




# 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

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/wow0524/article/details/122432361>

版权



[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
- 2、base64stego
- 3、功夫再高也怕菜刀
- 4、easycap
- 5、reverseMe
- 6、Hear-with-your-Eyes
- 7、What-is-this
- 8、normal-png
- 9、something in image
- 10、wireshark-1
- 11、pure\_color
- 12、Aesop\_secret
- 13、a\_good\_idea
- 14、simple\_transfer
- 15、Training-Stegano-1
- 16、2017\_Dating\_in\_Singapore
- 17、can\_has\_stdio?
- 18、János-the-Ripper
- 19、Erik-Baleog-and-Olaf
- 20、Test-flag-please-ignore
- 21、hit-the-core

- 22、快乐游戏题
- 23、glance-50
- 24、misc\_pic\_again
- 25、Banmabanma
- 26、stage1
- 28、red\_green
- 29、Recover-Deleted-File
- 30、适合作为桌面
- 31、就在其中
- 32、base64+4
- 33、很普通的数独
- 34、再见李华
- 35、Hidden-Message
- 36、embarrass
- 37、神奇的Modbus
- 38、MISCall
- 39、flag\_universe
- 40、Get-the-key.txt
- 41、Reverse-it
- 42、打野
- 43、3-11
- 44、我们的秘密是绿色的
- 45、小小的PDF
- 46、倒立屋
- 47、Become\_a\_Rockstar
- 48、intoU
- 49、Cephalopod
- 50、Excaliflag

---

## 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{sajbcibzskjicnbhsbvcjbjszcszbkzj}

Hex editor view showing memory dump. The file path is `flag.txt`. The sidebar shows file properties for `[f1fc23f5c743425d9e007388...]` in the `Ext3` file system.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0011EC00	F7	00	00	00	0C	00	01	02	2E	00	00	00	02	00	00	00
0011EC10	24	00	02	02	2E	2E	00	00	3A	02	00	00	18	00	00	01
0011EC20	2E	86	6C	61	67	2E	74	78	74	2E	73	77	70	00	00	00
0011EC30	3B	02	00	00	D0	03	08	01	66	6C	61	67	2E	74	78	74
0011EC40	74	2E	73	77	78	00	00	00	00	00	00	00	00	00	00	00
0011EC50	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011EC60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011EC70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011EC80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011EC90	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECA0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECB0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECC0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECD0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECE0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ECF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ED00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ED10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ED20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ED30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0011ED40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

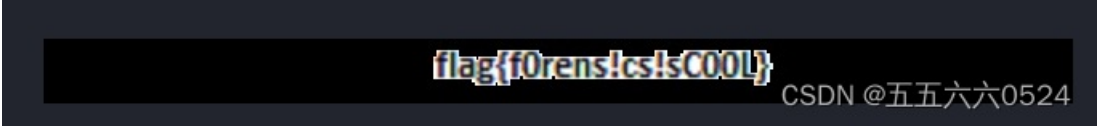
File explorer view showing the directory structure. The file `flag.txt` is highlighted.

名称	扩展名	大小	创建时间	修改时间	i-节点已修改	属性	第一扇区
.. = (根目录)		3.0 KB		2015/10/30 ...	2015/10/30 ...	rw-r--r--	584
. = O7avZhikgKgbF		1.0 KB		2018/08/08 ...	2018/08/08 ...	rw-r--r--	2,294
.flag.txt.swp	swp	0 B		2018/08/08 ...	2018/08/08 ...	rw-r--r--	
flag.txt	txt	53 B		2018/08/08 ...	2018/08/08 ...	rw-r--r--	13,322

Hex editor view showing memory dump. The ASCII text is `ZmxhZ3tzYWpiY2li`.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00681400	5A	6D	78	68	5A	33	74	7A	59	57	70	69	59	32	6C	69
00681410	65	6E	4E	72	61	6D	70	6A	62	6D	4A	6F	63	32	4A	32
00681420	59	32	70	69	61	6E	4E	36	59	33	4E	36	59	6D	74	36
00681430	61	6E	30	3D	0A	00	00	00	00	00	00	00	00	00	00	00
00681440	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00681450	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00681460	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00681470	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00681480	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00681490	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006814A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006814B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006814C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006814D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
006814E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

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

```
50 4B 01 02 3F 03 14 03 09 00 08 00 08 BF
FF 32 7D 4B F9 00 00 00 R5 1R 00 00 09 00
```

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

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity. The word steganography is of Greek origin and means "concealed writing" from the Greek words steganos meaning "covered or protected", and graphein meaning "to write". The first recorded use of the term was in 1499 by Johannes Trithemius in his Steganographia, a treatise on cryptography and steganography disguised as a book on magic. Generally, messages will appear to be something else: images, articles, shopping lists, or some other covertext and, classically, the hidden message may be in invisible ink between the visible lines of a private letter.

The advantage of steganography, over cryptography alone, is that messages do not attract attention to themselves. Plainly visible encrypted messages (no matter how unbreakable) arouse suspicion, and may in themselves be incriminating in countries where encryption is illegal. Therefore, whereas cryptography protects the contents of a message, steganography can be said to protect both messages and communicating parties.

加密 >

< 解密

```
U3RlZ2Fub2dyYXB0eSBpcyB0aGUgYXJ0IGFuZCBzY2llbmNlIG9m
IHdyaXRpbmcaGlkZGVuIG1lc3NhZ2VzIGludH00Zm90Zm90Zm90
yBvbmV=
LCBhcGFydCBmcm9tIH00Zm90Zm90Zm90Zm90Zm90Zm90Zm90
dCwgZm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
Y3RlZHRoZSBleGlzdGVuY2Ugb2YgdGhllIG1lc3M=
YWdlLCBhIGZvcml0b2Ygc2VjdXJpdHkgdGhyb3VnaCBvYnNjdXJpdHkuI
aG90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
bWVhbnMglmNvbmNlYW==
bGVkIHdyaXRpbmcaGlkZGVuIG1lc3NhZ2VzIGludH00Zm90Zm90
W5pbmcaGlkZGVuIG1lc3NhZ2VzIGludH00Zm90Zm90Zm90Zm90
dmV5ZWQgb3l0Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
IHc=
cm00Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
ludH00Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
YW5uZXkgVHJpdGh0Zm90Zm90Zm90Zm90Zm90Zm90Zm90Zm90
YW==
dGlzZSBvbiBjcmludG9ncmFwaHkgYW5kIH00Zm90Zm90Zm90Zm90
```

