ctfshow-Misc入门图片篇(1-34)

原创

御七彩虹猫 于 2021-10-14 20:01:50 发布 371 收藏 4

分类专栏： CTFSHOWMISC 文章标签： web安全 安全

Misc-图片篇（基础操作&信息附加）

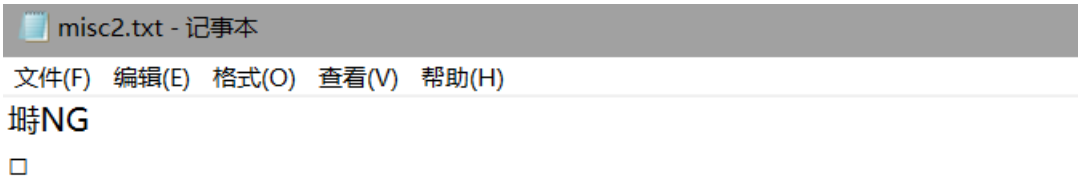
misc1

签到题下载下来解压打开就能发现flag

ctfshow{22f1fb91fc4169f1c9411ce632a0ed8d}

misc2

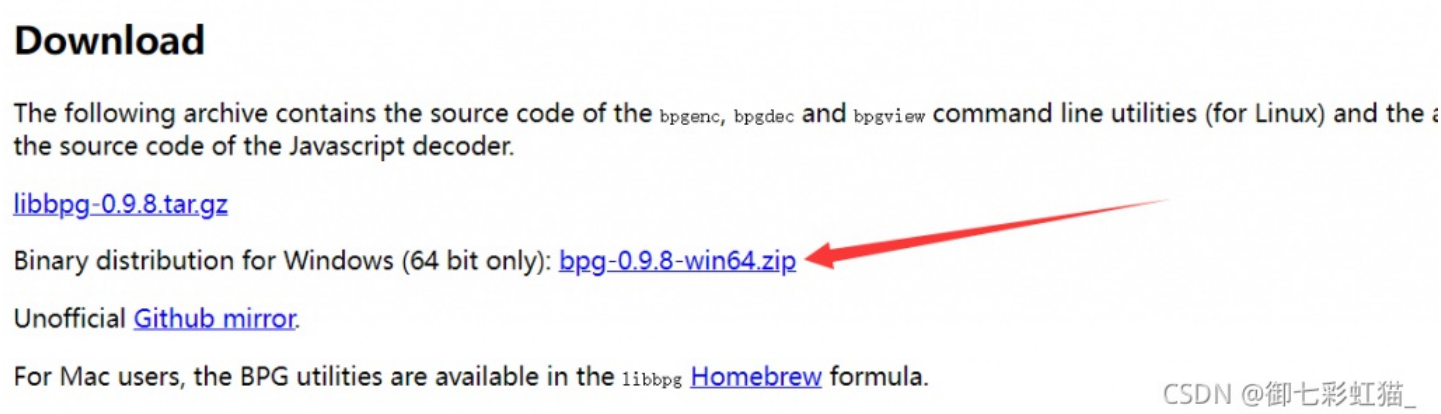
下载下来发现是个TXT文件，打开发现是个乱码文件，然后查看文件头如下图



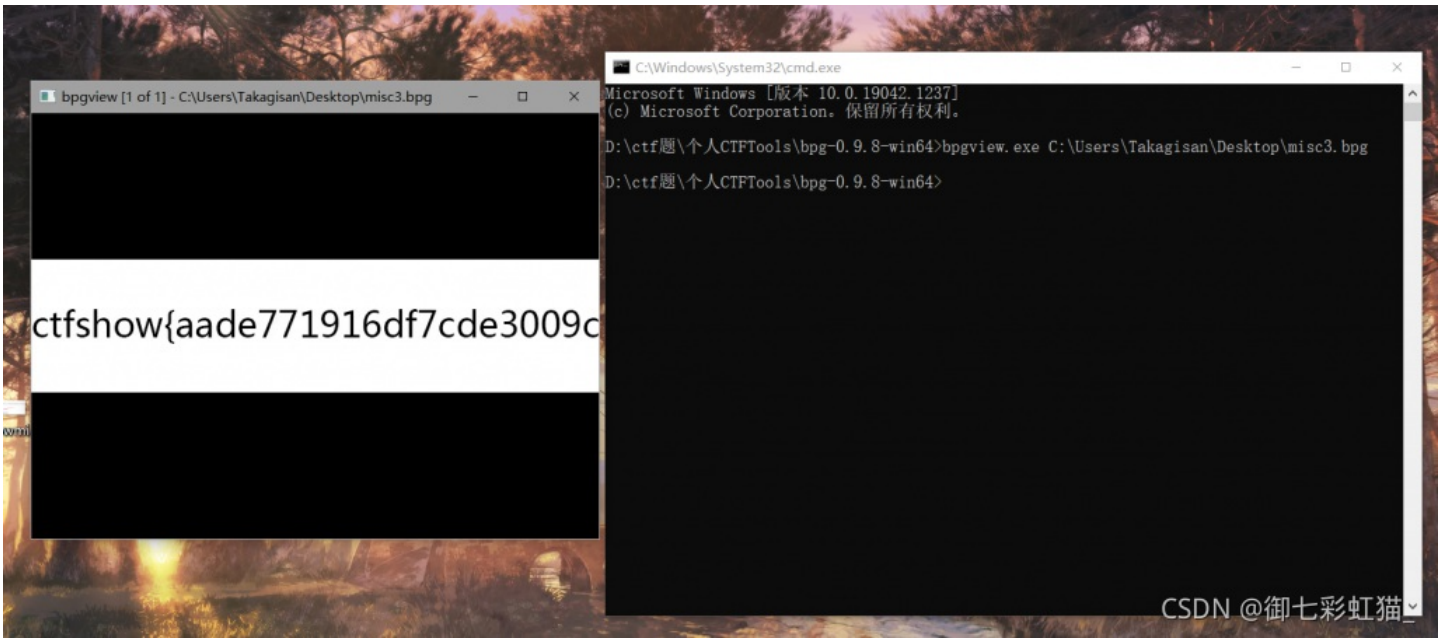
发现是PNG文件的文件头，我们只需要修改文件后缀flag就出来了修改后缀为.png然后打开图片flag就出来了

