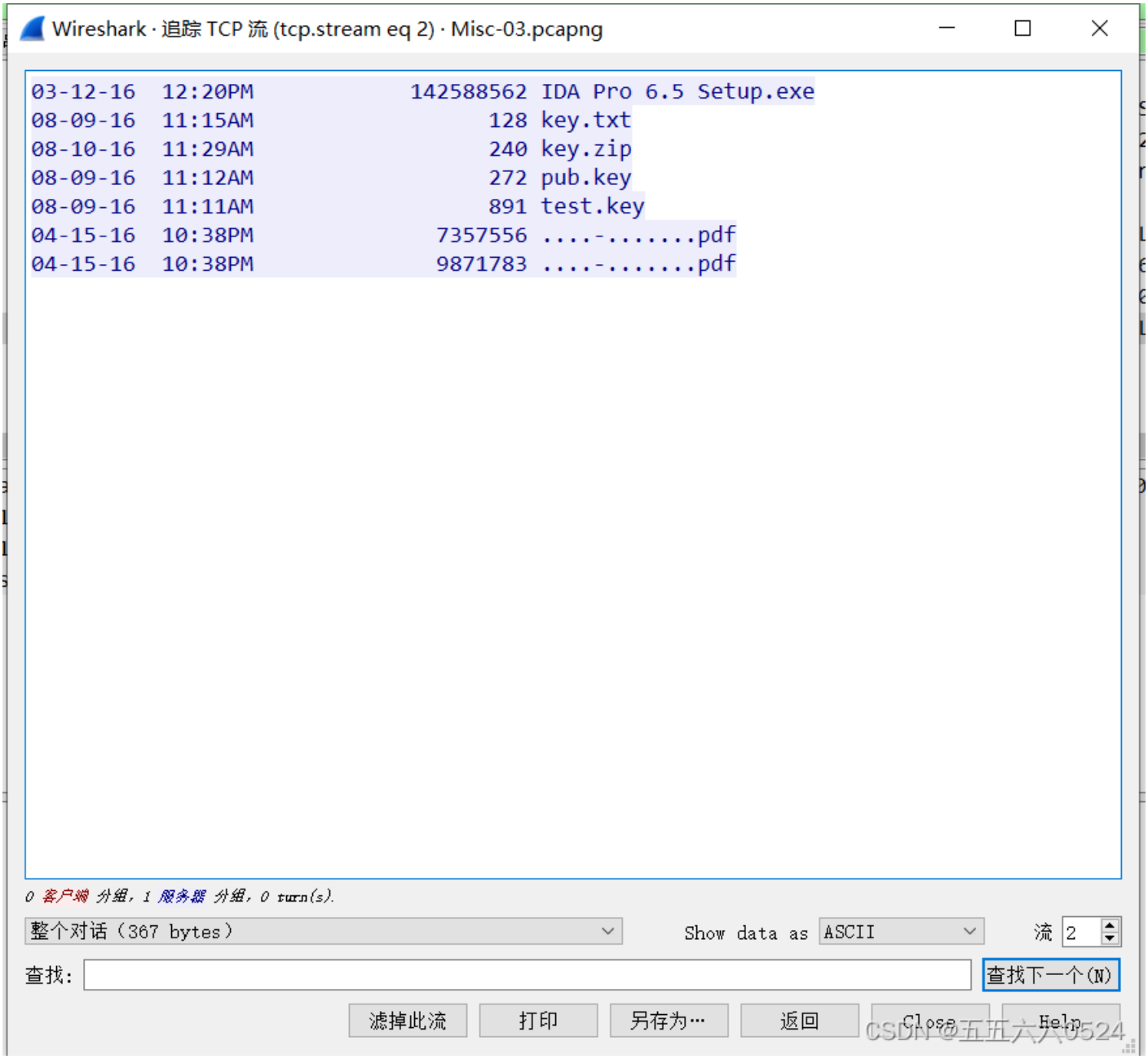
stegslove得到一张二维码，关键是我扫不出来啊，wp攻防世界Misc 适合作为桌面_hhh-CSDN博客_攻防世界适合作为桌面，flag{38a57032085441e7}



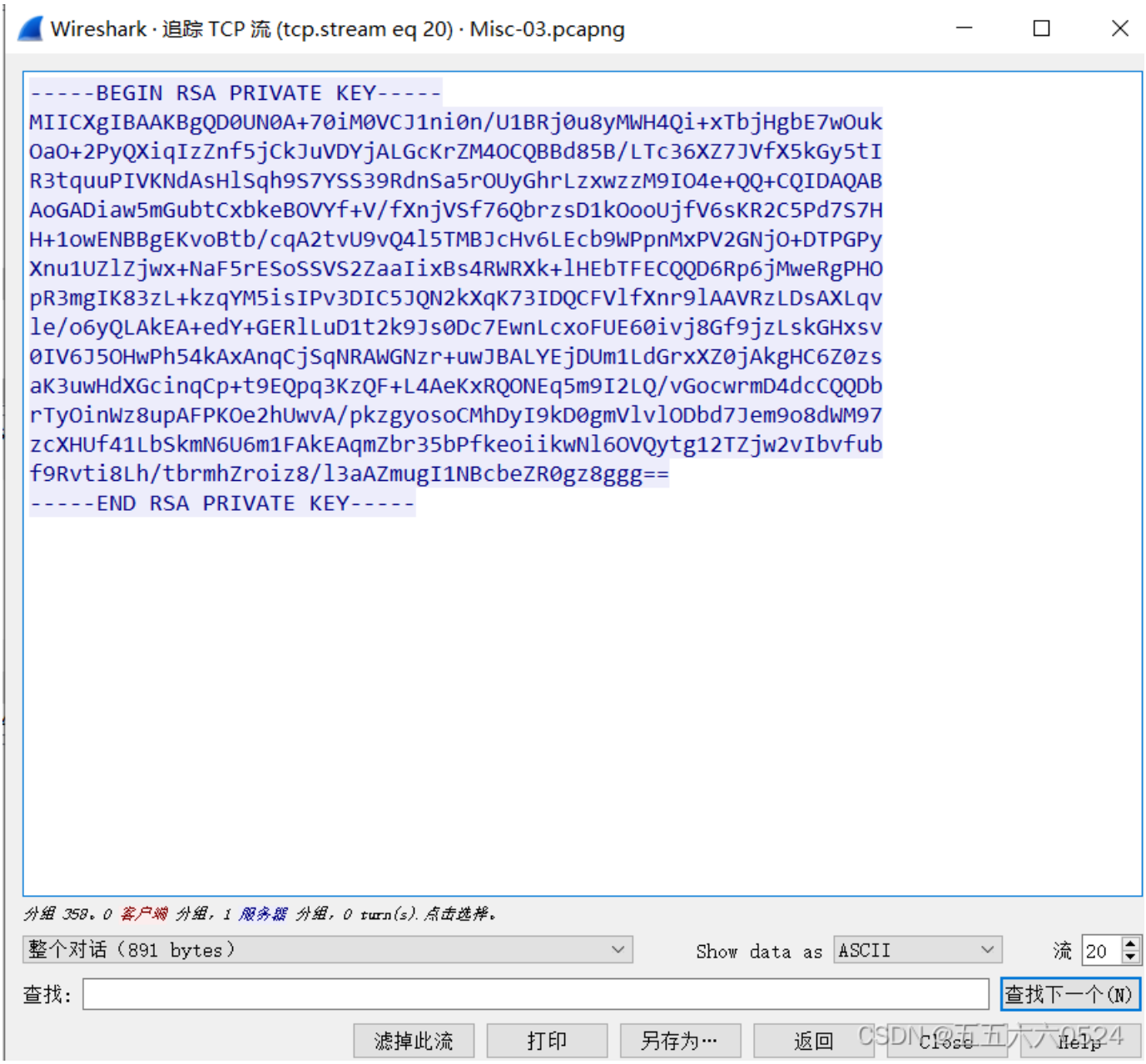
31、就在其中

攻防世界 MISC 高手进阶区 就在其中WP 解题思路_C w h的博客-CSDN博客_攻防世界就在其中

TCP（Transmission Control Protocol传输控制协议）是一种面向连接的，可靠的，基于字节流的传输层通信协议，FTP（文件传输协议）是网络共享文件的传输协议 。有TCP、FTP应该是在传输某些文件，在追踪第二个流时发现一些文档，pub.key公钥，推测有私钥



搜索private，找到私钥，保存下来，保存为pub.key



foremost流量包会得到jpg、pdf、zip，zip解压后得到key.txt，用openssl解密，flag is {haPPy_Use_0penSsl}

> 命令：openssl rsautl -decrypt -in key.txt -inkey pub.key -out flag.txt
> -in 为要解密的加密文档 -inkey 为密钥 -out 为输出文档
>
> 具体的详见如何运用OpenSSL 对文件进行加密和解密_petpig0312的博客-CSDN博客_openssl加密文件

32、base64÷4

base16解密，flag{E33B7FD8A3B841CA9699EDDBA24B60AA}

33、 很普通的数独

nono这一题一点也不简单，写脚本那一点给我卡的死死的，flag{y0ud1any1s1}

看大佬的wp吧攻防世界 Misc高手进阶区 3分题 很普通的数独_闵行小鱼塘-CSDN博客_攻防世界很普通的数独

34、再见李华

foremost提取出来一个zip，根据题目的意思，有4位，后面还要加上“LiHua”，用ARPHR的掩码攻击，设置掩码为“????LiHua”，那个问号是英文的，如果是中文问号，会提示“错误：未找到掩码符号”，这个真坑，我捯饬了好久，破解出密码是15CCLiHua，flag为Stay hungry, Stay foolish.