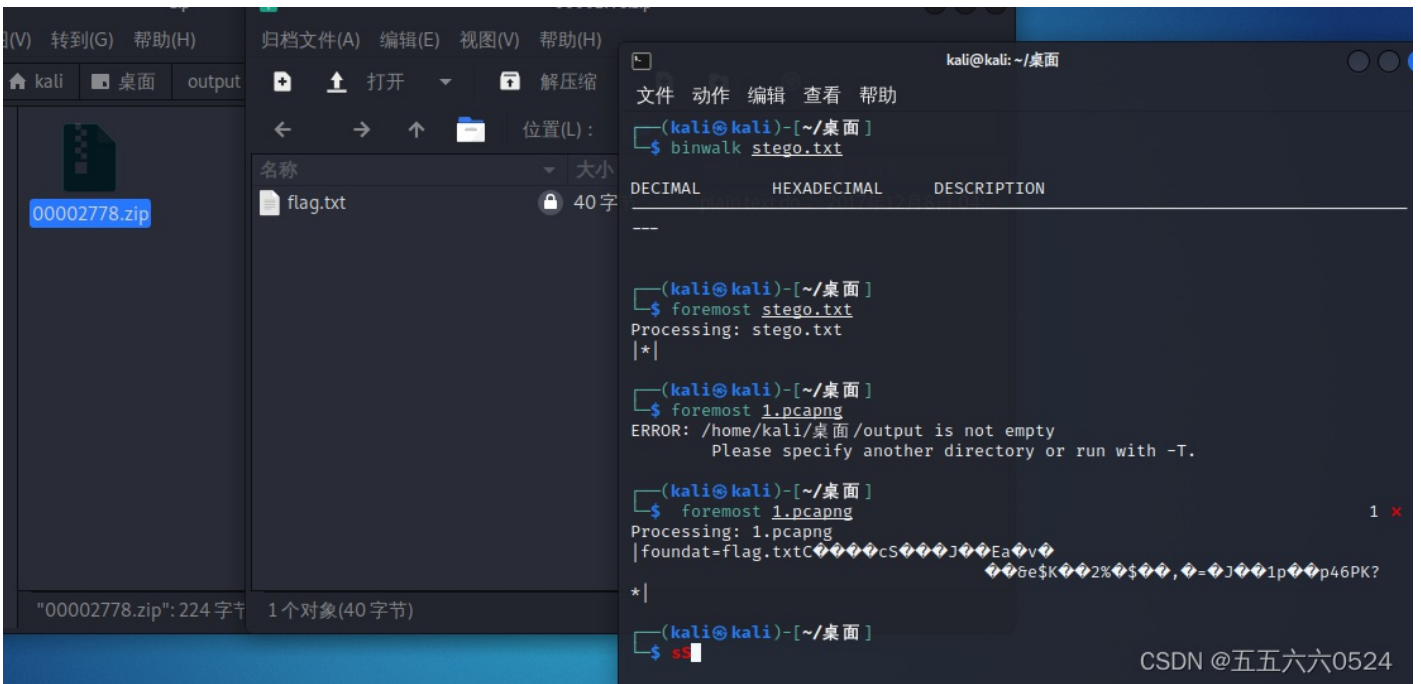
CSDN @五五六六0524

第四行的路径记得要改成绝对路径，flag{Base\_sixty\_four\_point\_five}

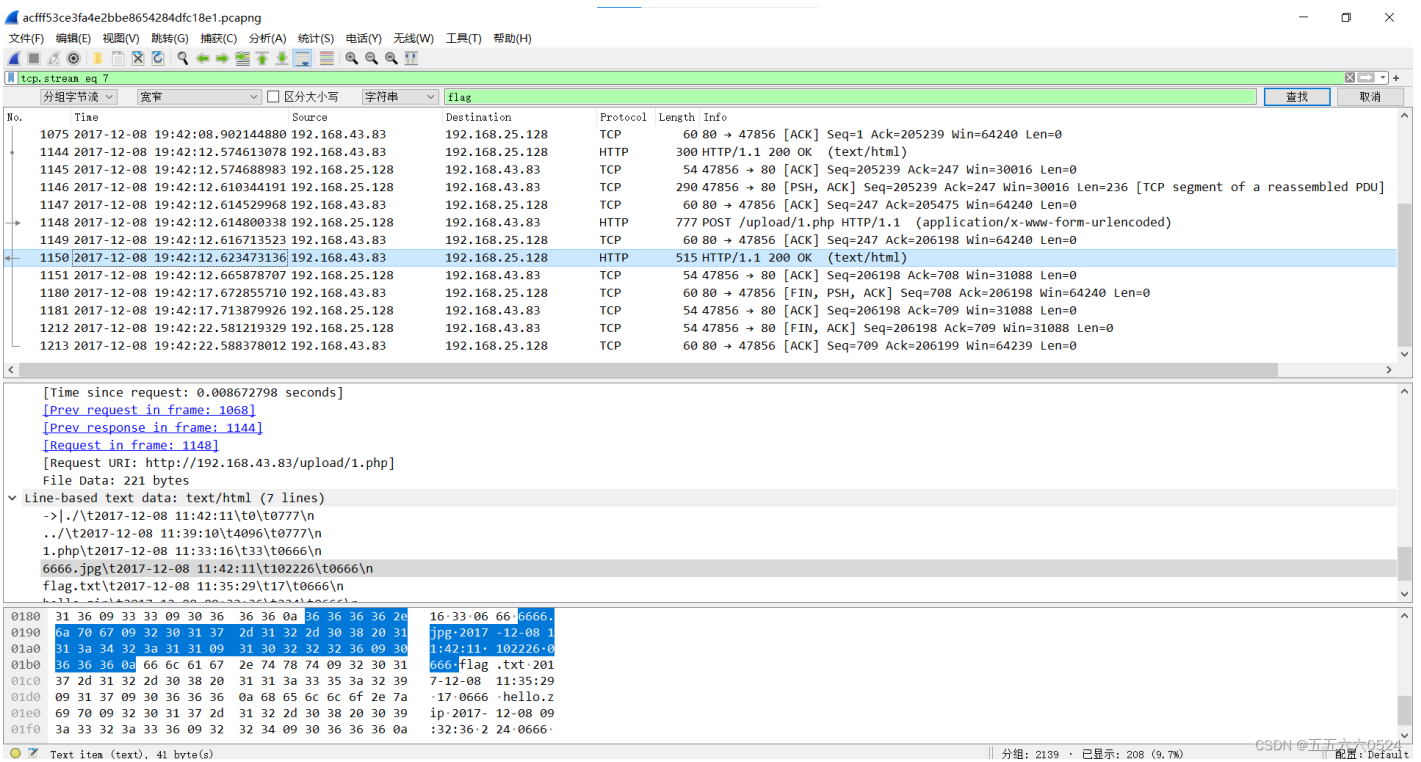
```
import base64
bin_str=''
b64chars = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
with open('stego.txt','r') as f:
    for line in f.readlines():
        stegb64=""
        rowb64=""
        offset=abs(b64chars.index(stegb64.replace('=', ''))[-1])-b64chars.index(rowb64.replace('=', ''))[-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

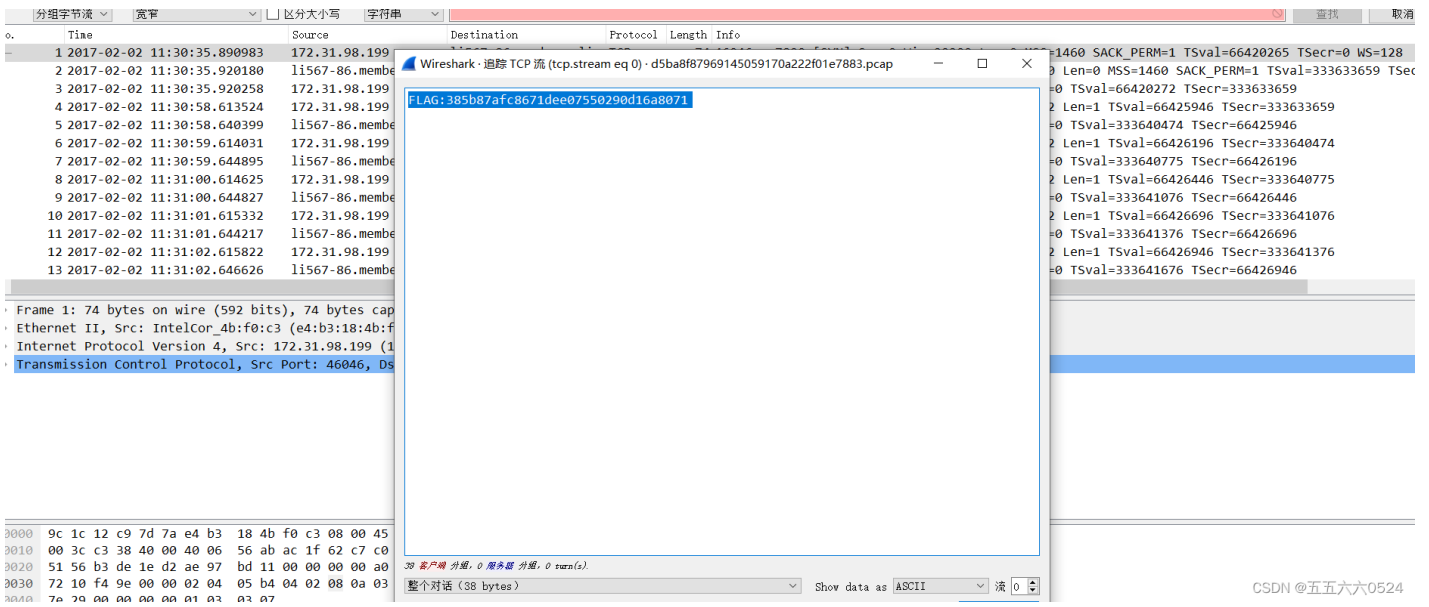


在wireshark中搜索flag.txt，只有1150最奇葩，有一个666.jpg









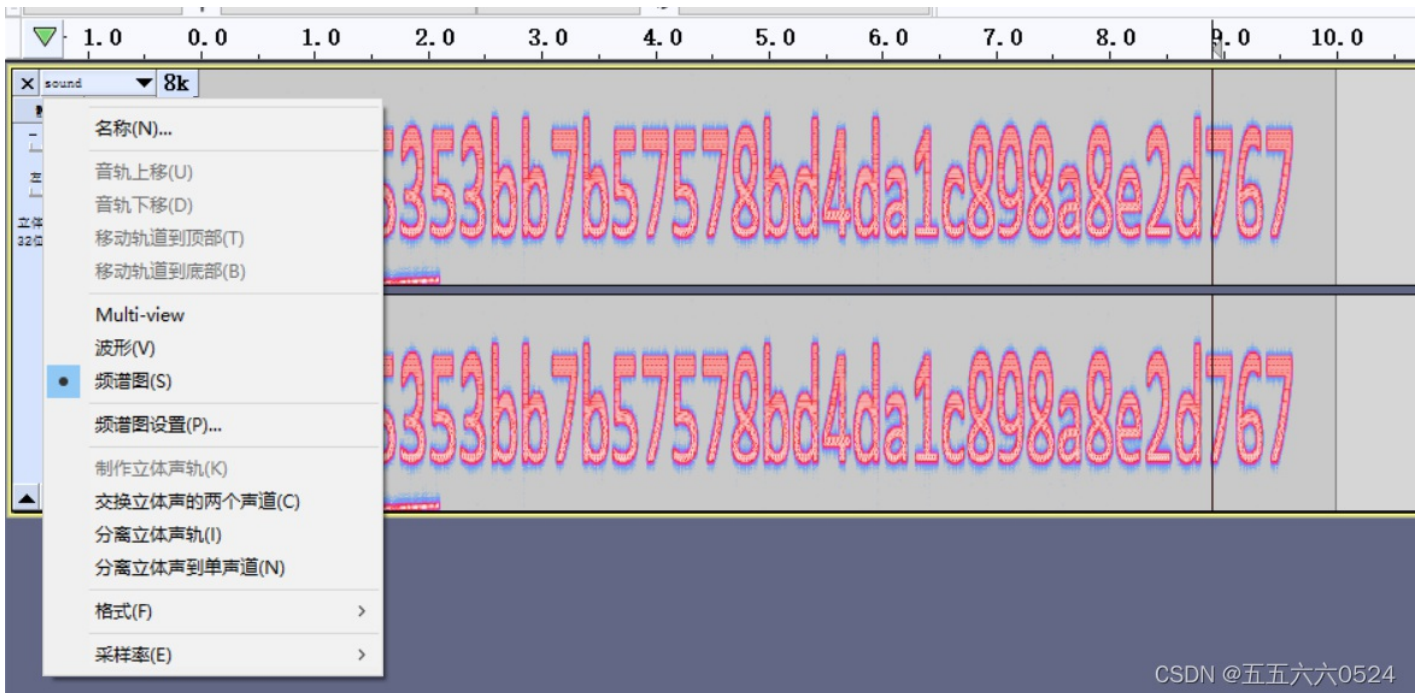
CSDN @五五六六0524

## 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

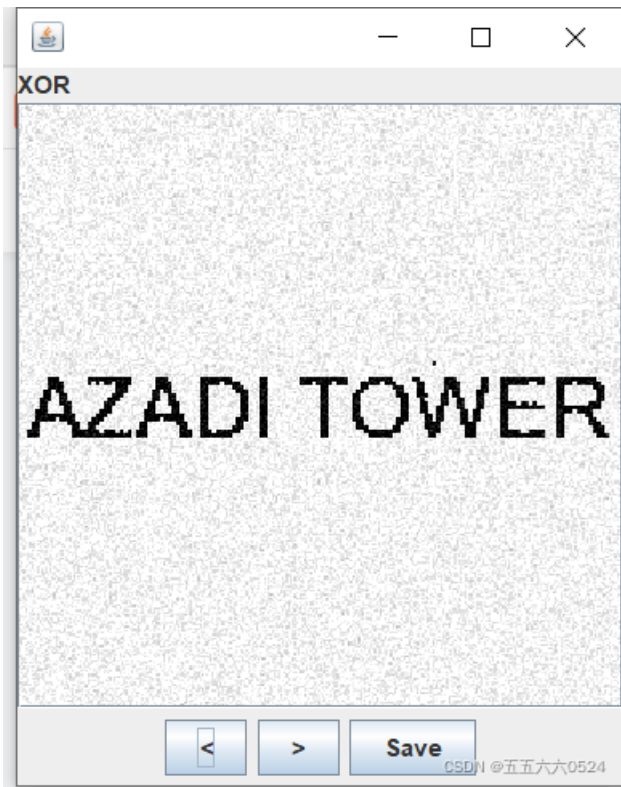
Bandzip能直接解压，得到一段音频wav，在Audacity中打开，看不出有什么不一样，根据题意我还以为是摩斯密码，但是形状对不上，点那个小三角，切换到频道图，flag就出来了，奇葩的是这个只要提交后面的字符串，全输进去还不对



CSDN @五五六六0524

## 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	!PNG IHDR
00000010	00	00	02	6C	00	00	03	6B	08	06	00	00	00	36	B4	F5	1 k 6'8
00000020	FD	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	y pHYs
00000030	13	01	00	9A	9C	18	00	00	0A	4D	69	43	43	50	50	68	! MiCCPPb
00000040	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66	otashan TCC prof





flag{B8B68DD7007B1E406F3DF624440D31E0}

CSDN @五五六六0524

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}

```
kali@kali: ~/桌面
文件 动作 编辑 查看 帮助
(kali@kali)-[~/桌面]
└─$ strings badimages | grep -i flag
.Flag.txt.swpe
Flag.txttt.swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txttt.swx
.Flag.txt.swpe
.Flag.txt.swpe
Flag.txttt.swx
.Flag.txt.swpe
.Flag.txt.swx
.Flag.txt.swpe
Flag.txttt.swx
Flag.txt
Flag.txt
Flag.txt
Flag.txt
/mnt/test/Flag.txt
Flag{}
Flag{yc4pl0fvjs2k1t7T}
/mnt/test/Flag.txt
Flag{}
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

85 2025 06 20 22:10:25 522602 102 168 1 102 61 172 207 120 100

< [Full request URI: <http://www.wooyun.org/user.php?action=login&do=login>]  
 [HTTP request 1/3]  
 [Response in frame: 26]  
 [Next request in frame: 48]  
 File Data: 65 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

- > Form item: "email" = "flag"
- > Form item: "password" = "ffb7567a1d4f4abdfdb54e022f8facd"
  - Key: password
  - Value: ffb7567a1d4f4abdfdb54e022f8facd
- > Form item: "captcha" = "BYUG"

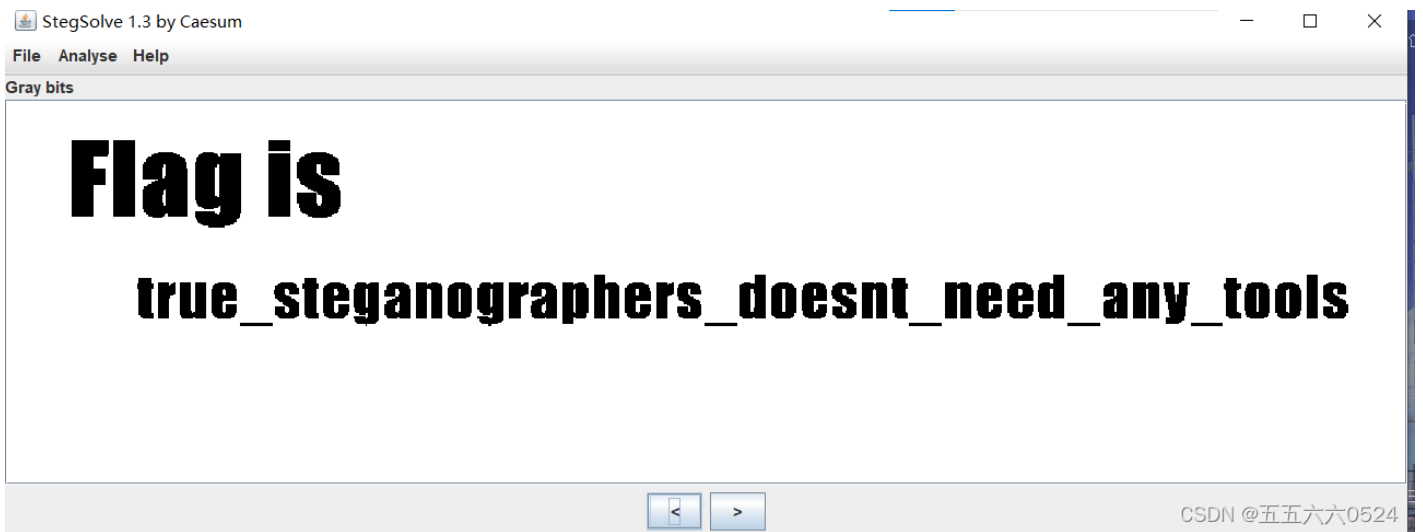
---

02c0	6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d	onnectio n: keep-
02d0	61 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 2d 54	alive···C ontent-T
02e0	79 70 65 3a 20 61 70 70 6c 69 63 61 74 69 6f 6e	ype: app lication
02f0	2f 78 2d 77 77 77 2d 66 6f 72 6d 2d 75 72 6c 65	/x-www-f orm-urle
0300	6e 63 6f 64 65 64 0d 0a 43 6f 6e 74 65 6e 74 2d	ncoded··· Content-
0310	4c 65 6e 67 74 68 3a 20 36 35 0d 0a 0d 0a 65 6d	Length: 65····em
0320	61 69 6c 3d 66 6c 61 67 26 70 61 73 73 77 6f 72	ail=flag &passwor
0330	64 3d 66 66 62 37 35 36 37 61 31 64 34 66 34 61	d=ffb756 7a1d4f4a
0340	62 64 66 66 64 62 35 34 65 30 32 32 66 38 66 61	bdfdb54 e022f8fa
0350	63 64 26 63 61 70 74 63 68 61 3d 42 59 55 47	cd&captc ha=BYUG