misc3

下载文件下来发现是一个bpg文件打不开，我们用在线网站打开，然后下载文件
软件下载



软件不能直接打开使用，如图命令操作然后flag就能出来了，如下图



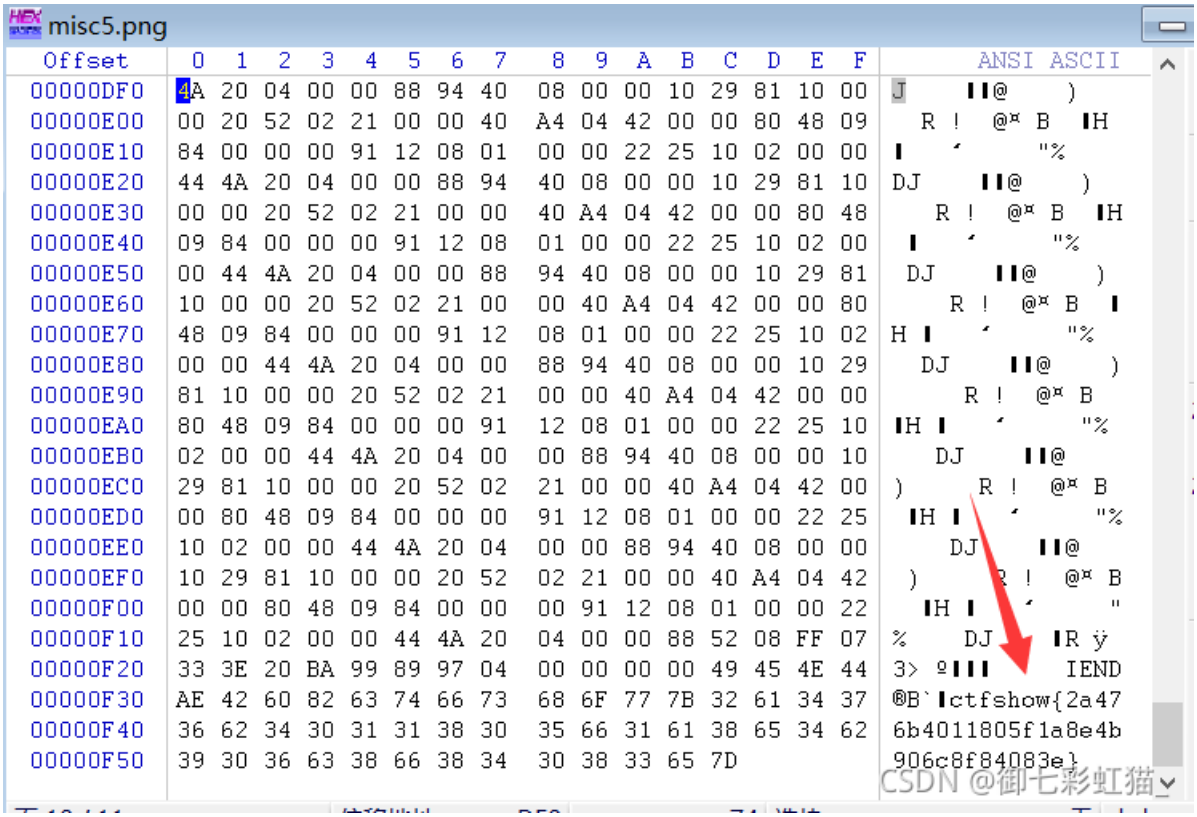
misc4

下载压缩包解压之后发现是一堆txt文件并且打开全是乱码，看样子全是图片，可以把所有文件的后缀全部改成png的后缀然后flag，最好在1-6拼接出来即可
但是看了完整WP，应该依次顺序是 png、jpg、bmp、gif、tif、webp 的后缀

图片篇(信息附加)