ARCHPR

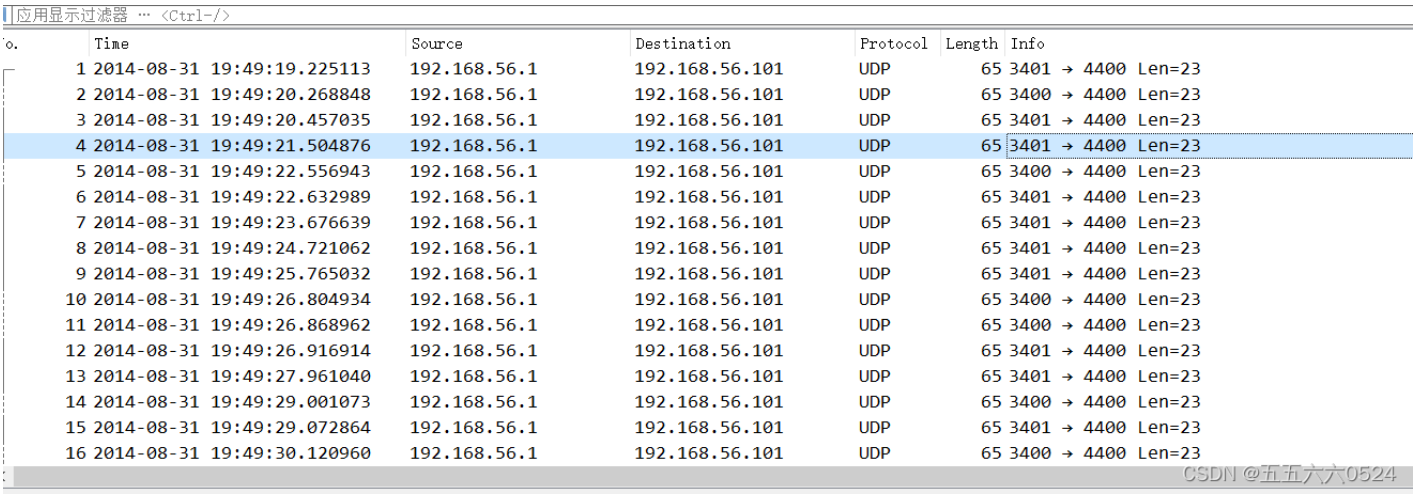
1、暴力破解：尝试选择范围内所有的字符组合
例如：选择范围：数字 长度：1-6
从1开始跑到999999
2、掩码：已知道密码某个位置的字符
掩码默认为：？
例如：掩码为：www.?????.com 范围选小写a-z
从www.aaaaa.com 跑到www.zzzzz.com
3、字典：在字典中寻找密码
密码必须在字典内
最后，破解成功会提示密码

35、Hidden-Message

攻防世界 Misc高手进阶区 4分题 Hidden-Message_闵行小鱼塘-CSDN博客

我是怎么也没有想到会是端口隐写，追踪流udp.stream eq 0和udp.stream eq 1加在一起是量子物理学的概念之类的东西，UDP协议传输数据包

端口的最后一个数字提取出来



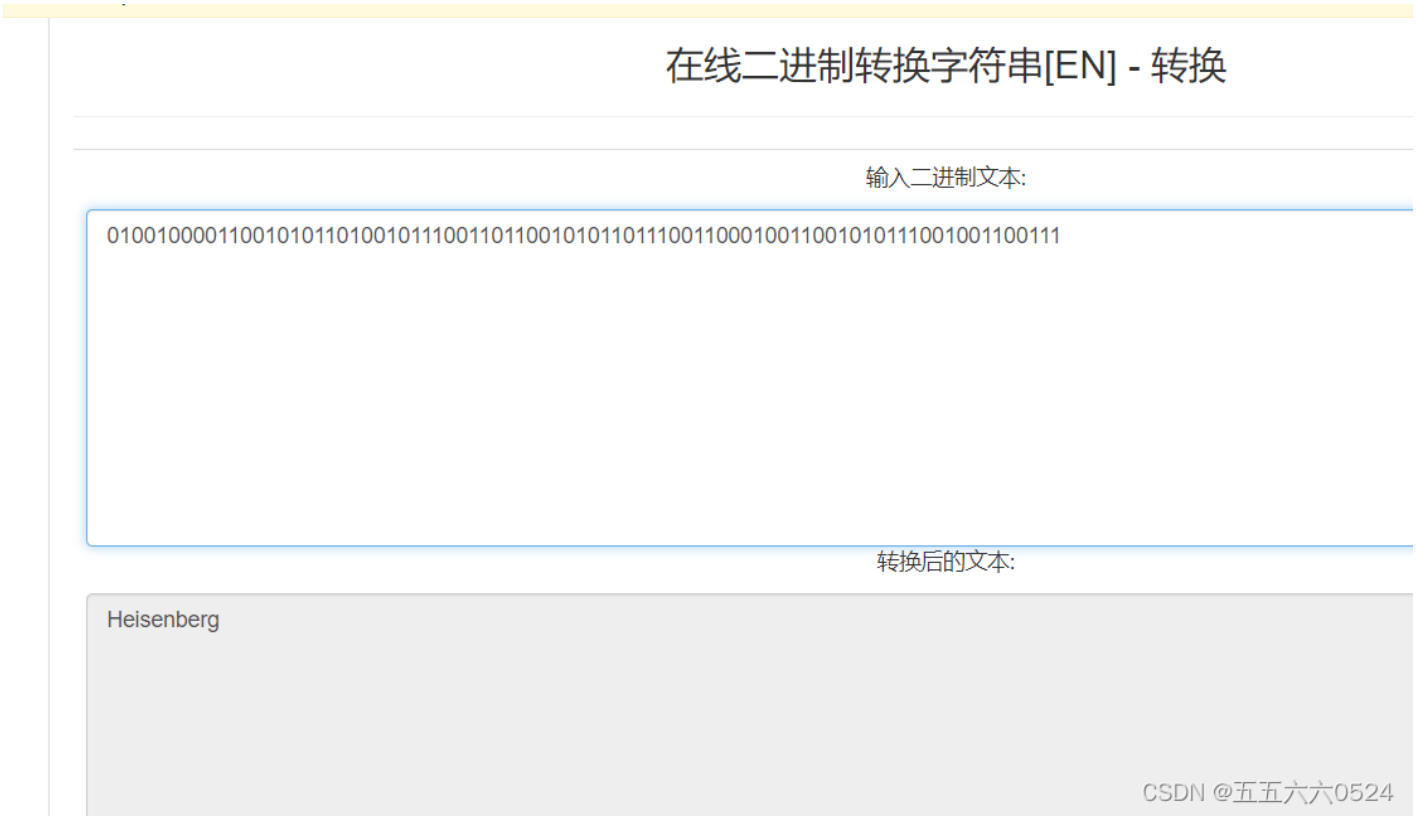
应用显示过滤器 … <Ctrl-/>

o.	Time	Source	Destination	Protocol	Length	Info
1	2014-08-31 19:49:19.225113	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
2	2014-08-31 19:49:20.268848	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23
3	2014-08-31 19:49:20.457035	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
4	2014-08-31 19:49:21.504876	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
5	2014-08-31 19:49:22.556943	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23
6	2014-08-31 19:49:22.632989	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
7	2014-08-31 19:49:23.676639	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
8	2014-08-31 19:49:24.721062	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
9	2014-08-31 19:49:25.765032	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
10	2014-08-31 19:49:26.804934	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23
11	2014-08-31 19:49:26.868962	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23
12	2014-08-31 19:49:26.916914	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
13	2014-08-31 19:49:27.961040	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
14	2014-08-31 19:49:29.001073	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23
15	2014-08-31 19:49:29.072864	192.168.56.1	192.168.56.101	UDP	65	3401 → 4400 Len=23
16	2014-08-31 19:49:30.120960	192.168.56.1	192.168.56.101	UDP	65	3400 → 4400 Len=23

1和0之间互转，再转换成字符串，flag是Heisenberg

```
import binascii
a='10110111100110101001011010001100100110101001000110011101100110101000110110011000'
b=''
for i in a:
    if i=='1':
        b+='0'
    elif i=='0':
        b+='1'
print(b)
```

> 01001000011001010110100101110011011001010110111001100010011001010111001001100111