字节 818-849: Value (urlencoded-form value)

## 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..

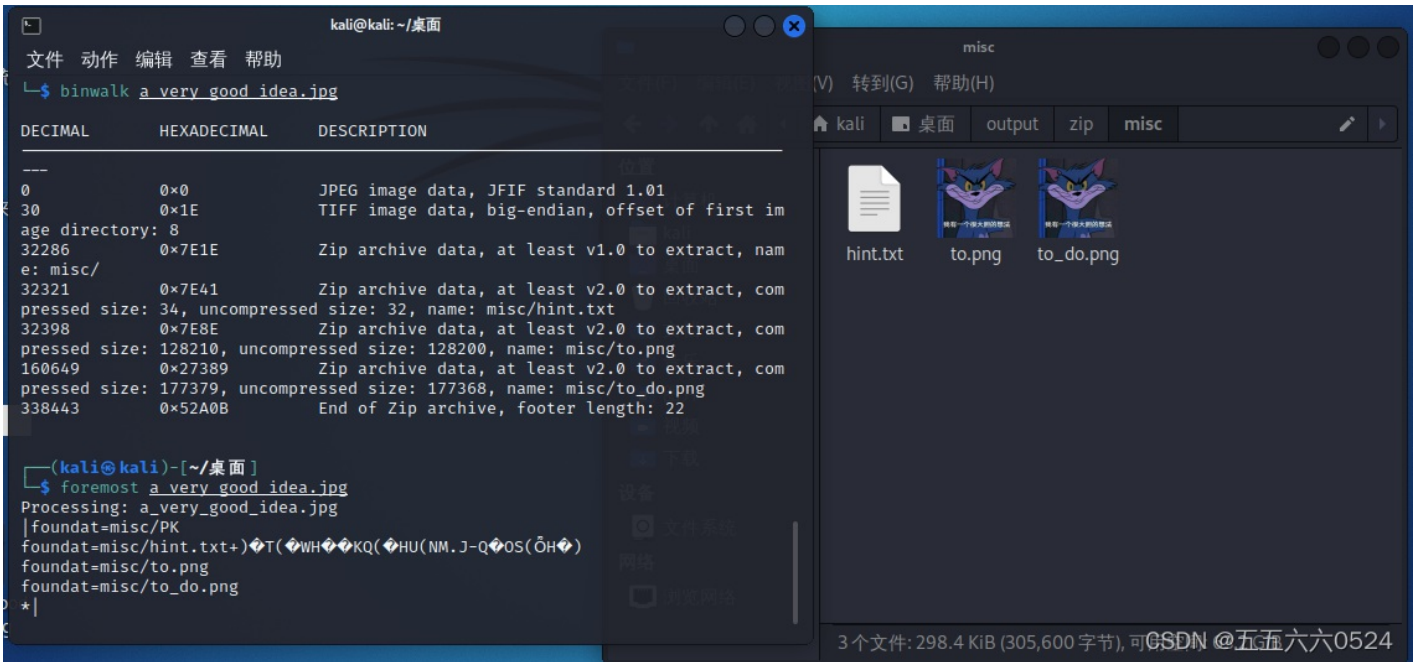
<https://blog.csdn.net/gongjingege/article/details/108059130>先把动图分解GIF动态图片分解，多帧动态图分解成多张静态图片\_图片工具网页版，得到一个ISCC，winhex打开动图，最后的一部分看起来不太对劲

```
- ,d=>e0w00ee A +1
Ä ki±² ;U2FsdG
VkX19QwGkcgD0fTj
ZxgijRzQ0GbCWALh
4sRDec2w6xsY/ux5
3VuJ/AMZBDJ87qyZ
L5kAf1fmAH40e13I
u435bfRBuZgHpnRj
TBn5+xsDH0NiR3t0
+0a8yG/tOKJMNuau
edvMyN4v4QKiFunw
==
```

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

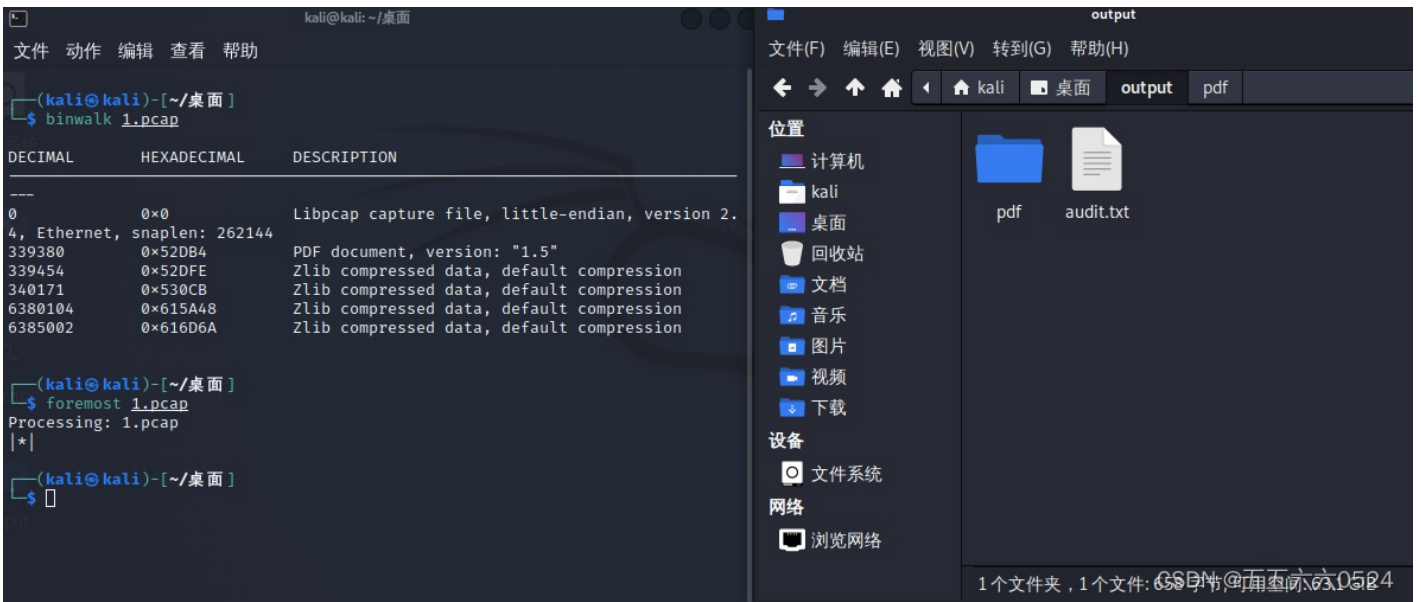


用stegsolve的图片合并功能，先打开第一张图片，然后Analyse-Image Combiner，点两下箭头直接得到二维码，得NCTF{m1sc\_1s\_very\_funny!!!}



## 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

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	42	4D	66	00	00	00	00	00	00	00	36	00	00	00	28	00	BMf 6 (
00000010	00	00	04	00	00	00	04	00	00	00	01	00	18	00	00	00	
00000020	00	00	30	00	00	00	00	00	00	00	00	00	00	00	00	00	0
00000030	00	00	00	00	00	00	4C	6F	6F	6B	20	77	68	61	74	20	Look what
00000040	74	68	65	20	68	65	78	2D	65	64	69	74	20	72	65	76	the hex-edit rev
00000050	65	61	6C	65	64	3A	20	70	61	73	73	77	64	3A	73	74	ealed: passwd:st
00000060	65	67	61	6E	6F	49											eganol

CSDN @五五六六0524 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
00000240 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 ++
00000260 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000270 20 20 20 20 20 20 20 20 0A 20 20 20 20 20 20 20
00000280 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 2B 2B 3E 2B 2B ++>++
000002A0 2B 2B 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
000002C0 20 20 20 20 20 20 20 0A 20 20 20 20 20 20 20
000002D0 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 +
00000300 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00000310 20 20 20 20 20 20 0A 20 20 20 20 20 20 20 20
00000320 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 >+++++++
00000340 3E 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
00000360 20 20 20 20 0A 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
00000380 20 20 20 20 20 20 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
000003A0 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
000003C0 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 3E 2B 2B ++++++++>++
000003E0 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
00000400 20 20 0A 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
00000420 20 20 20 2B 2B 2B 2B 2B 2B 2B 2B 3E 2B 2B 2B ++++++++>+++
00000430 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
00000450 20 0A 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
00000470 20 20 2B 2B 2B 2B 2B 2B 2B 2B 3E 2B 2B 2B 2B

```

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?!}

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	PK	çwDDÔ
00000000	50	4B	03	04	14	00	03	00	08	00	0E	A2	77	44	44	D4	PK	çwDDÔ
00000010	88	77	27	00	00	00	19	00	00	00	08	00	00	00	66	6C	lw'	f1
00000020	61	67	2E	74	78	74	00	10	01	4B	93	FF	03	EE	9C	FA	ag.txt	Kly ilú
00000030	D3	12	83	A1	57	88	57	8C	BF	41	AA	41	87	16	F6	85	Ó	llWllLlAAl öll
00000040	FE	40	02	DA	73	CA	1F	AC	16	97	89	44	3A	50	4B	01	p@ ÚsÊ ~ llD:PK	
00000050	02	14	00	14	00	03	00	08	00	0E	A2	77	44	44	D4	88		çwDDÔll
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	ll6
00000090	00	00	00	4D	00	00	00	00	00								M	

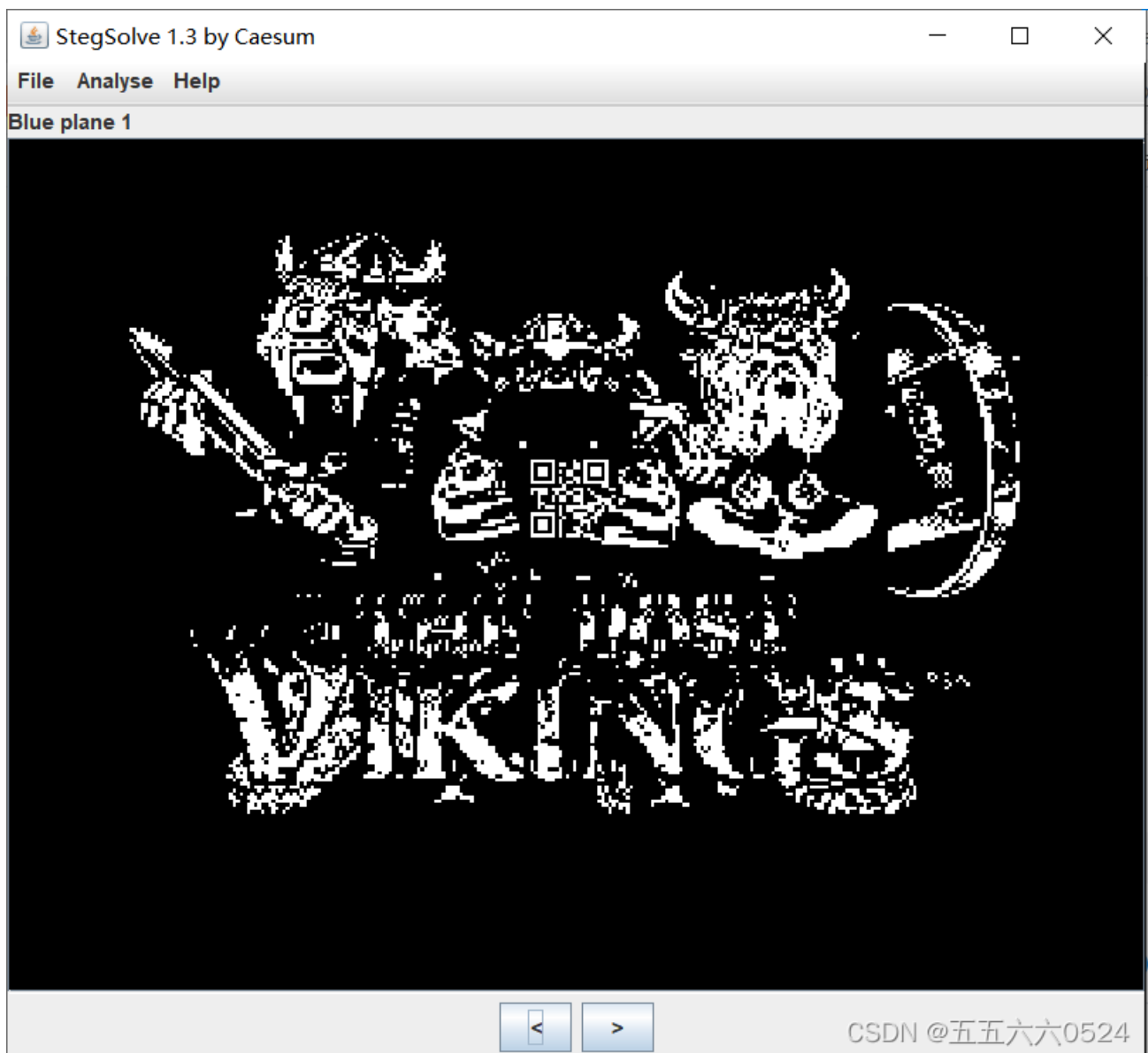
### 19、Erik-Baleog-and-Olaf



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	I à ³K
00000020	B3	00	00	68	81	49	44	41	54	78	DA	ED	BD	6F	72	DB	³ h IDATxÚi¼orÚ
00000030	4C	BA	E5	09	57	F4	26	FC	C1	DD	60	F5	2E	1C	F1	C6	L²á Wò&üÁÝ`ð. ñÆ
00000040	0C	B5	0D	45	79	5A	56	EF	62	14	8E	09	B7	EE	2E	46	µ EyZVib I ·i.F
00000050	56	B7	2B	D4	BB	B8	62	77	BC	11	DE	C5	14	71	EF	FB	V·+Ô»_bw¼ ÞÁ qiù
00000060	41	BB	A8	9A	62	9E	84	70	12	4F	3E	C9	04	41	0A	12	A»"IbIlp O>É A
00000070	75	7E	1F	68	38	09	24	12	09	50	91	07	CF	BF	A6	11	u~ h8 \$ P' İ¿
00000080	42	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	B I I B I I B I I B
00000090	08	21	84	10	42	08	21	84	10	42	08	21	84	10	42	08	I I B I I B I I B

得到一张图片，根据文件名，肯定是要用stegsolve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}

stego100	misc10																
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	36	36	36	63	36	31	36	37	37	62	36	38	36	35	36	63	666c616777b68656c
00000010	36	63	36	66	35	66	37	37	36	66	37	32	36	63	36	34	6c6f5f776f726c64
00000020	37	64															7d

CSDN @五五六六0524

## 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}