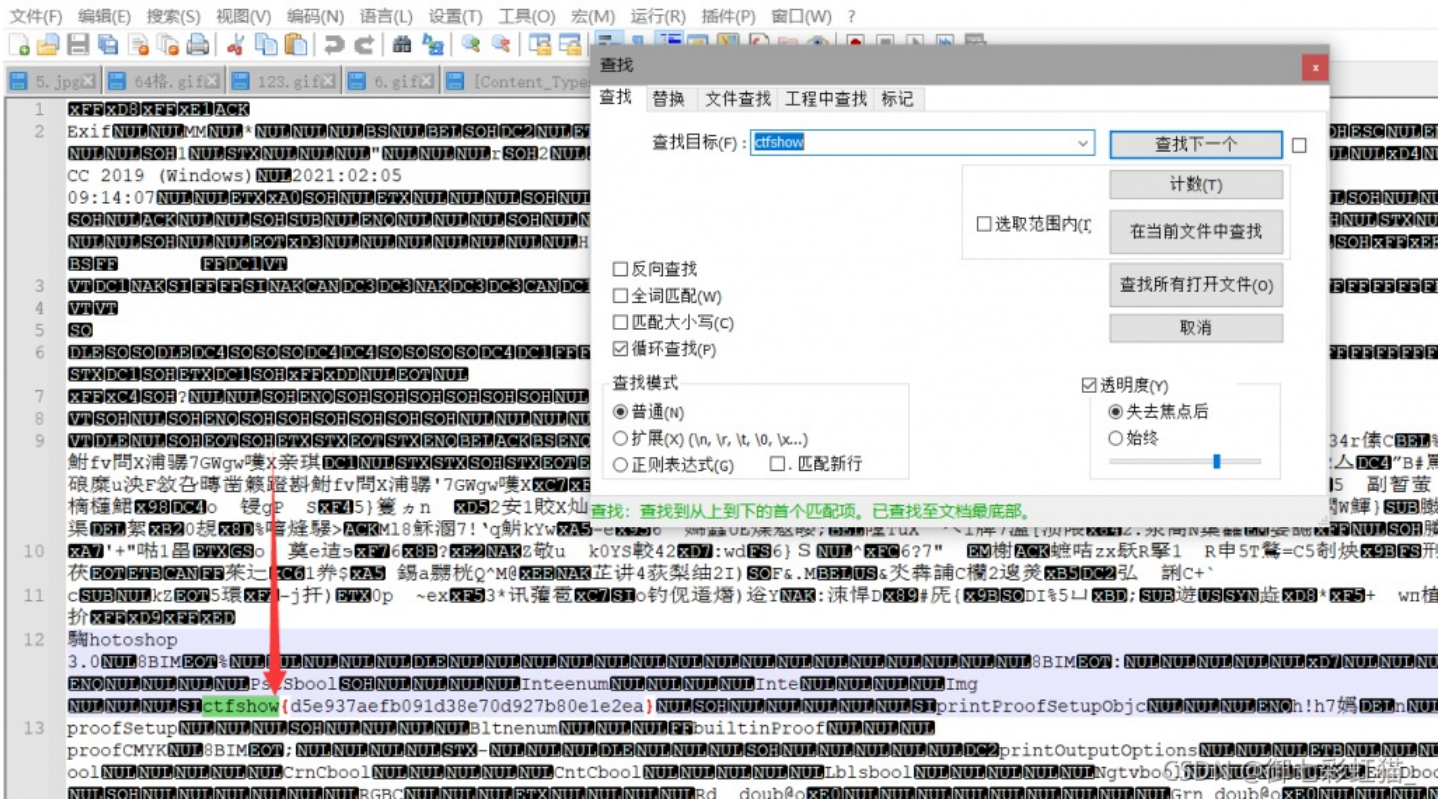
misc5

直接使用Winhex打开在图片末尾出现flag如下图所示



misc6

和上个题差不多，用txt打开或者其他文本文档打开都可以我这里是使用Notepad++打开然后ctrl+f搜索的如下图

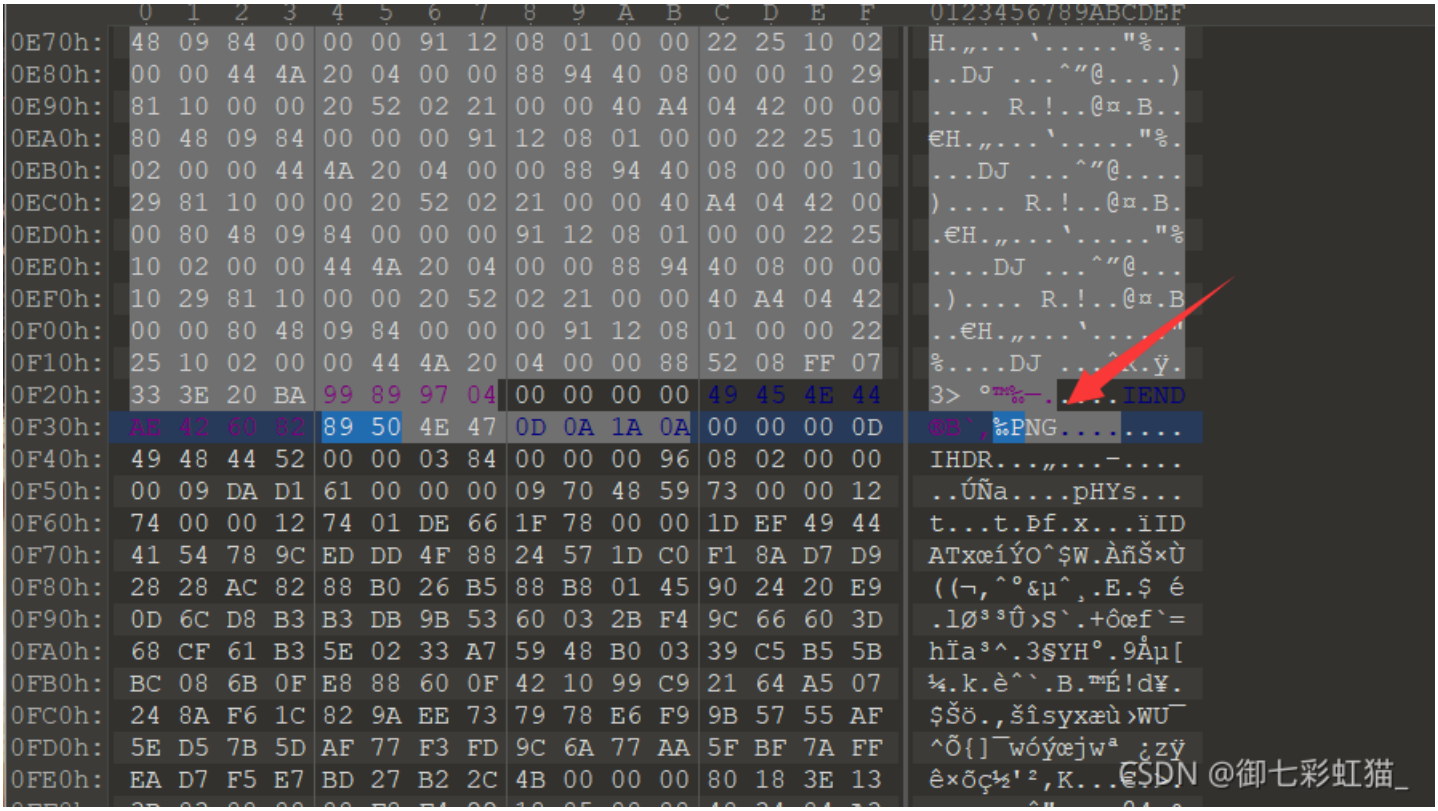


misc7

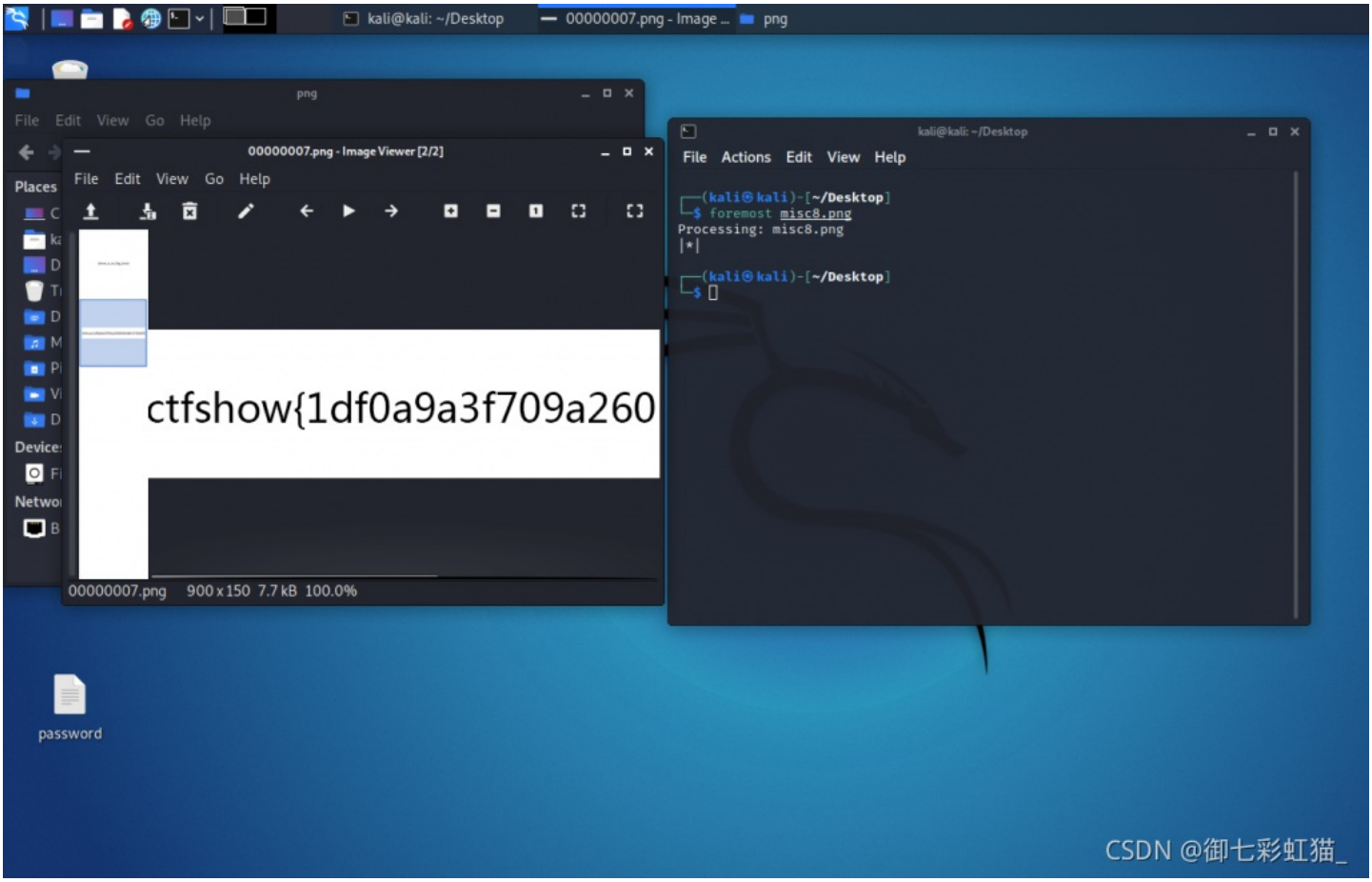
和misc6做法一样

misc8

拖入Winhex或者010edit或者其他软件，往下面发现图片里面隐藏了PNG图片，我们可以手动分离出来或者直接使用binwalk或者foremost

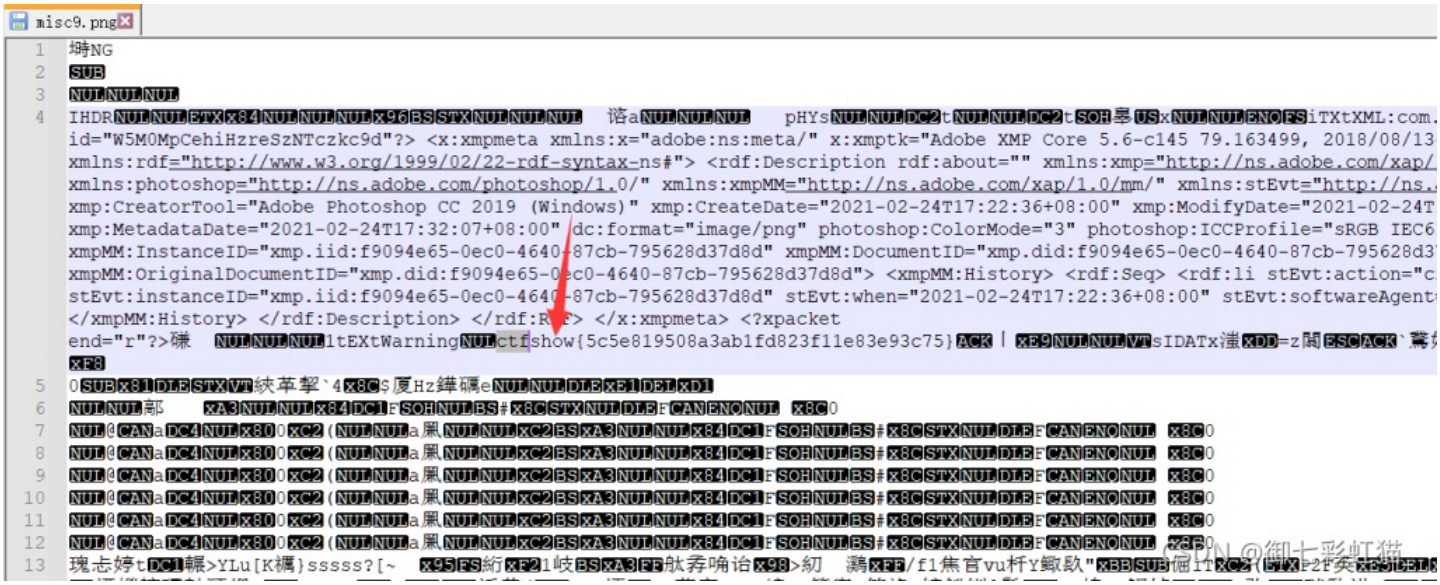