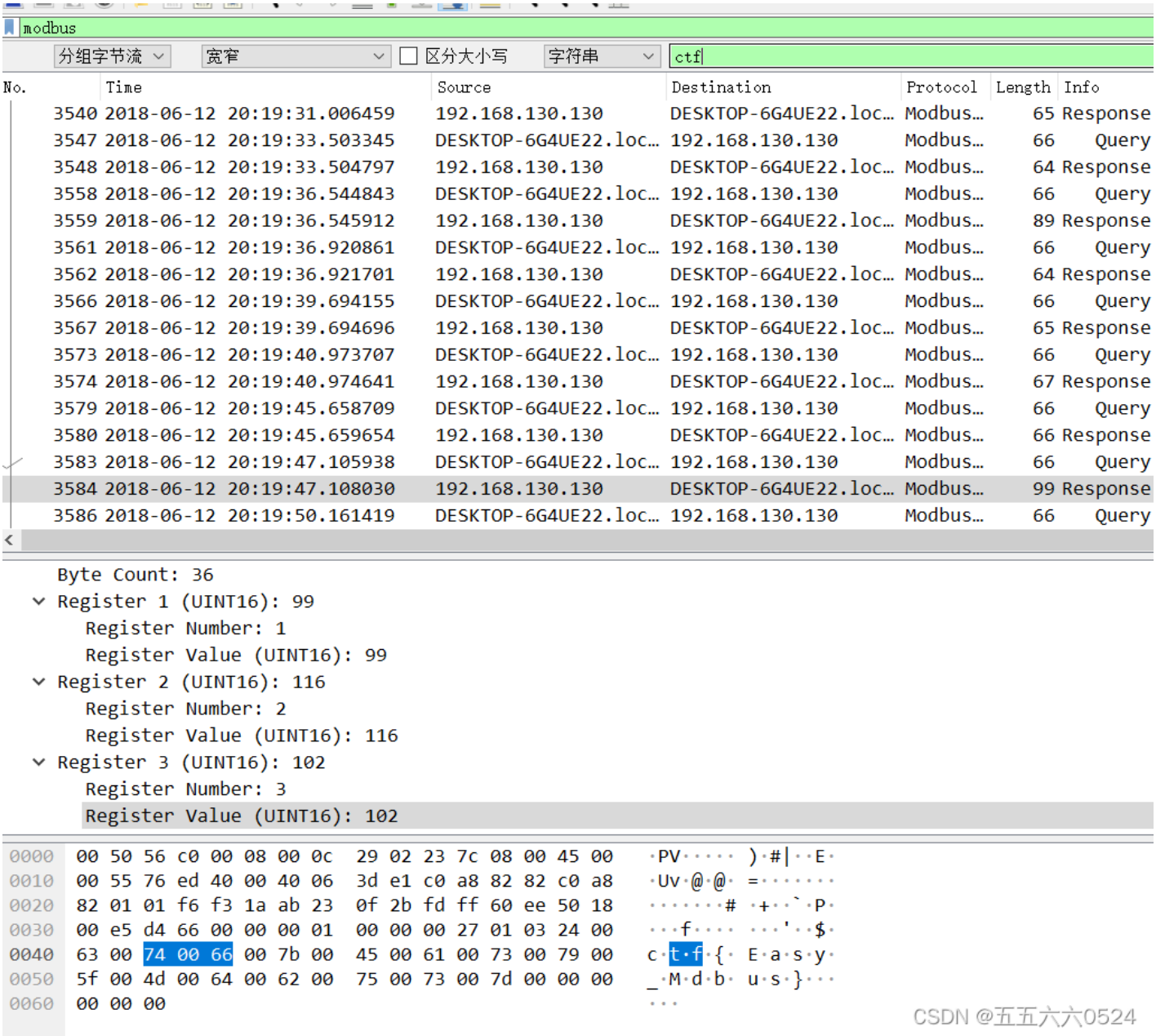


36、embarrass

wireshark里没有分析出来什么，拖到linux里出了，flag{Good_b0y_W3ll_Done}

37、神奇的Modbus

工业设备消息传输使用modbus协议，过滤一下直接搜，但是它ctf{Easy_Mdbus}，中间加个o，ctf{Easy_Modbus}就对了



38、MISCall

在kali里file一下文件，是bzip2格式，改后缀为bz2，直接解压，里面有flag.txt和.git目录，或者用“tar -xvjf +文件名”这个命令，会直接显示git目录下的文件同时生成一个ctf文件夹，里面只有flag.txt，flag.txt里面什么也没有

> *.bz2 // bzip2程序压缩产生的文件Linux下文件的打包、解压缩指令—tar，gzip，bzip2，unzip，rar - yhjoker - 博客园本文是笔者对鸟叔的Linux私房菜(基础学习篇) 第三版(中文网站)中关于 Linux 环境下打包和解压缩指令的内容以及日常操作过程中所接触的相关指令的总结和记录，以供备忘和分享。更多详细信息可直接参https://www.cnblogs.com/yhjoker/p/7568680.html#tar

> .git文件夹是git init后在当前目录生成的一个管理git仓库的文件夹，

> git log命令，会输出所有的日志

```
┌──(kali㉿kali)-[~/桌面/ctf]
└─$ git log
commit bea99b953bef6cc2f98ab59b10822bc42afe5abc (HEAD → master)
Author: Linus Torvalds <torvalds@klaava.Helsinki.Fi>
Date:   Thu Jul 24 21:16:59 2014 +0200

    Initial commit
```


git list查看修改列表，有东西

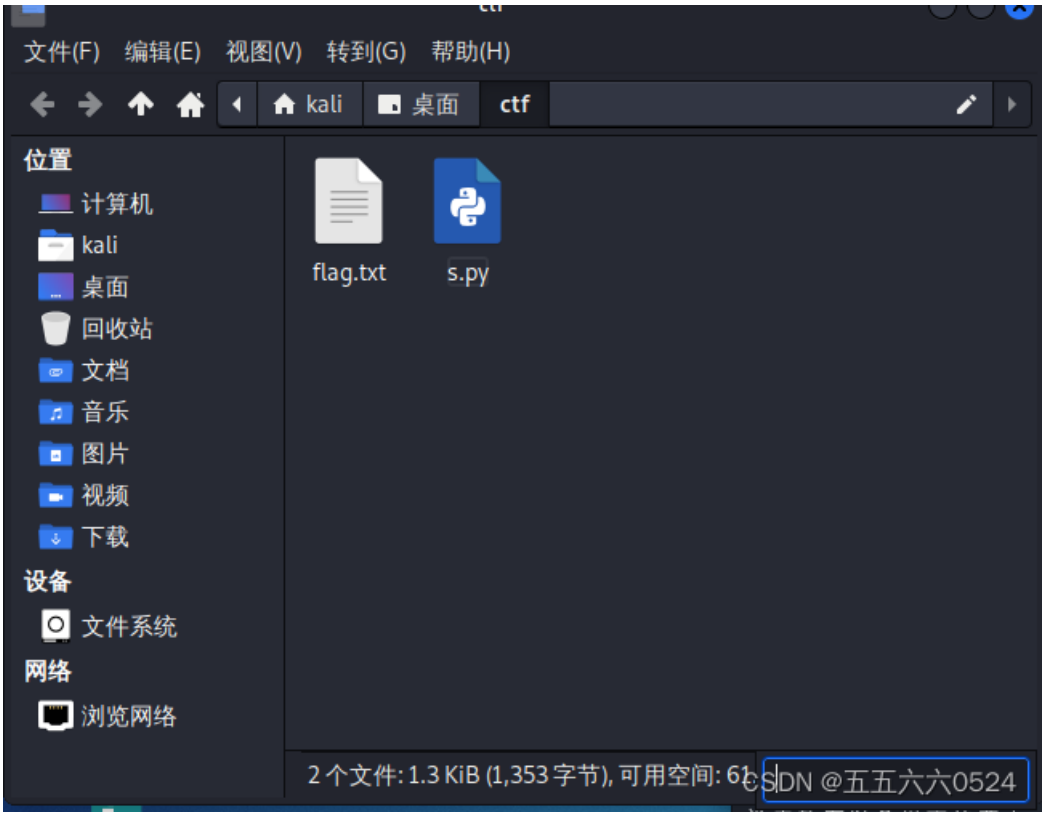
```
┌──(kali㉿kali)-[~/桌面/ctf]
└─$ git stash list
stash@{0}: WIP on master: bea99b9 Initial commit
```


git stash show显示做了哪些改动，默认show第一个存储,如果要显示其他存贮，后面加stash@{$num}，比如第二个 git stash show stash@{1}

```
┌──(kali㉿kali)-[~/桌面/ctf]
└─$ git stash show
 flag.txt | 25 +++++++++++++++++++++++++-
 s.py     |  4 ++++
 2 files changed, 28 insertions(+), 1 deletion(-)
```


git stash apply应用某个存储,但不会把存储从存储列表中删除，默认使用第一个存储,即stash@{0}，如果要使用其他个，git stash apply stash@{$num}， 比如第二个：git stash apply stash@{1}，相当于恢复改变的内容

```
┌──(kali㉿kali)-[~/桌面/ctf]
└─$ git stash apply
位于分支 master
要提交的变更：
  （使用 "git restore --staged <文件>..." 以取消暂存）
        新文件：    s.py

尚未暂存以备提交的变更：
  （使用 "git add <文件>..." 更新要提交的内容）
  （使用 "git restore <文件>..." 丢弃工作区的改动）
        修改：      flag.txt
```