```
kali@kali: ~/桌面
文件 动作 编辑 查看 帮助
putchar
_Jv_RegisterClasses
_ITM_registerTMCloneTable
_ITM_deregisterTMCloneTable
__gmon_start__
UH-
ffffff.
AWAVA
AUATL
[]A\A]A^A_
cvqAeqacLtqazEigwiXobxrCrtuiTzahfFreqc{bnjrKwgk83kgd43j85ePgb_e_rwqr7fvbmHjkl
o3tews_hmkogooyf0vbnk0ii87Drfgh_n kiwutfb0ghk9ro987k5tfb_hjiouo087ptfcv}
;*3$"
(q9e
aliases
ethers
group
gshadow
hosts
initgroups
netgroup
networks
passwd
protocols
publickey
services
shadow
CSDN @五五六六0524
```

## 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)

CSDN @五五六六0524

题目是一张动漫图，总感觉高度不对，04改成05，得到一个StRe1izia

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	CSDN @五五六六0524



## StRe1izia

CSDN @五五六六0524

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>
    
    ZmxhZ3tPel80bmRfSGlyMF9sb3YzX0ZvcjN2ZXJ9
  </body>
</html>
```

GET /kiss.png HTTP/1.1

CSDN @五五六六0524

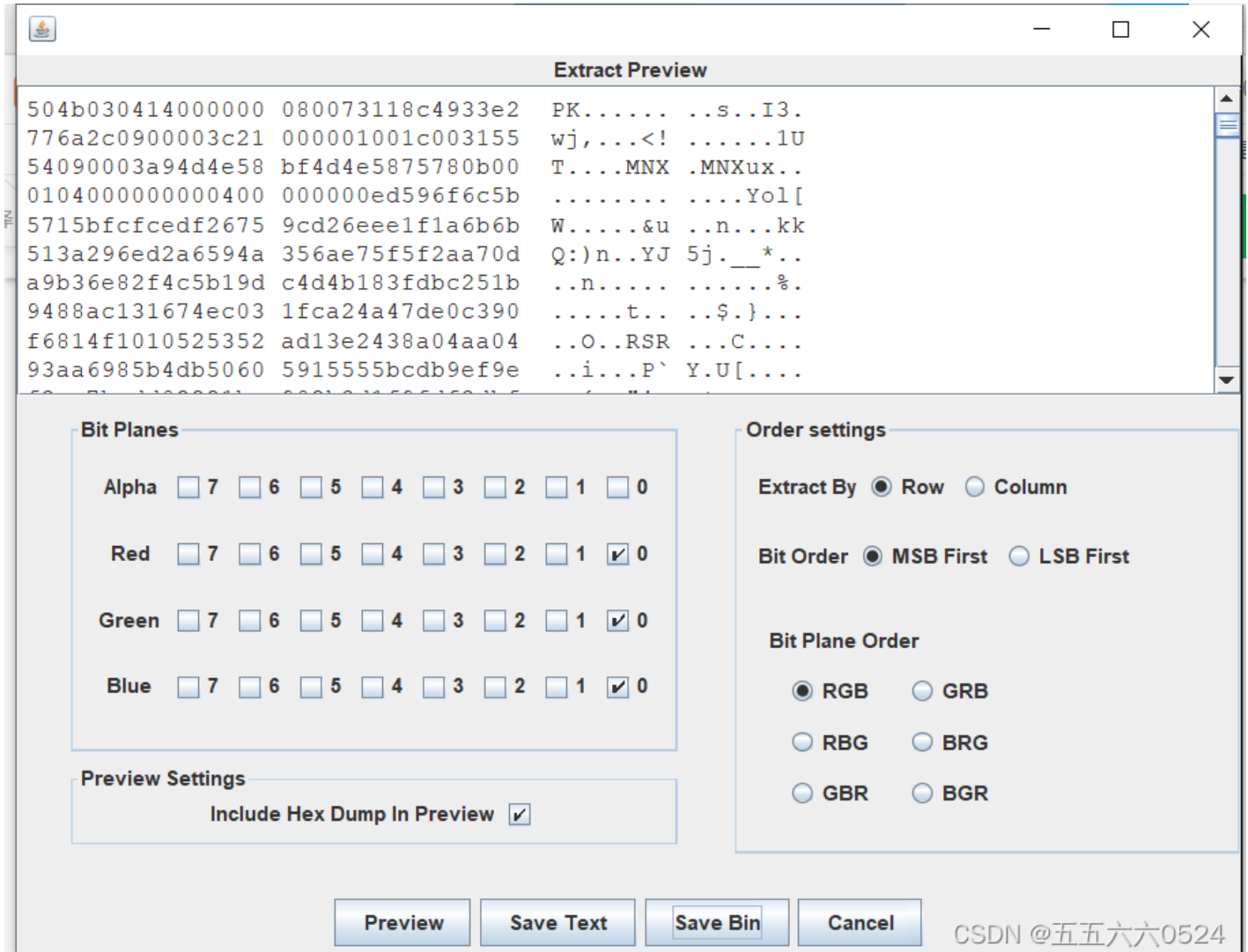
Zmxh的开

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stegsolve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|i H|Ã 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 | ln~~~}
00000600 | 00 00 05 00 00 00 04 FE FF FF 7C 00 00 00 44 FE | ;0
00000610 | FF FF 4C 00 00 00 31 FF FF FF A4 00 00 00 54 FF | pÿ| Dp
| iÿÿÿ Ty
```



## 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 Application Read Code39, Code128, PDF417, DataMatrix, QR, and other barcodes from TIF, PDF and other image documents <https://online-barcode-reader.inliterearch.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@inlitteresearch.com](mailto:support@inlitteresearch.com)  
email your sales inquiries to [sales@inlitteresearch.com](mailto:sales@inlitteresearch.com)

---

**File:** 斑马斑马.png New File

**Pages:** 1 **Barcodes:** 1

---

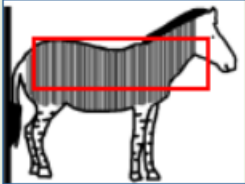
**Barcode:** 1 of 1 **Type:** Code39

**Length:** 16 **Rotation:** none

**Module:** 1.6pix **Rectangle:** {X=71,Y=93,Width=410,Height=119}

FLAG IS **TENSHINE**

Page 1 of 1



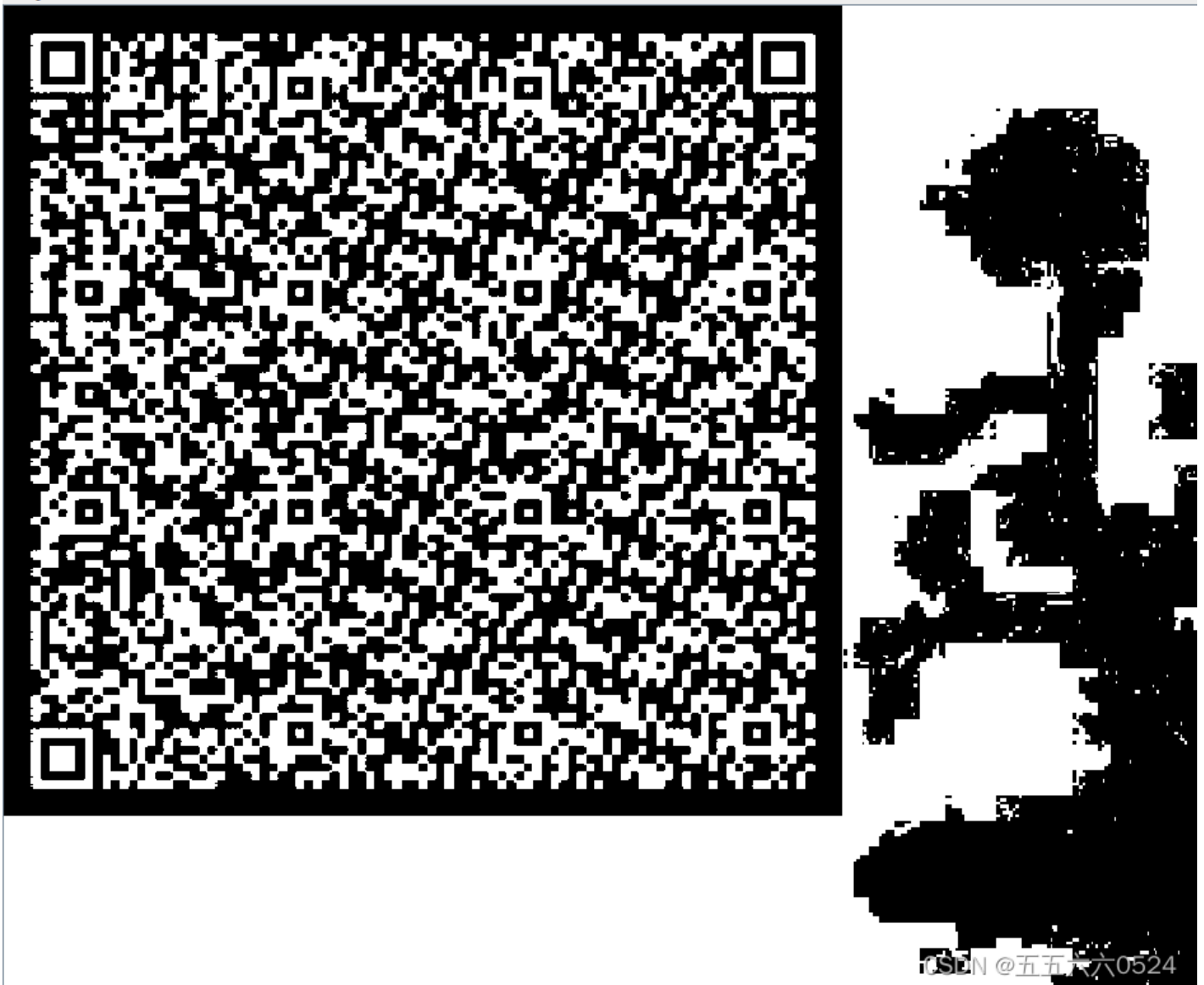
CSDN @五五六六0524

## 26、stage1

还是熟悉的图片，stegslope得到，扫一下出来一堆十六进制的数



Gray bits



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

1.pyc																			
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F			
00000000	03	F3	0D	0A	B6	26	6A	57	63	00	00	00	00	00	00	00	00	ó	¶&jWc
00000010	00	01	00	00	00	40	00	00	00	73	0D	00	00	00	64	00	00	@	s d
00000020	00	84	00	00	5A	00	00	64	01	00	53	28	02	00	00	00	00	!	Z d S(
00000030	63	00	00	00	00	03	00	00	00	08	00	00	00	43	00	00	00	c	C
00000040	00	73	4E	00	00	00	64	01	00	64	02	00	64	03	00	64	00	sN	d d d d
00000050	04	00	64	05	00	64	06	00	64	05	00	64	07	00	67	08	00	d	d d d d g
00000060	00	7D	00	00	64	08	00	7D	01	00	78	1E	00	7C	00	00	00	}	d } x
00000070	44	5D	16	00	7D	02	00	7C	01	00	74	00	00	7C	02	00	00	D]	©SDN @五五六六0524
00000080	00	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	!	73

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