下面是使用Kali里面的foremost分离出来，然后直接打开图片就可以了



misc9

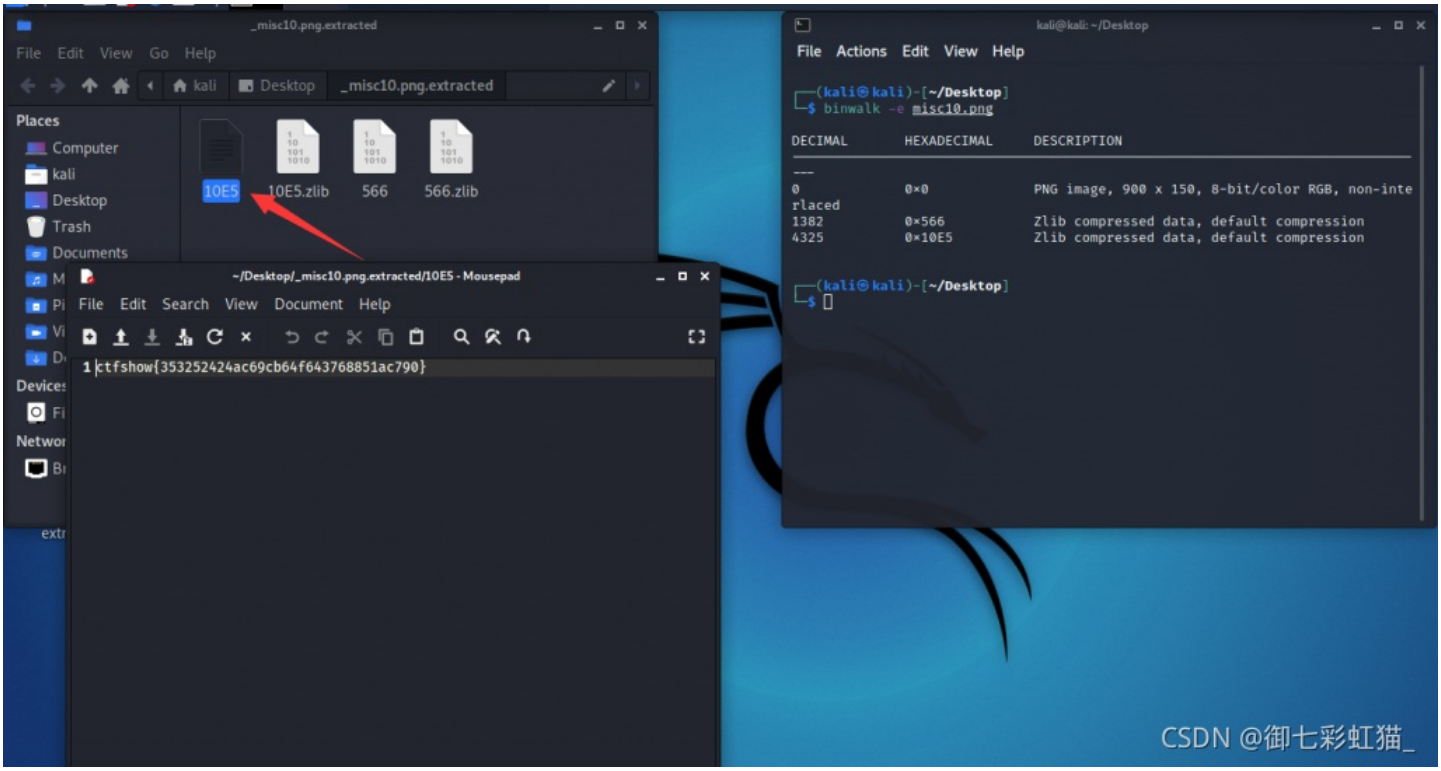
和misc6的做法一样，如下图



misc10

三板斧binwalk或者foremost如下图
命令是

> binwalk -e 文件

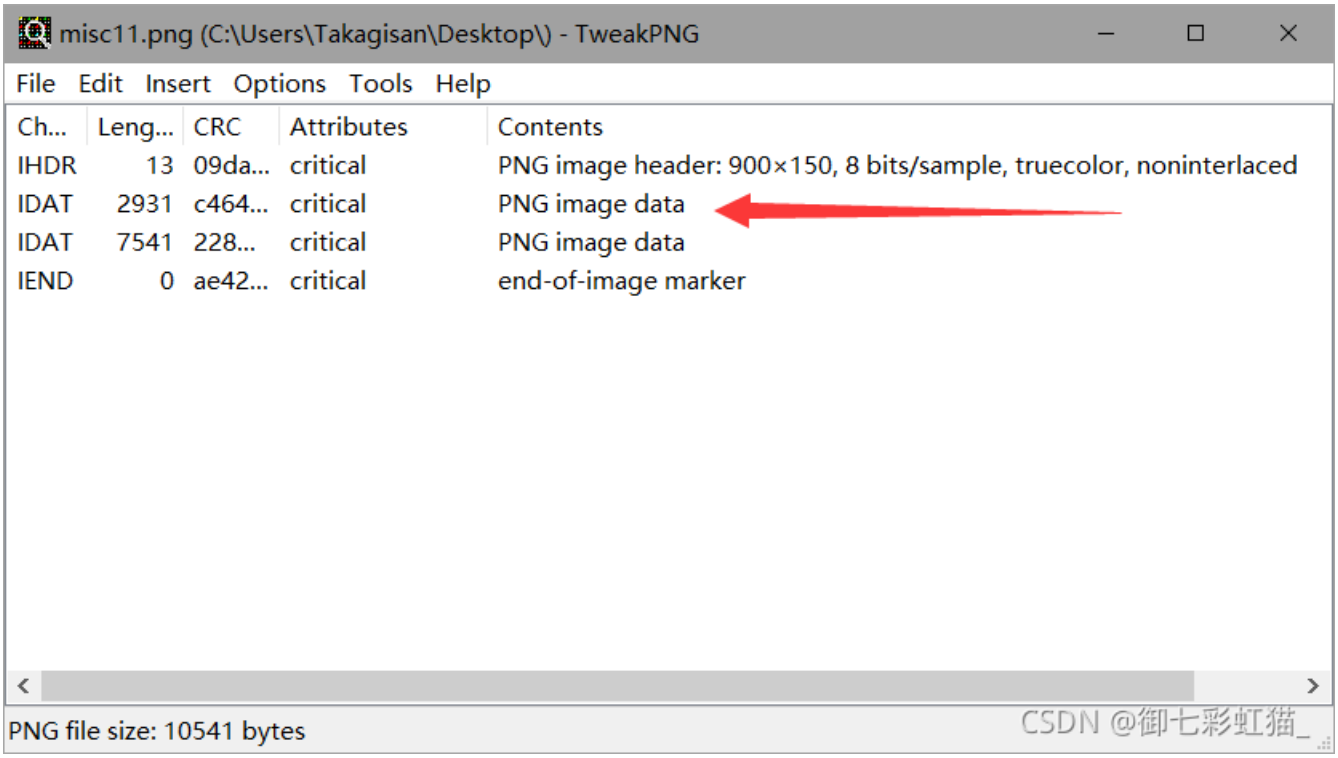


misc11

发现用了上面的办法没有一个有用的时候就要想想是否是自己知识太少了，是的是我知识太少了，然后看了WP是要使用一个叫做TweakPNG的工具，然后来删除IDAT块，然后另存为另外一个图片flag就出来了，下面这个链接是工具下载链接

工具下载链接