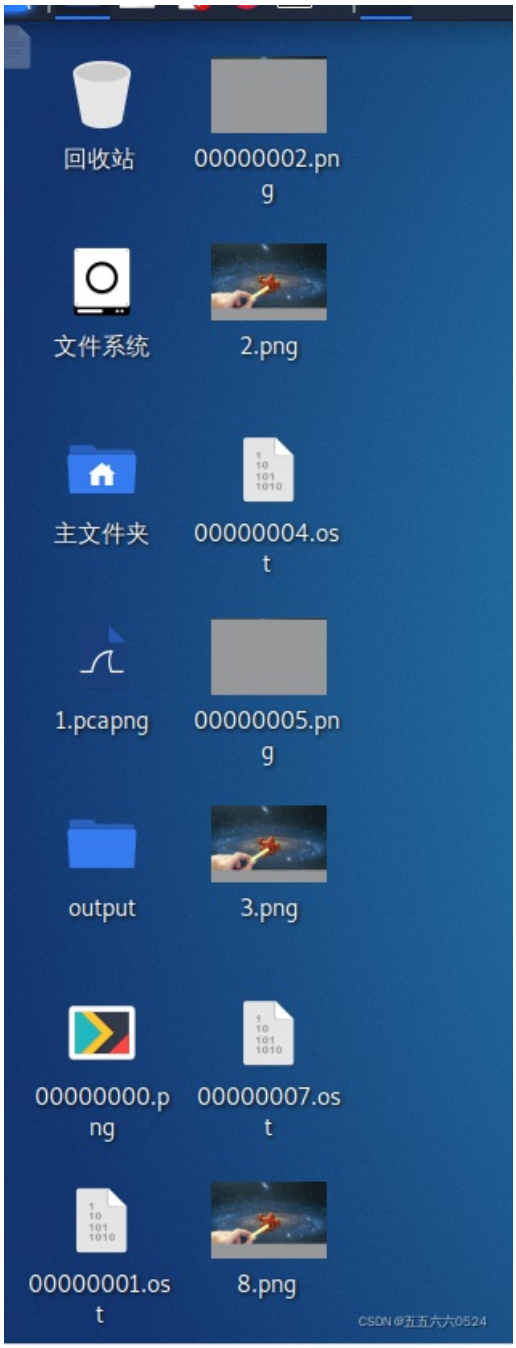


运行s.py，改一下路径，直接出NCN4dd992213ae6b76f27d7340f0dde1222888df4d3

39、flag_universe

流量包里搜索universe，能找到几个universe.png，应该是从流量包里提取图片，binwalk一下发现确实有图片，但是foremost提取不出来，网上教的用wireshark提取图片实在没看懂，参考了这位的教程流量取证-流量中提取文件 - micr067 - 博客园以前整理的一些东西，拿出来做备忘 PCAP 报文就是抓取实际在网络中传输的图片，视频等数据，然后以PCAP 格式存储形成的文件。工作中对离线的数据包进行回溯分析，有时会遇到将 PCAP 中的码流还原成https://www.cnblogs.com/micr067/p/14076573.html

> 流量包中提取文件：1、tcpxtract -f +文件名 2、用NetworkMiner 3、用wireshark 4、foremost -v -i +文件名 5、用Chaosreader



1成功了，235不会用，4没成功，1之后出来一大堆图，

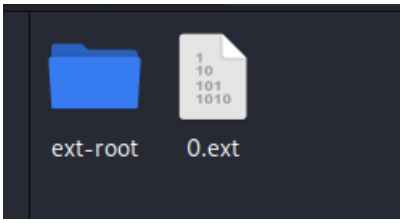
改了名，zsteg试到8.png的时候出了flag，flag{Plate_err_klaus_Mail_Life}

40、Get-the-key.txt

拖进kali里查看文件类型

```
└─$ file forensic100
forensic100: Linux rev 1.0 ext2 filesystem data, UUID=0b92a753-7ec9-4b2b-8cdb-79c1fa1502a9
```


binwalk -e forensic100分离一下得到一大堆gzip文件和一个0.ext



在文件夹里查一下匹配到1，给1加后缀.zip，然后解压直接得到flag， SECCON{@]NL7n+-s75FrET]vU=7Z}

```
┌──(kali㉿kali)-[~/桌面/_forensic100.extracted/ext-root]
└─$ grep -r key.txt
grep: 1：匹配到二进制文件
```

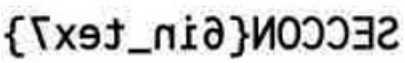

41、Reverse-it

拖进winhex里观察，开头是FFD9，结尾是FFD8，全部的十六进制反过来了

```
9D FF 70 0D B6 DA FC 93
B4 25 D4 7B 8C 00 A1 AB
7B 5A D6 84 BB D4 B9 7F

A2 00 D4 D4 00 00 66 96  87 54 2D 00 1E FF 00 00  ¢
84 00 84 00 10 10 10 00  64 94 64 A4 01 00 0E FF  ▌
8D FF                                             ÿ
```


```
a='9DFF700DB6DAFC937263282222BDD218B425D47B8C00A1AB609EE707BB0431567B5AD684BBD4B97E00A

b=a[::-1]

print(b)
```

复制新得到的数据，winhex剪切板数据—粘贴至新文件—ASCII HEX，然后保存成图片格式就行了，SECCON{6in_tex7}



42、打野

binwalk，winhex都没有发现什么，zsteg隐写发现了，qwxf{you_say_chick_beautiful?}

```
┌──(kali㉿kali)-[~/桌面]
└─$ zsteg 1.bmp
[?] 2 bytes of extra data after image end (IEND), offset = 0x269b0e
extradata:0         .. ["\x00" repeated 2 times]
imagedata           .. text: ["\r" repeated 18 times]
b1,lsb,bY           .. <wbStego size=120, ext="\x00\x8E\xEE", data="\x1Ef\xDE\x9E\xF6\xAE\xFA\xCE\x86\x9E" ... , even=fa
lse>
b1,msb,bY           .. text: "qwxf{you_say_chick_beautiful?}"
b2,msb,bY           .. text: "i2,C86k0."
b2,r,lsb,xY         .. text: "UUUUUU9VUUUUUUUUUUUUUUUUUUUUU"
b2,g,msb,xY         .. text: ["U" repeated 22 times]
b2,b,lsb,xY         .. text: ["U" repeated 10 times]
b3,g,msb,xY         .. text: "V9XDR\\d@"
b4,r,lsb,xY         .. file: TIM image, Pixel at (4353,4112) Size=12850×8754
b4,g,lsb,xY         .. text: "3\"\"\"\"\"3###33##3#UDUEEEEEDDUETEDEDDUEEDTEEEUT#!"
b4,g,msb,xY         .. text: "\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"\"DDDDDDDDDDD\"\"\"\"DDDDDDDDDDD*LD"
b4,b,lsb,xY         .. text: "gfffffvwgwfgwwfw"
```


43、3-11

binwalk发现有Zlib

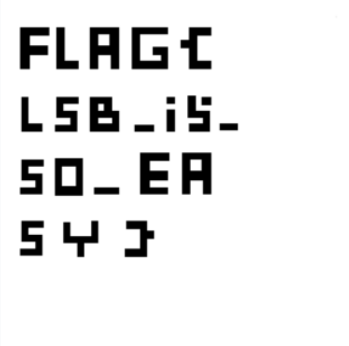
```
└─$ binwalk 2.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1440 x 1080, 8-bit/color RGB, non-interlaced
41            0x29            Zlib compressed data, default compression
```


zsteg隐写发现有压缩包

```
┌──(kali㉿kali)-[~/桌面]
└─$ zsteg 2.png
imagedata           .. file: Apple DiskCopy 4.2 image \352, 4278190592 bytes, 0xfffcfc tag size, MFM CAV dsdd (720k),
0x2 format
b1,rgb,lsb,xy       .. file: Zip archive data, at least v2.0 to extract
b2,g,lsb,xy         .. file: MIPSEB-LE MIPS-III ECOFF executable not stripped - version 196.207
b2,rgb,msb,xy       .. text: "U@E@A|yh"
b3,r,lsb,xy         .. text: "eM*I&'*z-6`"
b3,g,msb,xy         .. text: "ChvVe!'c~"
b3,rgb,msb,xy       .. text: "2mI$AA&`"
b3,bgr,lsb,xy       .. file: PGP Secret Key -
b4,r,lsb,xy         .. text: "eWvwuwfgTn"
b4,r,msb,xy         .. text: "fUU3U5U]eb"
b4,g,lsb,xy         .. text: "gggwfvwf"
b4,b,lsb,xy         .. text: "&t235SV^"
b4,rgb,msb,xy       .. text: "375SS5375SS=U35m"
b4,bgr,msb,xy       .. text: "373US3573US5=S="
```