```
*1.pyc_dis
文件(F) 编辑(E) 搜索(S) 选项(O) 帮助(H)
# Embedded file name: test.py

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

    print flag
```

CSDN @五五六六0524

改一下把数值扔进去就是

```
str = [
    65,
        108,
        112,
        104,
        97,
        76,
        97,
        98]

flag = '03F30D0AB6266A57630000000000000000100000040000000730D0000006400008400005A0000640100532802000000630'
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
b1,r,lsb,xy .. file: JPEG image data, JFIF standard 1.01, aspect ratio, density 100x100, segment length 16, Exif Standard: [TIFF image data, big-endian, direntries=10, manufacturer=Canon, model=Canon EOS DIGITAL REBEL XTi, orientation=upper-left, xresolution=168, yresolution=176, resolutionunit=2, software=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]
└─$ ss
```

CSDN @五五六六0524



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

```
(root@kali)~/桌面/RECOVERED_FILES
# ./flag
zsh: 权限不够: ./flag

(root@kali)~/桌面/RECOVERED_FILES
# chmod 777 flag
126 x

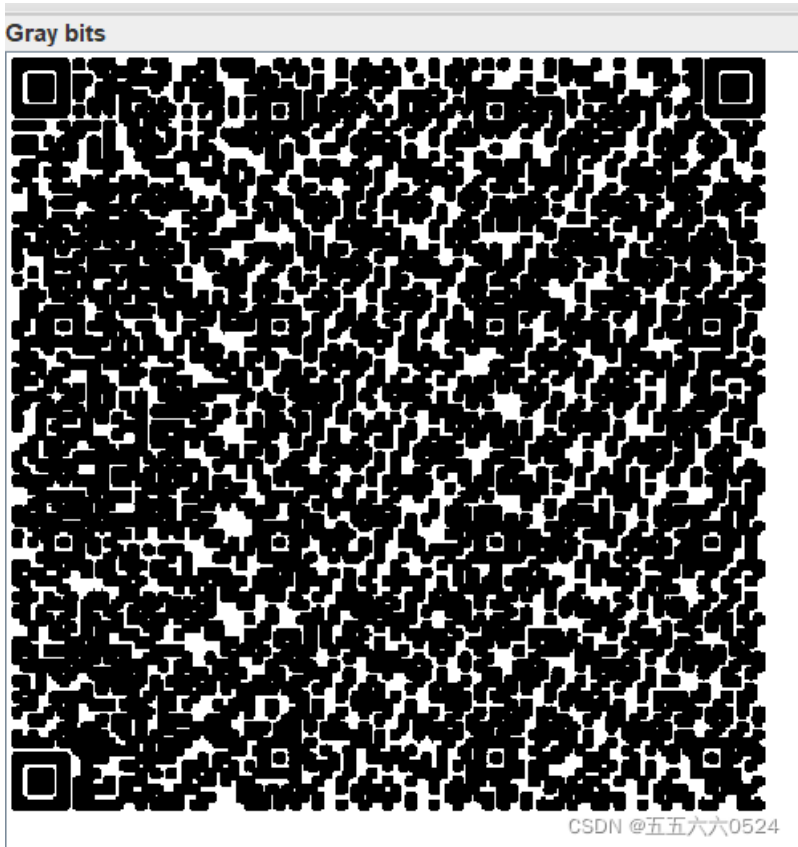
(root@kali)~/桌面/RECOVERED_FILES
# ./flag
your flag is:
de6838252f95d3b9e803b28df33b4baa

(root@kali)~/桌面/RECOVERED_FILES
# s$
```

CSDN @五五六六0524

### 30、适合作为桌面

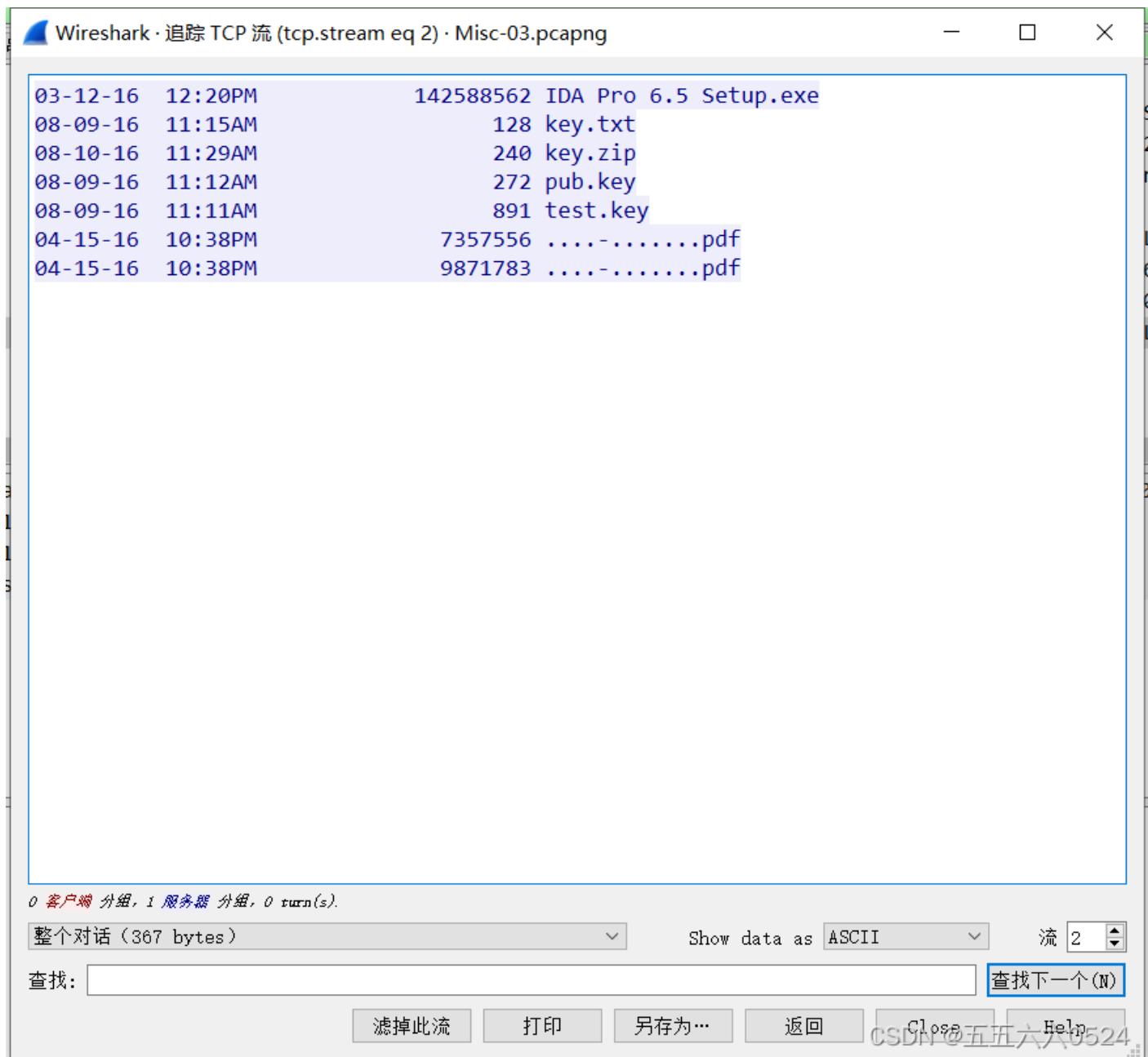
stegslope得到一张二维码，关键是我扫不出来啊，wp攻防世界Misc 适合作为桌面\_hhh-CSDN博客\_攻防世界适合作为桌面，flag{38a57032085441e7}



### 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

```

-----BEGIN RSA PRIVATE KEY-----
MIICXgIBAAKBgQD0UN0A+70iM0VCJ1ni0n/U1BRj0u8yMWH4Qi+xTbjHgbE7wOuk
OaO+2PyQXiQIzZnf5jCkJuVDYjALGcKrZM40CQBBd85B/LTc36XZ7JVfX5kGy5tI
R3tquuPIVKNdAshlSgh9S7YSS39RdnSa5rOUyGhrLzxwzzM9IO4e+QQ+CQIDAQAB
AoGADiaw5mGubtCxbkeBOVYf+V/fXnjVSf76QbrzsD1kOooUjfv6sKR2C5Pd7S7H
H+1owENBBgEKvoBtb/cqA2tvU9vQ4l5TMBJcHv6LEcb9WpPnMxPV2GNjO+DTPGPy
Xnu1UZlZjwx+NaF5rESoSsVS2ZaaIxBs4RWRXk+lHEbTFECQD6Rp6jMweRgPHO
pR3mgIK83zL+kzqYM5isIPv3DIC5JQN2kXqK73IDQCFVlfXnr9lAAVRzLDsAXLqv
le/o6yQLAKEA+edY+GERlLuD1t2k9Js0Dc7EwnLcxoFUE60ivj8Gf9jzLskGHxsv
0IV6J50HwPh54kAxAnqCjSqNRAWGNzr+uwJBALYEjDum1LdGrxXZ0jAkgHC6Z0zs
aK3uwHdXGcinqCp+t9EQpq3KzQF+L4AeKxRQONEq5m9I2LQ/vGocwrMD4dcCQQDb
rTyOinWz8upAFPKOe2hUwvA/pkzgyosoCMhDyI9kD0gmVlv1ODbd7Jem9o8dWM97
zcXHUF41LbSkMN6U6m1FAkEAqmZbr35bPfkeoiikwNl6OVQyTg12TZjw2vIbvub
f9Rvti8Lh/tbrmhZroiz8/l3aAZmugI1NBcbeZR0gz8ggg==
-----END RSA PRIVATE KEY-----

```

分组 358。0 客户端 分组, 1 服务器 分组, 0 turn(s)。点击选择。

整个对话 (891 bytes) Show data as ASCII 流 20

查找:  查找下一个 (N)

滤掉此流 打印 另存为... 返回 C13c8 五五六六0524 help

foremost流量包会得到jpg、pdf、zip，zip解压后得到key.txt，用openssl解密，flag是 {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{yOud1any1s1}

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ARCHPR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, 我折腾了好久, 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

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

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

CSDN@五五六六0524

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

## 在线二进制转换字符串[EN] - 转换

输入二进制文本:

```
01001000011001010110100101110011011001010110111001100010011001010111001001100111
```

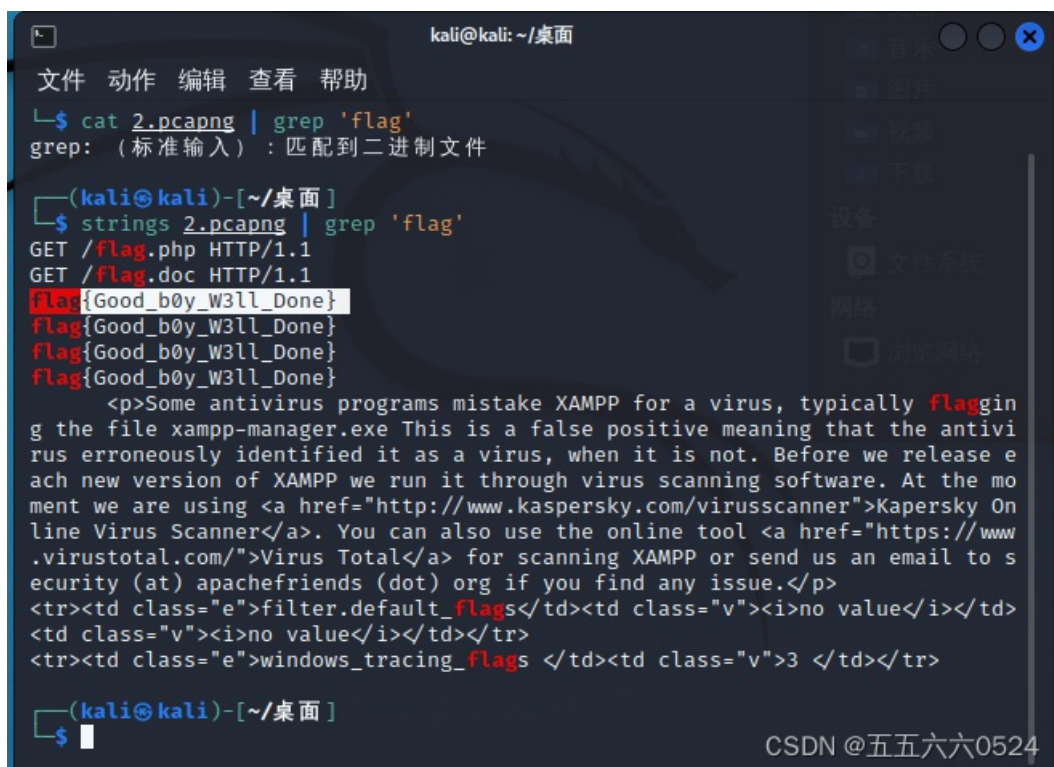
转换后的文本:

```
Heisenberg
```

CSDN @五五六六0524

### 36、embarrass

wireshark里没有分析出来什么，拖到linux里出了，flag{Good\_b0y\_W3ll\_Done}



```
kali@kali: ~/桌面
文件 动作 编辑 查看 帮助
└─$ cat 2.pcapng | grep 'flag'
grep: (标准输入) : 匹配到二进制文件

(kali@kali)-[~/桌面]
└─$ strings 2.pcapng | grep 'flag'
GET /flag.php HTTP/1.1
GET /flag.doc HTTP/1.1
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}
flag{Good_b0y_W3ll_Done}
<p>Some antivirus programs mistake XAMPP for a virus, typically flagging the file xampp-manager.exe This is a false positive meaning that the antivirus erroneously identified it as a virus, when it is not. Before we release each new version of XAMPP we run it through virus scanning software. At the moment we are using <a href="http://www.kaspersky.com/virusscanner">Kaspersky Online Virus Scanner</a>. You can also use the online tool <a href="https://www.virustotal.com/">Virus Total</a> for scanning XAMPP or send us an email to security (at) apachefriends (dot) org if you find any issue.</p>
<tr><td class="e">filter.default_flags</td><td class="v"><i>no value</i></td></tr>
<tr><td class="v"><i>no value</i></td></tr>
<tr><td class="e">windows_tracing_flags </td><td class="v">3 </td></tr>

(kali@kali)-[~/桌面]
└─$
```

CSDN @五五六六0524

### 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

No.	Time	Source	Destination	Protocol	Length	Info
3540	2018-06-12 20:19:31.006459	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	65	Response
3547	2018-06-12 20:19:33.503345	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3548	2018-06-12 20:19:33.504797	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	64	Response
3558	2018-06-12 20:19:36.544843	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3559	2018-06-12 20:19:36.545912	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	89	Response
3561	2018-06-12 20:19:36.920861	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3562	2018-06-12 20:19:36.921701	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	64	Response
3566	2018-06-12 20:19:39.694155	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3567	2018-06-12 20:19:39.694696	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	65	Response
3573	2018-06-12 20:19:40.973707	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3574	2018-06-12 20:19:40.974641	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	67	Response
3579	2018-06-12 20:19:45.658709	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3580	2018-06-12 20:19:45.659654	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	66	Response
3583	2018-06-12 20:19:47.105938	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query
3584	2018-06-12 20:19:47.108030	192.168.130.130	DESKTOP-6G4UE22.loc...	Modbus...	99	Response
3586	2018-06-12 20:19:50.161419	DESKTOP-6G4UE22.loc...	192.168.130.130	Modbus...	66	Query

Byte Count: 36

- ▼ Register 1 (UINT16): 99
  - Register Number: 1
  - Register Value (UINT16): 99
- ▼ Register 2 (UINT16): 116
  - Register Number: 2
  - Register Value (UINT16): 116
- ▼ Register 3 (UINT16): 102
  - Register Number: 3
  - Register Value (UINT16): 102

```

0000 00 50 56 c0 00 08 00 0c 29 02 23 7c 08 00 45 00  ·PV·····)·#|··E·
0010 00 55 76 ed 40 00 40 06 3d e1 c0 a8 82 82 c0 a8  ·Uv·@·@· =·····
0020 82 01 01 f6 f3 1a ab 23 0f 2b fd ff 60 ee 50 18  ·····#·+···`P·
0030 00 e5 d4 66 00 00 00 01 00 00 00 27 01 03 24 00  ···f····· ···'·$·
0040 63 00 74 00 66 00 7b 00 45 00 61 00 73 00 79 00  c·t·f·{· E·a·s·y·
0050 5f 00 4d 00 64 00 62 00 75 00 73 00 7d 00 00 00  _·M·d·b· u·s·}···
0060 00 00 00

```

CSDN @五五六六0524

## 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 -xvf +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
```

CSDN @五五六六0524

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
```

CSDN @五五六六0524

git stash show显示做了哪些改动，默认show第一个存储,如果要显示其他存储，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(-)
```

CSDN @五五六六0524

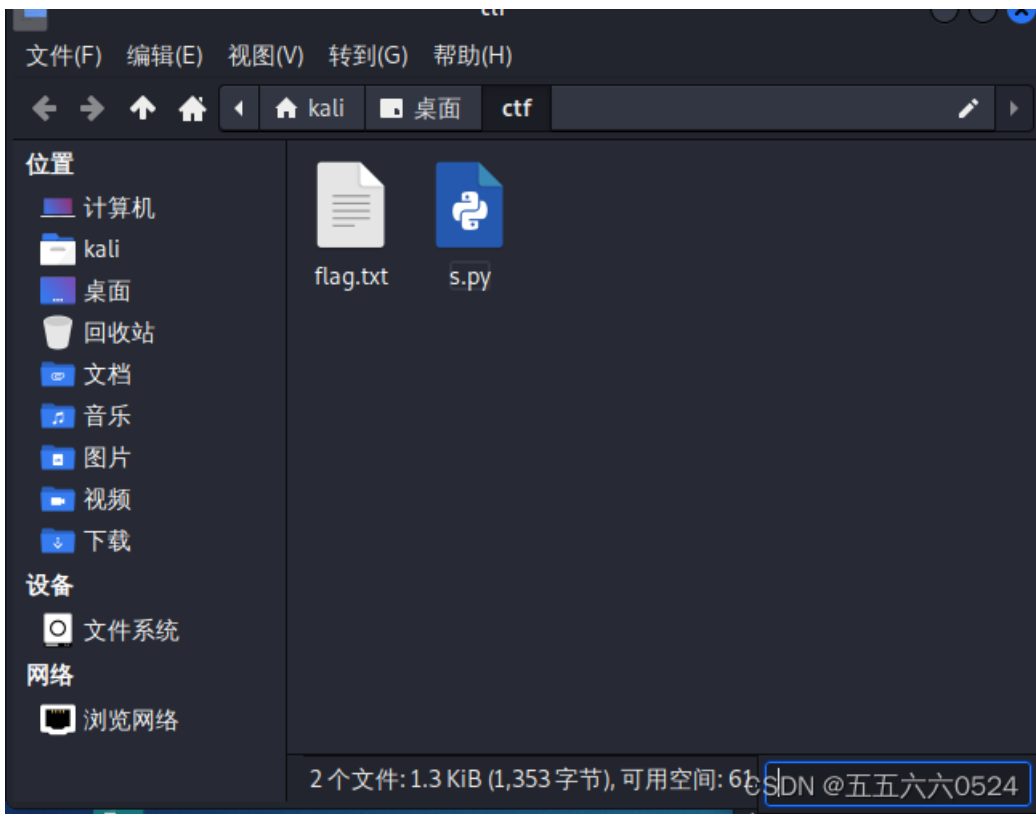
git stash apply应用某个存储,但不会把存储从存储列表中删除，默认使用第一个存储,即stash@{0}，如果要使用其他个，git stash apply stash@{num}，比如第二个：git stash apply stash@{1}，相当于恢复改变的内容

```
(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
```

CSDN @五五六六0524





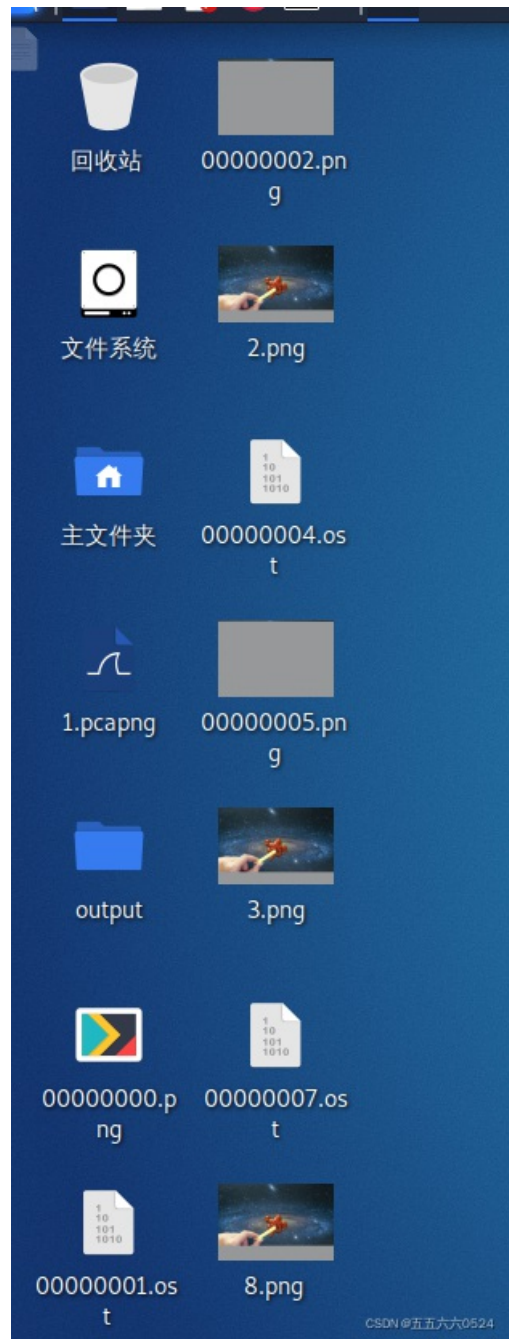
运行s.py，改一下路径，直接出NCN4dd992213ae6b76f27d7340f0dde1222888df4d3

### 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、tcpextract -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

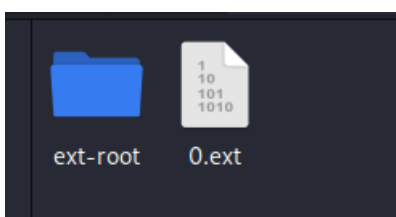
拖进kali里查看文件类型

```

└─$ file forensic100
forensic100: Linux rev 1.0 ext2 filesystem data, UUID=0b92a753-7e09-42b0-b052-49

```

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}



```

└─$ 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, 0xfffffc 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*Iσ'z-6`"
b3,g,msb,xy .. text: "ChvVe!'c~"
b3,rgb,msb,xy .. text: "2mI$AAσ`"
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: "σt235SV^"
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)