下载好软件之后，如下图所示把图片拖进去删除第一个IDAT数据块，然后ctrl+shift+s另存为另外一张新的png图片就出来了flag



ctfshow{44620176948fa759d3eeafeac99f1ce9}

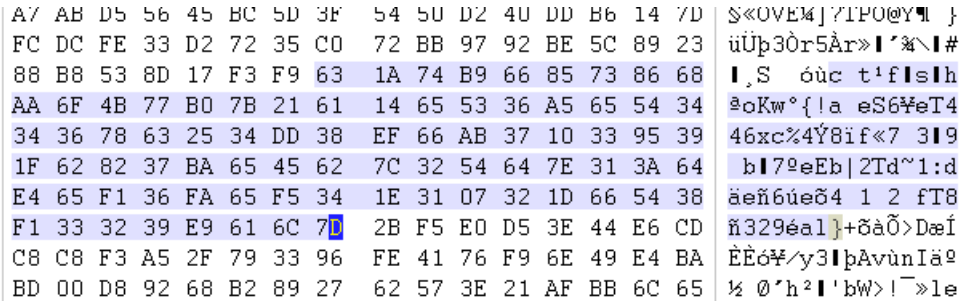
misc12

看了一眼题目提示“flag在另一张图里面"
想的是应该和上一题解题思路一样，因为按照之前的方法做不出来，然后打开了 TweakPNG
发现了，这个图片有很多很多个IDAT数据块，需要删掉前面8个IDAT块才能出来flag如下图

ctfshow{10ea26425dd4708f7da7a13c8e256a73}

misc13

题目提示是说flag在末尾,发现根本没有flag，说来也好笑，我是对flag敏感看见了一个"{"花括号我以为是flag结果看起来长得真的是flag如下图