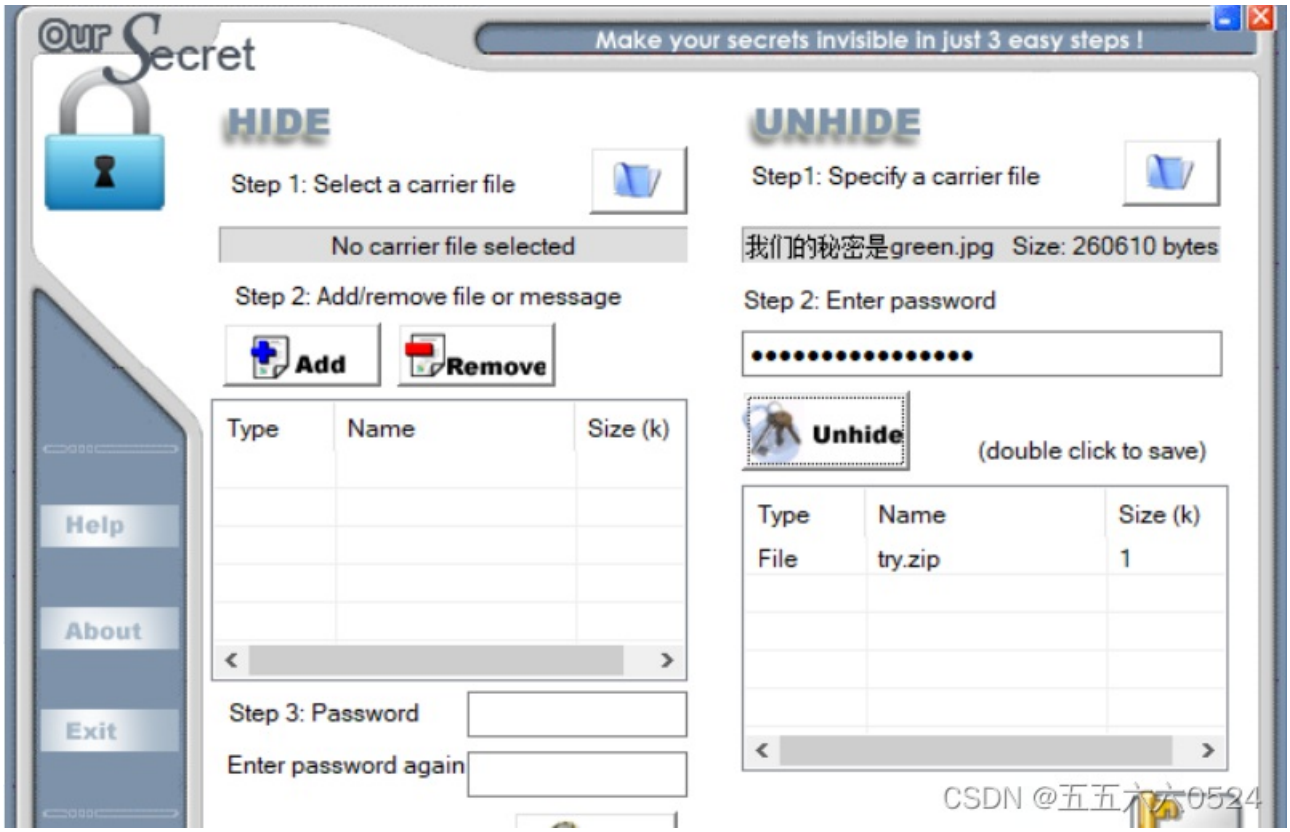

zsteg -e b1,rgb,lsb,xy 2.png -> 1.zip ，提取出来，解压，得到一串base64码，转图片，FLAG{LSB_i5_SO_EASY}（一直以为那个是R来着）

iVBORw0KGgoAAAANSUhEUgAAAPoAAAD6CAYAAACI7Fo9AAAAAXNSR0IArs4c6QAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEnQAABJ0Ad5mH3gA
AAVqSURBVHhe7d1bTuRGAEDRIftfK1tIxgNRJNINbpff9xzJmvlpP6q5KpeB5u3v334Bt/bX57/AjQkdAoQOAUKHAKFDgNAhQOgQIHQIEDoECB0ChA4BQoc
Le39///zf9/xSCwSY0SFA6BAgdAgQOgQIHQKEDgGLv7329vb2+b/1PDqVZ8dZ67uCe13HV1sc95GRcRo9x7XeI8aZ0SFA6BAgdAgQOgR4GLdw/99d/5xz2/
q61jD6Hp/pWuo2CX3NN3jrIJbuf/T6t76uNVzhHJnHrTsECB0ChM6fW/RHG/chdAgQOgQIHQJO8330Z6fx7DhrfYtni7XonHPb+rpe8eoYHHGOjDGjX9AU5t
wNJkLnzwz9aOM+hA4BQocAoUOA0CHAb68N7P+o1z7zaJ9r72+y1tizHzM6BAgdAoQOAZddo8/107mMrEO3eO3aRt6Lrcee/ZjRIUDoECB0CBA6BAgdAh
Y/dQeuw4wOAUKHAKFDgNAhQOgQIHQIEDoECB0ChA4BQocAoUOA0CFA6BAgdAgQOgQIHQKEDgFChwAfJcVDI3+cgvMZCv3RF8PaXwhb/7WQO/81k
pFY5772u/E789jUuHWHAKFDgNAhQOgQcMmHcWseY/Sh09rXexaj4zK569hckRmdIVPMzzbOQ+gQIHQIEPqNTevnRxs9t/7JuDnnssUX/lnWp8+ubY9xOc
sY8MGMDgFChwChQ4DQIeD0D+PmWnoulw+sJmcag69Gr22uM48BH8zoECB0CBA6BKy+Rp/rLOvEkWt45izr09E1+lbv75L9nmVMr+rUoY9YnJfncxZ
cef+Ta5p7fVu/vkv2+Mqb8n1t3CBA6BAgdAoTO7qb19qsbY4YexgHXYEaHAKFDgNAhQOgQIHQIEDoECB0ChA4BQocAoUOA0CFA6BAgdAgQOgT4NdW
TOeoz7I46Lvswo0OA0CFA6BBw6jX62uvGK6wZrdHZwmF/wGHy06GFvo451/3suCPv0RXGu0LoJ3NUOEK/N2t0CBA6BAgdAlZfo6+5LhvZ/9bntpWj1rzPjvvT
MY86X15jRocAoUOAW/eT+e5WelmR8Rp1hfGuMKNDwOqhTzPDnA3YjxkdAoQOAUKHAKFDwC6/j/7o4ducwy593WTktUf67kHlluf/7LhL36Fca7wowOA
UKHAKFDwOVCn9aDczbgP4sfxo3GNPKQZ6mfjrnW8UYeQu19zf96dtyR92lkHFiXW3cIEDoECB0ChA4Bu/xk3FJHPZiCuzl16MA63LpDgNAhQOgQIHQIEDo
ECB0ChA4BQocAoUOA0CFA6BAgdAgQOgQIHQKEDgFChwChQ8DQJ8yMftTTWT7c5tF1rH1ul2PlQ4AYdckZfYrm6wY859YdAoQOAUKHgNUfxu3x4Gjt4+