```

iVBORw0KGgoAAAANSUHeUgAAAPoAAAD6CAYA AACI7Fo9AAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwAAEnQAABJ0Ad5mH3gA
AAVqSURBVHhe7d1bTuRGAEDRlffK1txgNRJNINbpf9xzJmvlpP6q5KpeB5u3v334Bt/bX57/AjQkdAoQOAUKHAKFDgNAhQOgQIHQIEDoECB0ChA4BQoc
Le39///zf9/xScwSY0SFA6BAgdAgQOgQIHQKEDgGLv7329vb2+b/1PDqVZ8dZ67uCe13HV1sc95GRcRo9x7Xel8aZ0SFA6BAgdAgQOgR4GLdw/99d/5xz2/
q61jD6Hp/pWuo2CX3NN3jrJbuf/T6t76uNVzhHJnHrTsECB0ChM6fW/RHG/chdAgQOgQIHQJO8330Z6fx7DhrfYtni7XonHPb+rpe8eoYHHGOjDGjX9AU5t
wNjklNzww9aOM+hA4BQocAoUOA0ChAb68N7P+o1z7zaJ9r72+y1tizHzM6BAgdAoQOAZddo8/107mMrEO3eO3aRt6Lrcee/ZjRIUDoECB0CBA6BAgdAh
Y/dQeuw4wOAUKHAKFDgNAhQOgQIHQIEDoECB0ChA4BQocAoUOA0CFA6BAgdAgQOgQIHQKEDgFChwAfJcVDI3+cgvMZCv3RF8PaXwhb/7WQO/81k
pFY5772u/E789jUuHWHAKFDgNAhQOgQcMmHcWseY/Sh09rXexaj4zK569hckRmdIVPMzzbOQ+gQIHQIEPqNTevnRxs9t/7JuDnssUX/InWp8+ubY9xOc
sY8MGMDgFChwChQ4DQleD0D+PmWnoulw+sJmcag69Gr22uM48BH8zoECB0CBA6BKy+Rp/rLOvEkWt45izr09E1+lBv75L9nmVMr+rUoY9GuMcxHnllSJ
cef+Ta5p7fVu/vk2+Mqb8n1t3CBA6BAgdAoT07qb19qsbY4YexgHXyEaHAKFDgNAhQOgQIHQIEDoECB0ChA4BQocAoUOA0CFA6BAgdAgQOgT4NdW
T0eoz7146Lvswo0AOCFA6BBw6jX62uvGK6wZrdHZwmF/wGHy06GFvo451/3suCPv0RXGu0LJ3NUOEK/N2t0CBA6BAgdALZfo6+5LhvZ/9bntpWj1rzPjvwT
MY86X15jRocAoUOA0W/eT+e5WelmR8Rp1hfGuMKNdW0qhTzPDnA3YjxdAoQOAUKHAKFDwC6/f/7o4ducwy593WTKtUf67kHlluf/7LhL36fJFca7wowOA
UKHAKFDwOVCn9aDczbgP4sfxo3GNPKQZ6mfjrnW8UYeQu19zf96dtyR92lkHFIXW3cIEDoECB0ChA4Bu/xk3FJHPZiCuzl16MA63LpDgNAhQOgQIHQIEDo
ECB0ChA4BQocAoUOA0CFA6BAgdAgQOgQIHQKEDgFChwChQ8DQJ8yMftTTWT7c5tF1rH1ul2PIQ4AYdckZFyrm6wY859YdAoQOAUKHgNUfxu3x4Gjt4+
F+LUBP89M/LFNGL8d32P-Ql-A-37h-...f01-1-1UQKED-GUMAF-...dLh-...G-nh-...B-...K-hL-...Y-M-...A-...Y-d-...L-...Y-...N-...P-...D-...

```



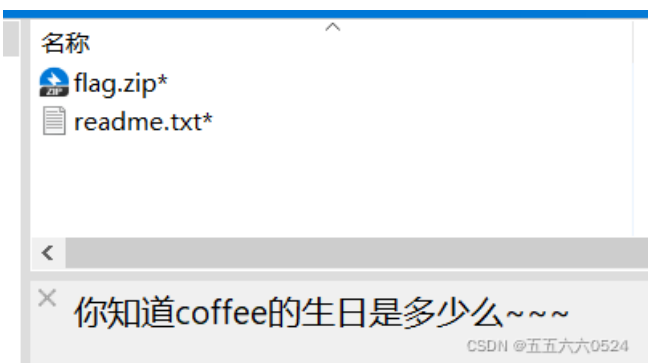
FLAG{LSB\_i5\_SO\_EASY}

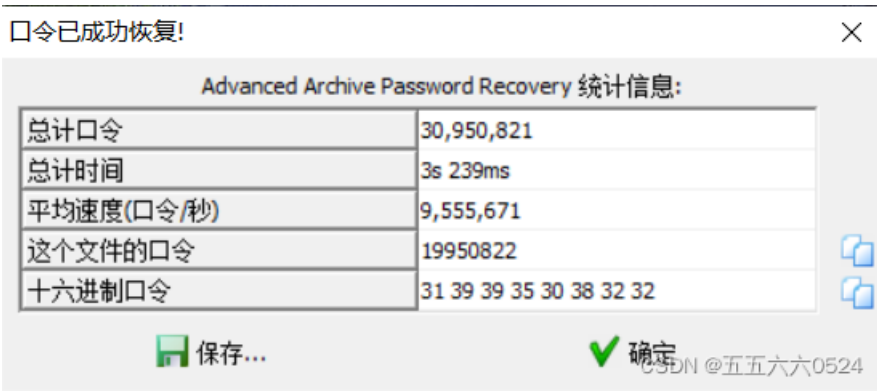
#### 44、我们的秘密是绿色的

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

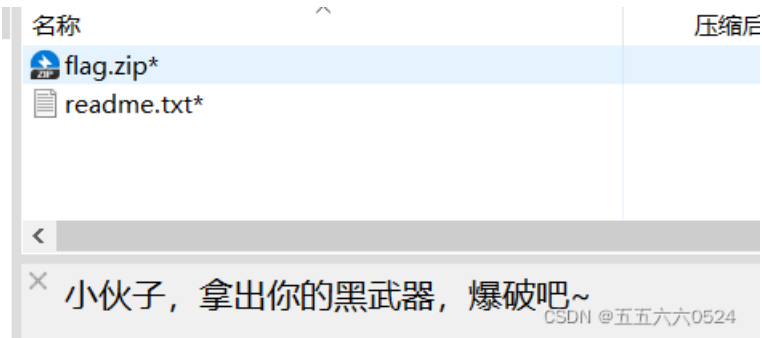


得到try.zip，有提示，爆破选纯数字，年月日共8位

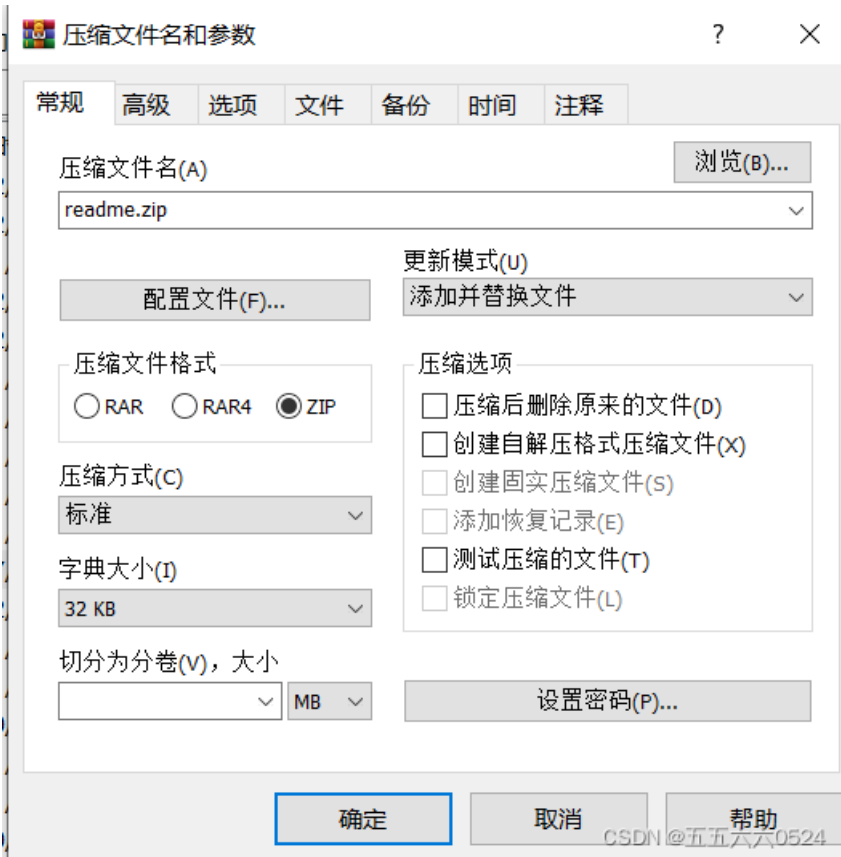




密码就是19950822，还需爆破，这个应该就是明文攻击了

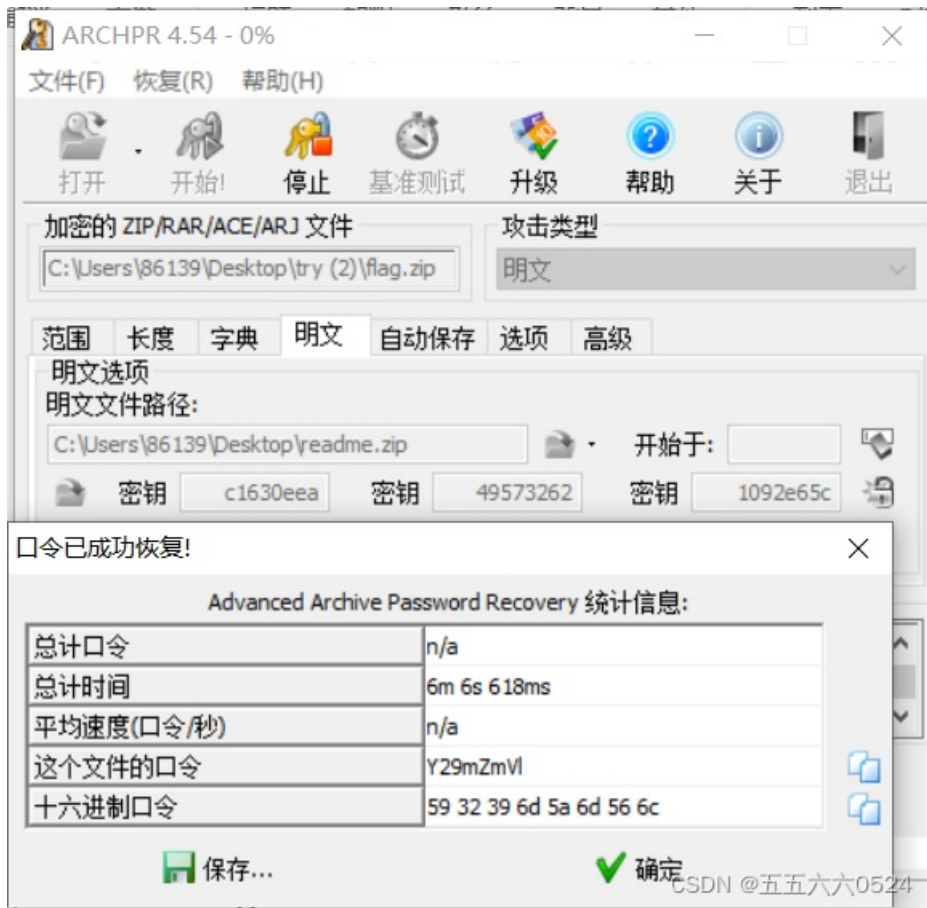


把readme.txt压缩了（不要用bandzip，否则会显示“在选定的档案中没有匹配的文件”，要用winrar），先选中那个文件，然后点添加，直接确定就行了



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

flag.zip	Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	00000000	50	4B	03	04	14	00	00	08	08	00	66	76	94	4A	7D	AF	PK fvIJ}~
	00000010	72	9F	1E	00	00	00	1E	00	00	00	08	00	00	00	66	6C	r! fl
	00000020	61	67	2E	74	78	74	2B	4C	49	29	28	2C	CF	2B	48	2E	ag.txt+LI)(,i+H.
	00000030	C8	49	CD	53	2D	28	02	B2	E3	AB	E3	AB	AA	B4	52	1C	EiIS-( 2ã«ã«a`R
	00000040	D2	0B	6B	01	50	4B	01	02	3F	00	14	00	01	09	08	00	ò k PK ?
	00000050	66	76	94	4A	7D	AF	72	9F	1E	00	00	00	1E	00	00	00	fvIJ}~r!
	00000060	08	00	24	00	00	00	00	00	00	00	20	00	00	00	00	00	\$
	00000070	00	00	66	6C	61	67	2E	74	78	74	0A	00	20	00	00	00	flag.txt
	00000080	00	00	01	00	18	00	E6	FC	D6	7E	A2	B9	D2	01	2C	E6	æü0~ç'ò ,æ
	00000090	57	65	82	B9	D2	01	2C	E6	57	65	82	B9	D2	01	50	4B	We!`ò ,æWe!`ò PK
	000000A0	05	06	00	00	00	00	01	00	01	00	5A	00	00	00	44	00	Z D
	000000B0	00	00	00	00													

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



```
qddpqwnpcplen%prqwn_{_zz*d@gq}|
```

每组字数

```
qwlr{ddneq_@dpnwzgp%nzqpp_*}
```

CSDN @五五六六0524

qwlr{ddneq\_@dpnwzgp%nzqpp\_\*}, 这玩意又是凯撒加密, 偏移量是11, 出  
flag{ssctf\_@secllover%coffee\_\*}, 这一题一套接一套的, 真真麻烦

明文:

```
flag{ssctf_@secllover%coffee_*}
```

偏移量

11

密文:

```
qwlr{ddneq_@dpnwzgp%nzqpp_*}
```

CSDN @五五六六0524

## 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

```
(kali@kali)-[~/桌面]
└─$ binwalk 1.png
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 649 x 487, 8-bit/color RGB, non-interlaced
41	0x29	Zlib compressed data, default compression

CSDN @五五六六0524

发现标红lsCc\_2019

```
(kali@kali)-[~/桌面]
└─$ zsteg 1.png
imagedata      .. text: "\t\t\t\r\r\r\r\r\r\r\r"
b1,rgb,lsb,xy  .. text: "IsCc_2019"
b2,r,msb,xy    .. text: "t^y\t_{!i0"
b2,g,msb,xy    .. text: "UUUUUU`\rUUUU"
b2,b,msb,xy    .. text: "UUUUUU`\rUUUU"
b2,rgb,msb,xy  .. text: "jZ]?0]k0"
b4,r,lsb,xy    .. text: "#UwcDS#z"
b4,r,msb,xy    .. text: ["f" repeated 8 times]
b4,g,lsb,xy    .. text: "w17ffd2T3EB"
b4,g,msb,xy    .. text: "wwwwwwwwwwww3{"
b4,b,lsb,xy    .. text: "ffffwww"
b4,b,msb,xy    .. text: "ffffffffffff\j"
b4,rgb,lsb,xy  .. text: "iVugVUUU6"
b4,bgr,lsb,xy  .. text: "YevWUUUV*SDN @五五六0524"
```

直接提交不对，根据题目反转一下，flag{9102\_cCs}（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

```
(kali@kali)-[~/桌面]
└─$ file 1.rock
1.rock: ASCII text
```

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](#)