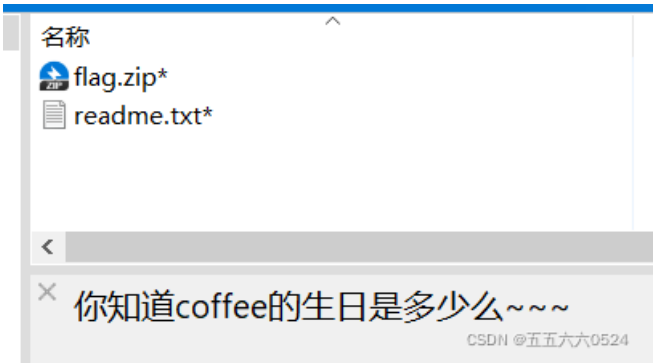


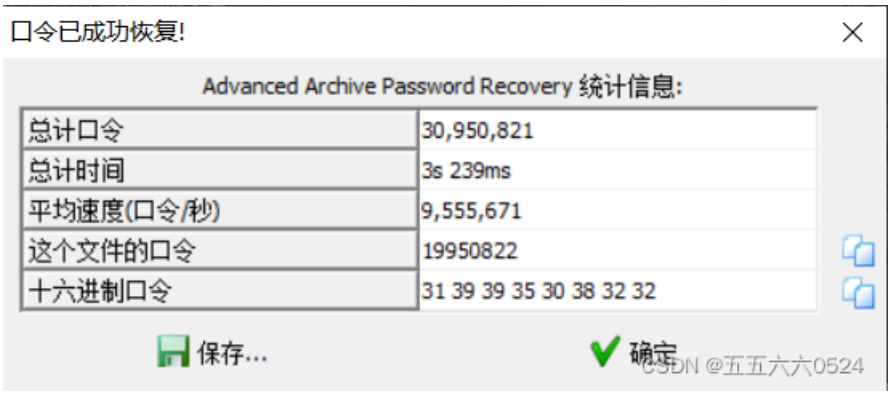
44、我们的秘密是绿色的

要用到our secret这个软件，密码就是那张图上绿色的数字0405111218192526

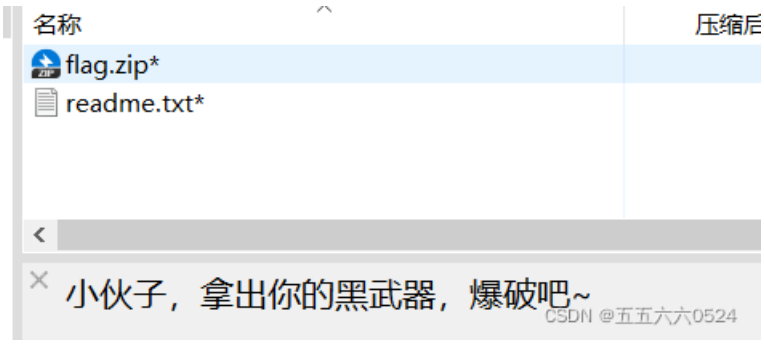


得到try.zip，有提示，爆破选纯数字，年月日共8位

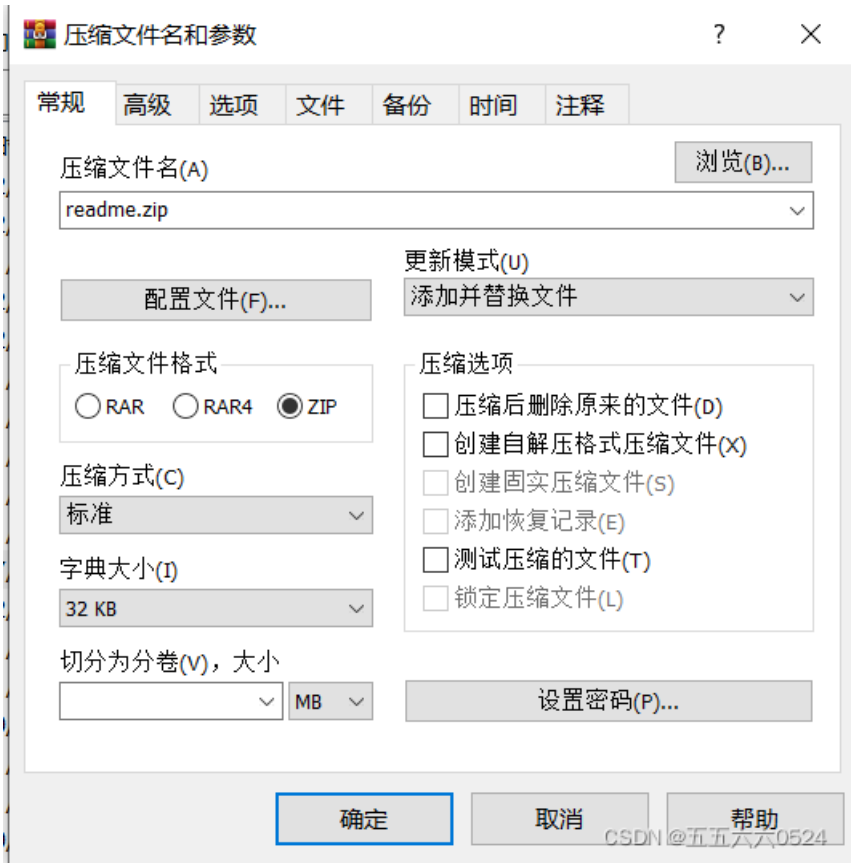




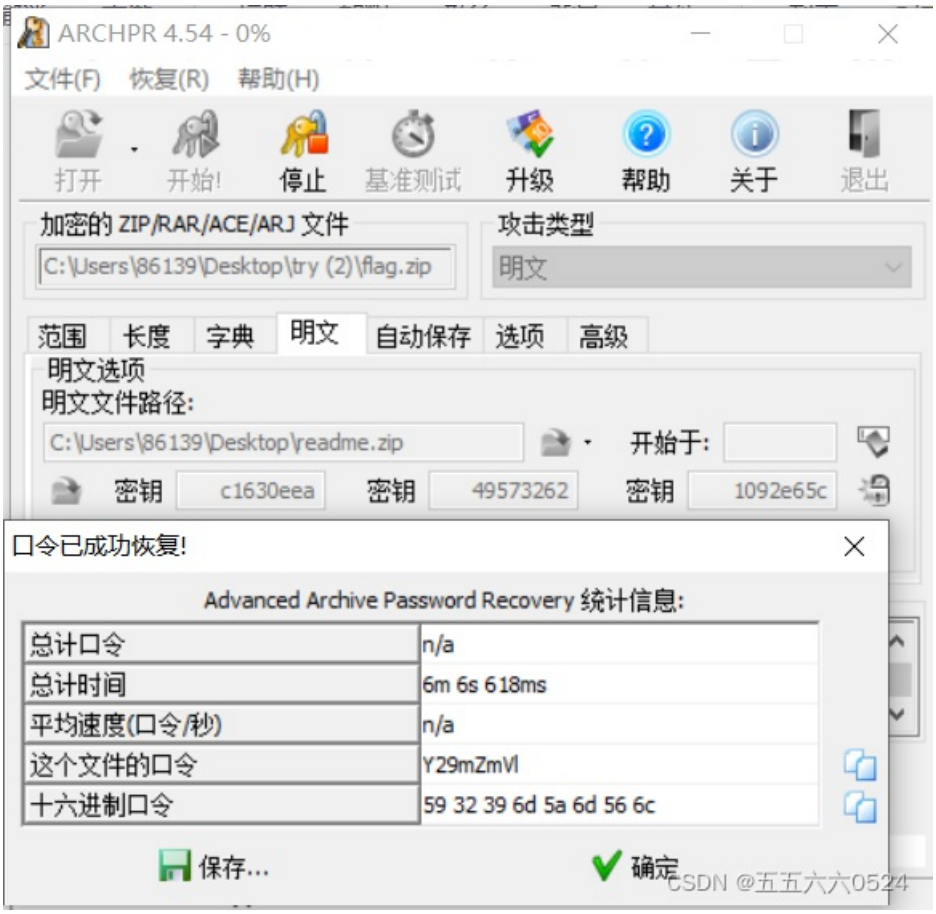
密码就是19950822，还需爆破， 这个应该就是明文攻击了



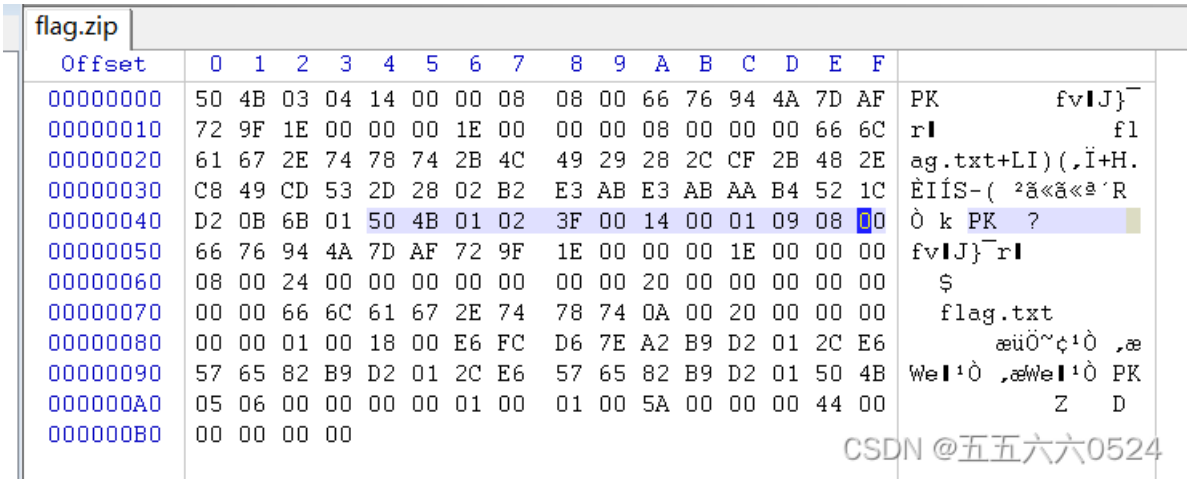
把readme.txt压缩了（不要用bandzip，否则会显示“在选定的档案中没有匹配的文件”，要用winrar），先选中那个文件，然后点添加，直接确定就行了



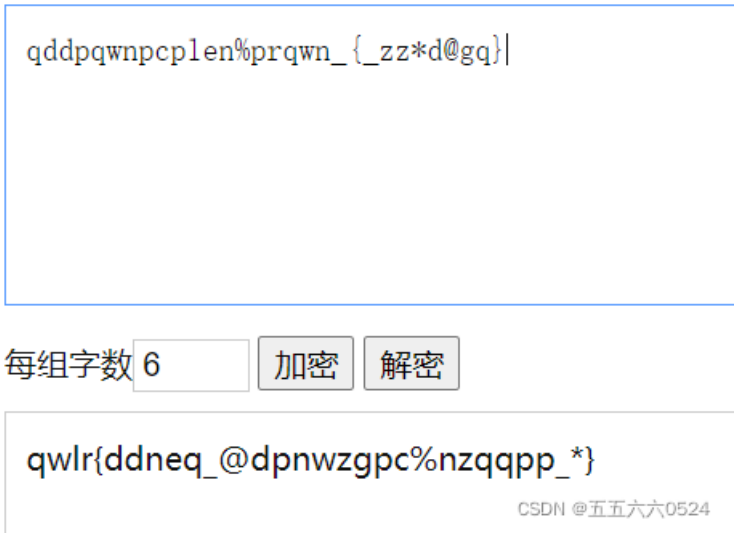
明文攻击，密码是Y29mZmVl



解压得到的zip居然还有密码，爆破不了，扔winhex里发现是伪加密， 01 改成00就行，不是把09改成08



得到qddpqwnpcplen%prqwn_{_zz*d@gq}，这玩意是栅栏加密，6的时候最像flag



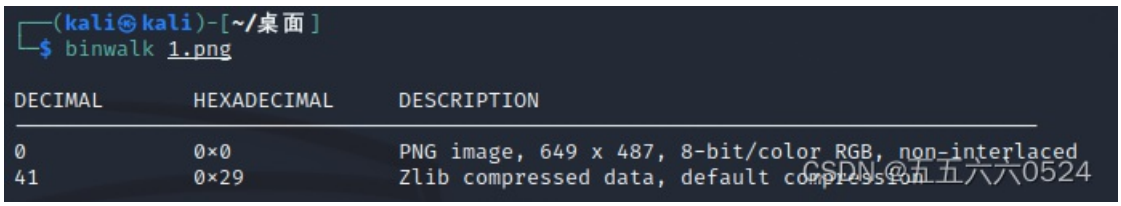
qwlr{ddneq_@dpnwzgpc%nzqqpp_*}，这玩意又是凯撒加密，偏移量是11，出flag{ssctf_@seclover%coffee_*}，这一题一套接一套的，真真麻烦