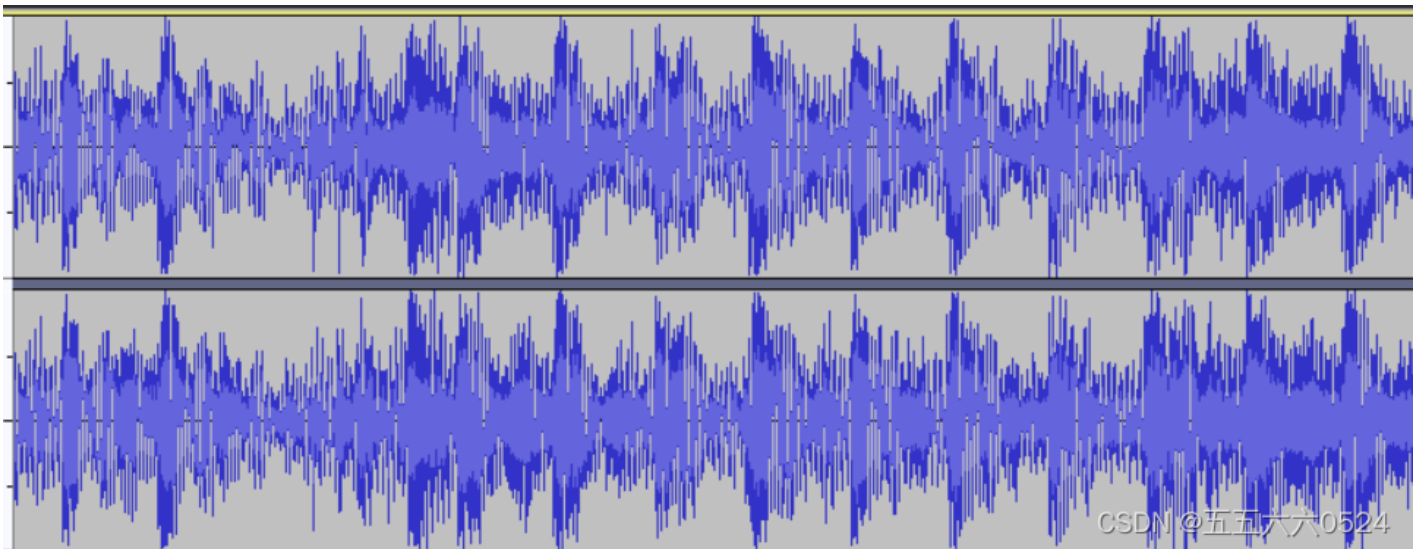
```

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

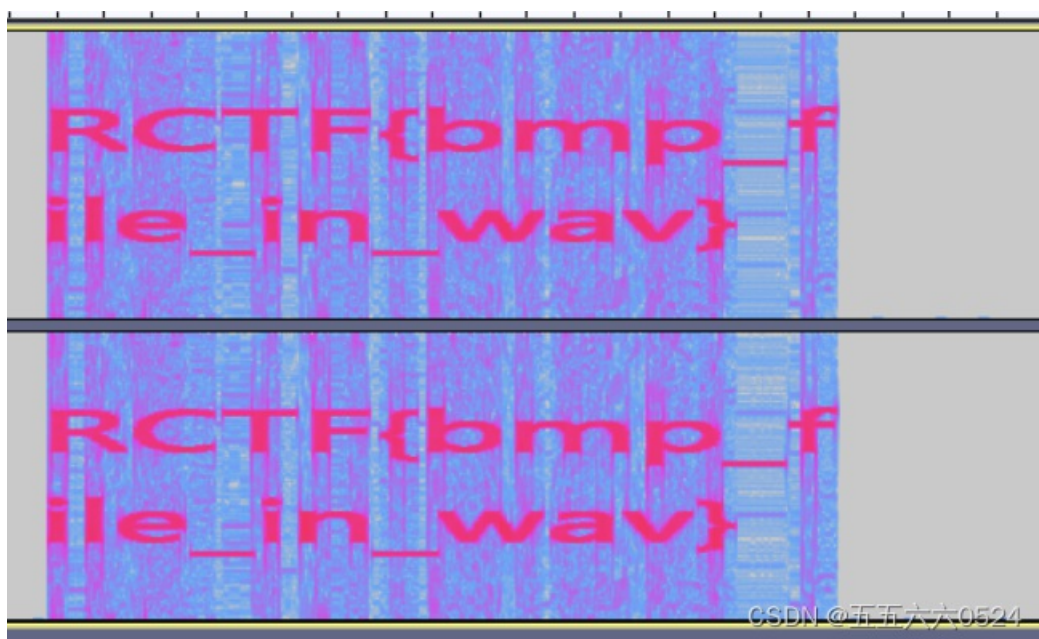
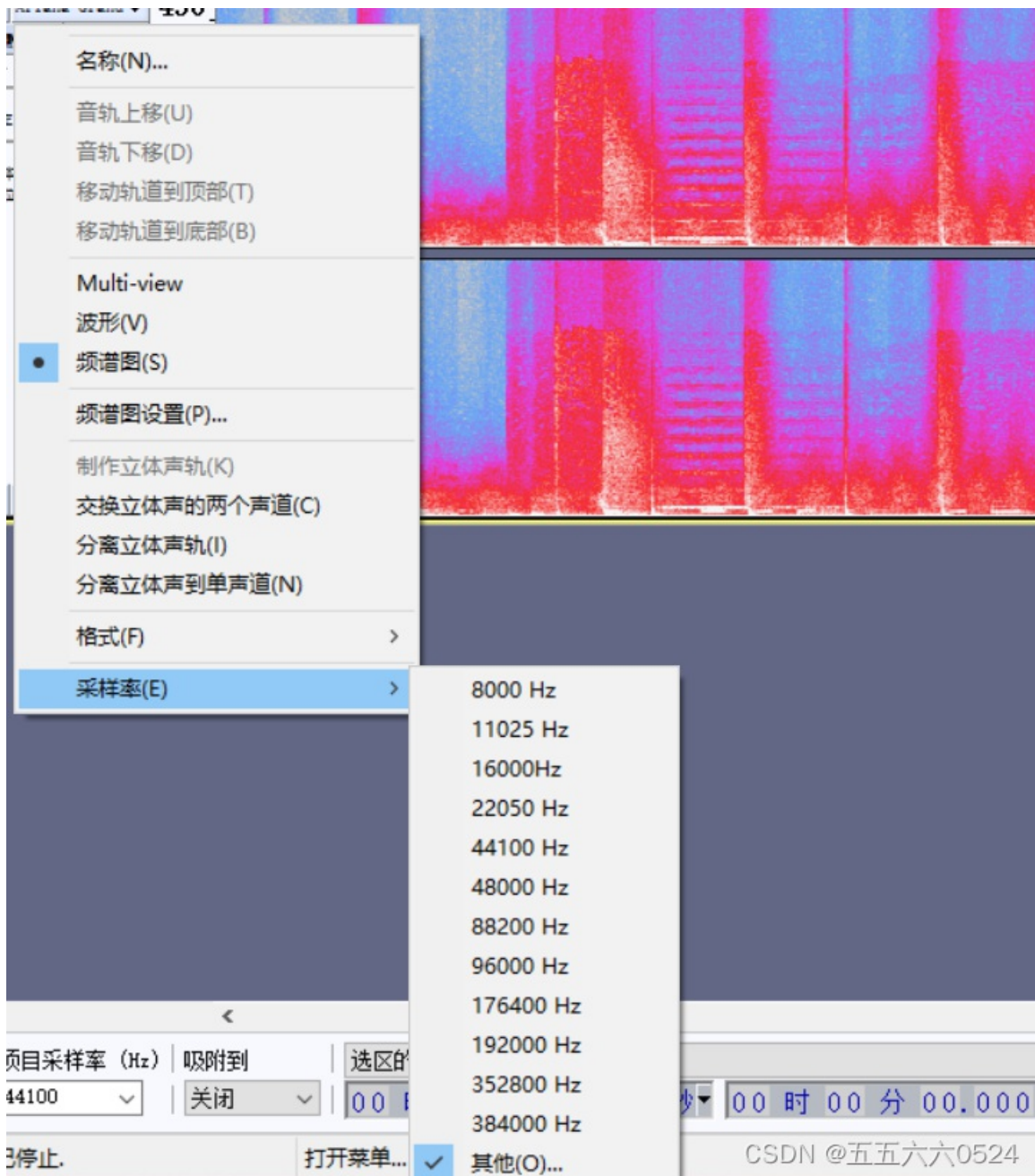


binwalk发现有东西但提取不出来

DECIMAL	HEXADECIMAL	DESCRIPTION
1406134	0x1574B6	MySQL MISAM index file Version 5
2411758	0x24CCEE	LZ4 compressed data, legacy
4459729	0x440CD1	MySQL ISAM index file Version 1
6503687	0x633D07	MySQL MISAM compressed data file Version 6
7094494	0x6C40DE	MySQL ISAM compressed data file Version 5
7210389	0x6E0595	MySQL ISAM compressed data file Version 2
7509633	0x729681	MySQL MISAM index file Version 3
7590019	0x73D083	MySQL MISAM index file Version 7
9374200	0x8F09F8	MySQL ISAM compressed data file Version 4
10011194	0x98C23A	MySQL MISAM index file Version 1
10084330	0x99DFEA	MySQL MISAM compressed data file Version 7
10738445	0xA3DB0D	MySQL MISAM index file Version 5
11269025	0xABF3A1	MySQL MISAM compressed data file Version 4
12322425	0xBC0679	MySQL MISAM index file Version 5

[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

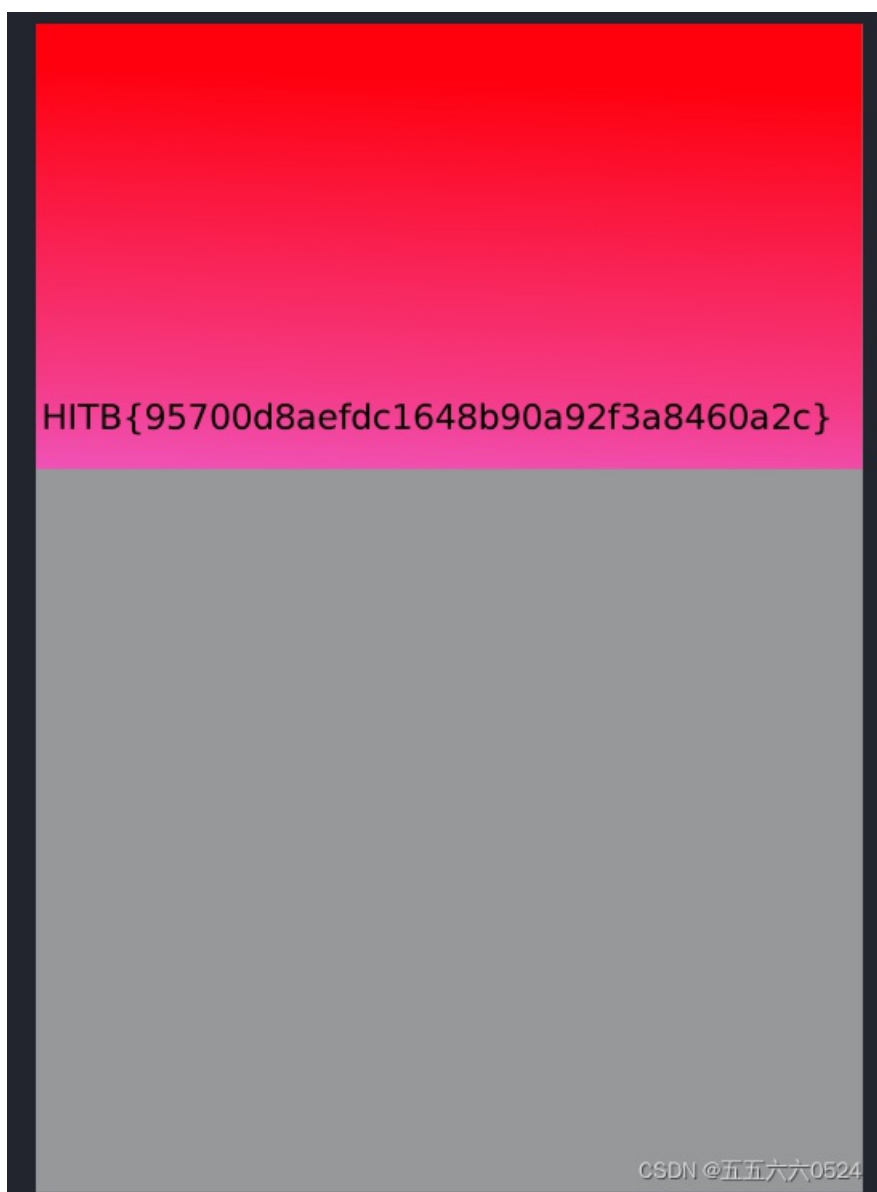
把流量包扔进wireshark，搜flag能出来，应该是里面有图片

```
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	.....

CSDN @五五六六0524

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



## 50、Excaliflag

stegslope在蓝色通道里找到了，3DS{Gr4b\_Only\_th1s\_B1ts}