45、小小的PDF

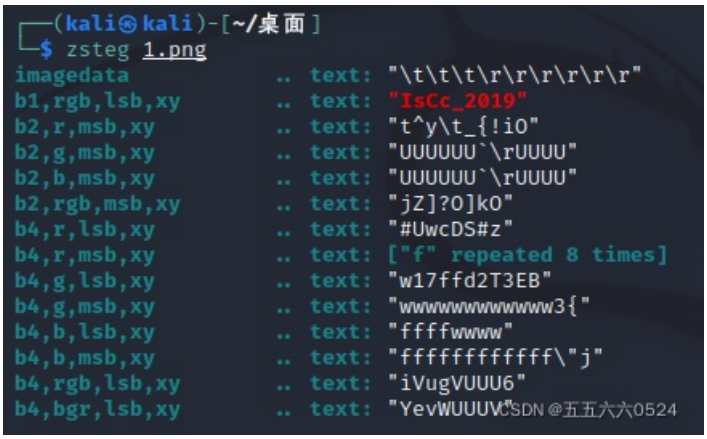
pdf打开是一张图，拖进kali里binwalk一下，发现有东西，binwalk -e提取不出来，foremost提取出来有图片， flag直接出SYC{so_so_so_easy}

46、倒立屋

把图拖进kali里binwalk一下，zsteg隐写



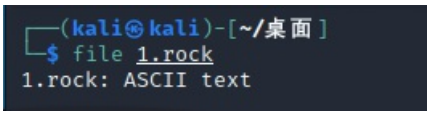
发现标红lsCc_2019



直接提交不对，根据题目反转一下，flag{9102_cCsl}（最后那个是大写的i，不是小写的L）

47、Become_a_Rockstar

拖进kali里查询文件类型，是文本，把rock改成txt ，得到一大堆

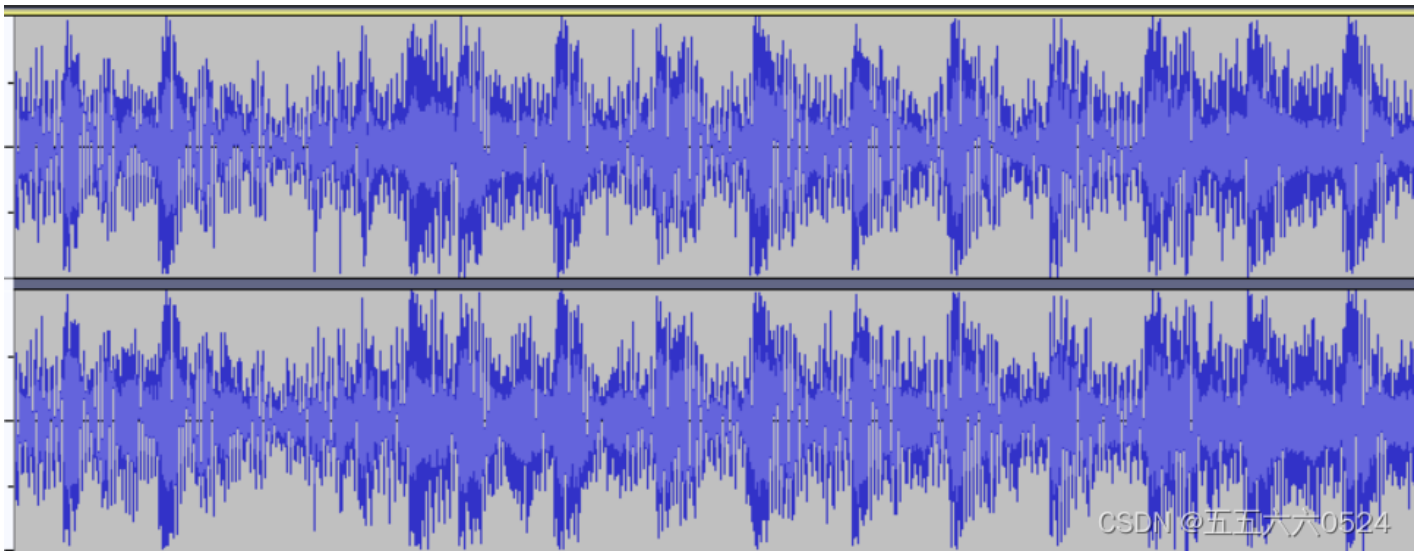


rockstar是一种语言，找到的脚本，flag就是NCTF{youarnicerockstar} WriteUp-攻防世界-MISC Become_a_Rockstar_一块萌肝的博客-CSDN博客

```
Leonard_Adleman = "star"
Problem_Makers = 76
Problem_Makers = "NCTF{"
def God(World):
    a_boy = "flag"
    the_boy = 3
def Evil(your_mind):
    a_girl = "no flag"
    the_girl = 5
Truths = 3694
Bob = "ar"
Adi_Shamir = "rock"
def Love(Alice, Bob):
    Mallory = 13
    Mallory = 24
Everything = 114514
Alice = "you"
def Reality(God, Evil):
    God = 26
    Evil = 235
Ron_Rivest = "nice"
def You_Want_To(Alice, Love, Anything):
    You = 5.75428
your_heart = input()
You = 5
your_mind = input()
Nothing = 31
if Truths * Nothing == Everything:
    Rsa = Ron_Rivest + Adi_Shamir + Leonard_Adleman
if Everything / Nothing == Truths:
    Problem_Makers = Problem_Makers + Alice + Bob
print(Problem_Makers)
the_flag = 245
the_confusion = 244
print(Rsa)
Mysterious_One = "}"
print(Mysterious_One)
This = 4
This = 35
This = 7
This = 3
This = 3
This = 37
```

48、intoU

用Audacity打开，没有发现什么

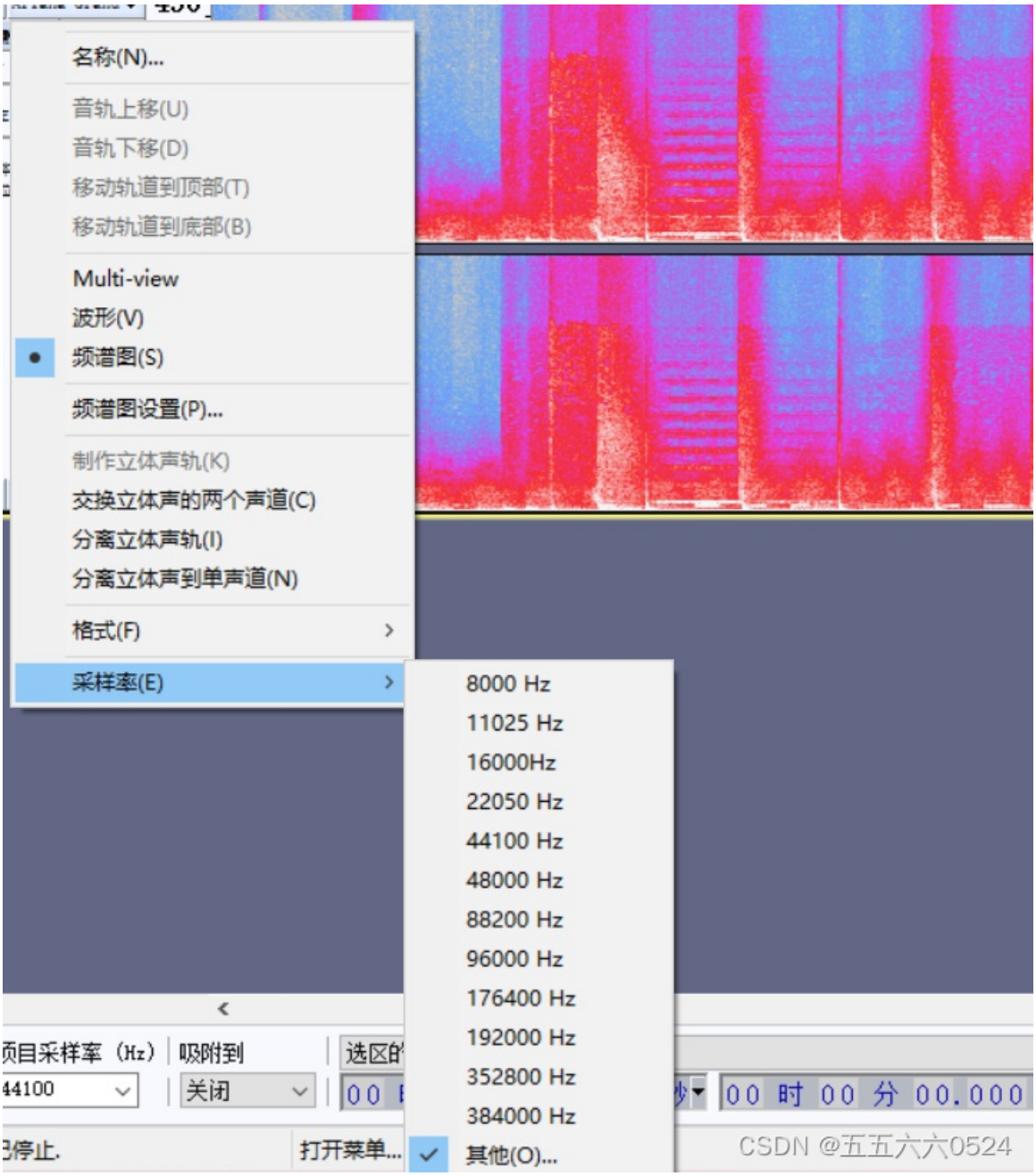


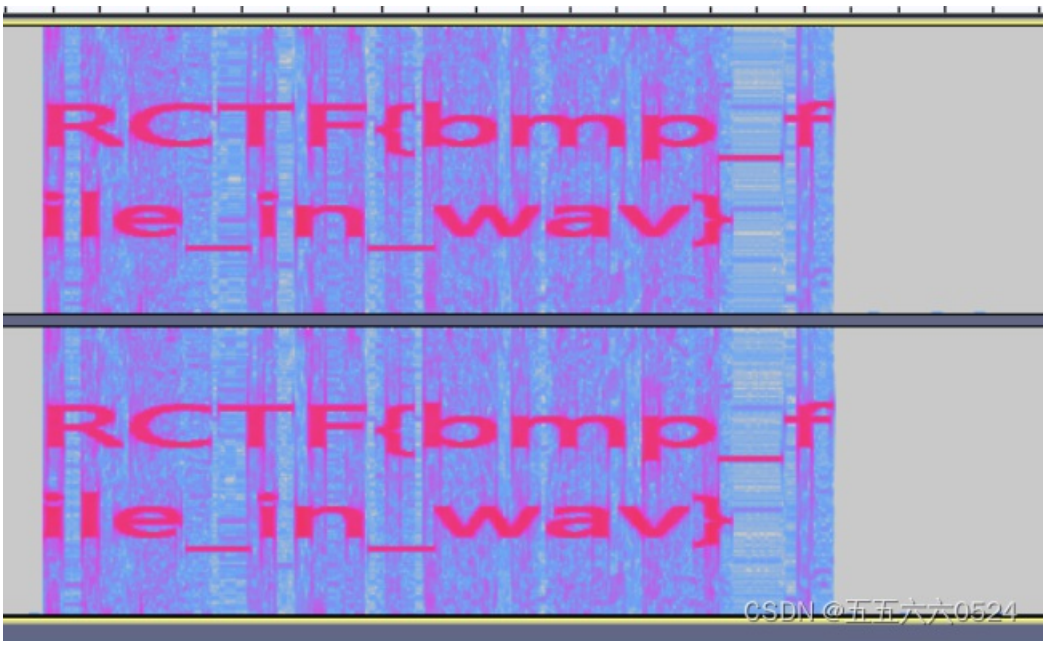
binwalk发现有东西但提取不出来

DECIMAL	HEXADECIMAL	DESCRIPTION
1406134	0×1574B6	MySQL MISAM index file Version 5
2411758	0×24CCEE	LZ4 compressed data, legacy
4459729	0×440CD1	MySQL ISAM index file Version 1
6503687	0×633D07	MySQL MISAM compressed data file Version 6
7094494	0×6C40DE	MySQL ISAM compressed data file Version 5
7210389	0×6E0595	MySQL ISAM compressed data file Version 2
7509633	0×729681	MySQL MISAM index file Version 3
7590019	0×73D083	MySQL MISAM index file Version 7
9374200	0×8F09F8	MySQL ISAM compressed data file Version 4
10011194	0×98C23A	MySQL MISAM index file Version 1
10084330	0×99DFEA	MySQL MISAM compressed data file Version 7
10738445	0×A3DB0D	MySQL MISAM index file Version 5
11269025	0×ABF3A1	MySQL MISAM compressed data file Version 4
12322425	0×BC0679	MySQL MISAM index file Version 3

CTF 隐藏的信息 intoU base64÷4 ..._艺博东的博客-CSDN博客

改成频谱图，并把采样率改成900，拉到最右面，出RCTF{bmp_file_in_wav}





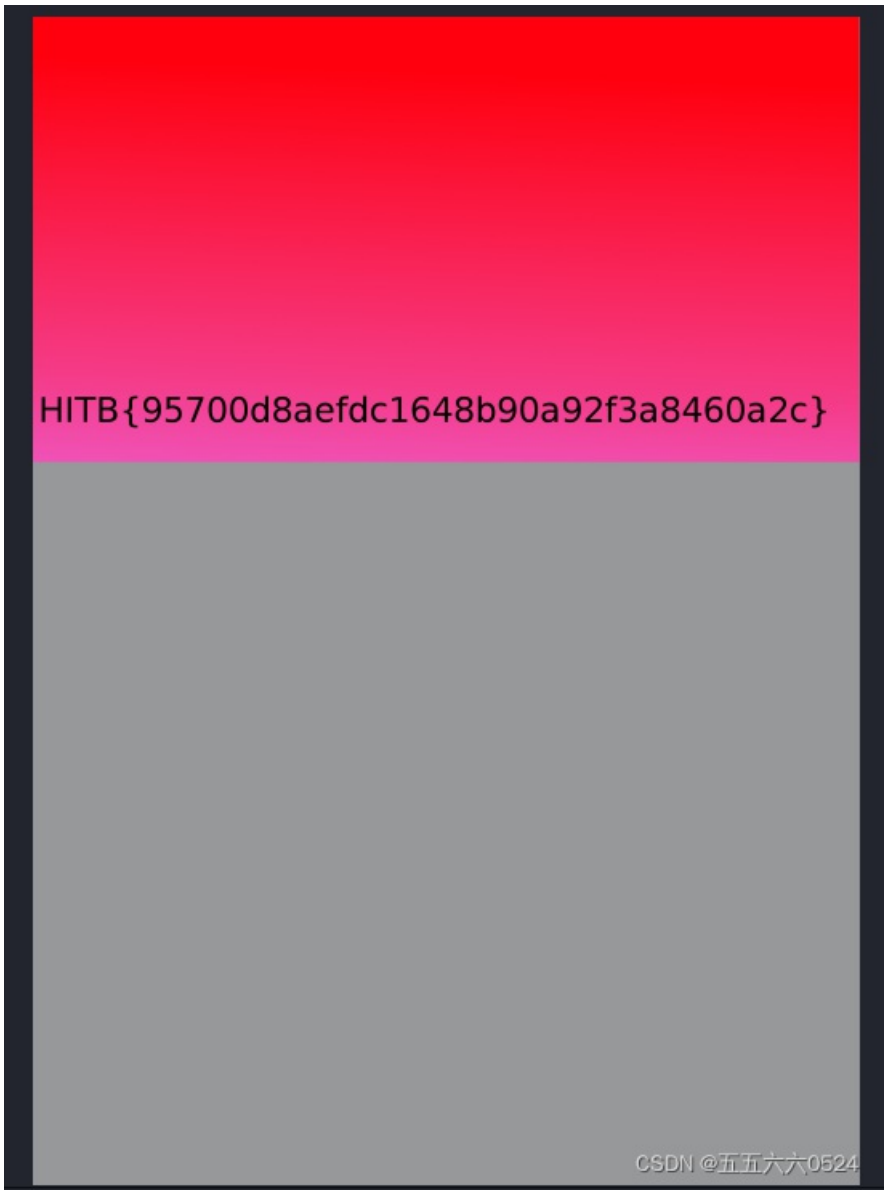
49、Cephalopod

把流量包扔进wireshark，搜flag能出来，应该是里面有图片

```
    Inode: 0
  Path, Inode: 0x0000000000000001, Rel: "flag.png"
    Encoding Version: 0x01
    Inode: 0x0000000000000001
  Relative component: flag.png
      Size: 8
      Data: flag.png
```

```
0d0  01 01 00 00 00 00 00 00  00 08 00 00 00 66 6c 61   ........ .....fla
0e0  67 2e 70 6e 67 01 00 00  00 00 00 00 00 00 00 00   g.png... ........
0f0  00 00 01 00 00 00 00 00  00 00 03 00 00 00 00 00   ........ ........
```


拖进kali里binwalk一下，果然，binwalk -e、foremost提取不出来，用tcpxtract -f +文件名出来两张图，图片上直接就是flag，HITB{95700d8aefdc1648b90a92f3a8460a2c}



50、Excaliflag

stegslove在蓝色通道里找到了，3DS{Gr4b_Only_th1s_B1ts}