```
A7 AB D5 56 45 BC 5D 3F  54 5U D2 4U DD B6 14 7D
FC DC FE 33 D2 72 35 C0  72 BB 97 92 BE 5C 89 23
88 B8 53 8D 17 F3 F9 63  1A 74 B9 66 85 73 86 68
AA 6F 4B 77 B0 7B 21 61  14 65 53 36 A5 65 54 34
34 36 78 63 25 34 DD 38  EF 66 AB 37 10 33 95 39
1F 62 82 37 BA 65 45 62  7C 32 54 64 7E 31 3A 64
E4 65 F1 36 FA 65 F5 34  1E 31 07 32 1D 66 54 38
F1 33 32 39 E9 61 6C 7D  2B F5 E0 D5 3E 44 E6 CD
C8 C8 F3 A5 2F 79 33 96  FE 41 76 F9 6E 49 E4 BA
BD 00 D8 92 68 B2 89 27  62 57 3E 21 AF BB 6C 65
```

仔细观察了一下好像是有那么ctfshow的字眼，最后是发现的是隔了一位字符每隔一位选一个字符就是ctfshow，然后写了python代码，代码如下

```
s = "631A74B96685738668AA6F4B77B07B216114655336A56554333465786125340D38EF66AB35103195381F628237BA6545347C3254647E373A64E465F136FA66F5341E3107321D665438F1333239E9616C7D"
flag = ""
for  i in range(0, len(s), 4):
    flag += s[i]
    flag += s[i + 1]
print(flag)
```

个人解释一下代码，有错请指正
flag=""定义了一个空的字典

一个循环从0开始到len(s)的长度，len()是获取的是s的长度信息，step是4

在16进制里面两个字节等于的是一个字符比如"63"等于的是c，所以下面有两个flag+=

第一个flag += 是获取的是第一位的信息，第二个就是第二位

比如第一个flag += s[i]，第一次循环，i下标是6然后i+1下标是1然后就是3就获取到了第一位字符是63也就是c

step是4刚好跳过了63和74中间的两个字符1A，所以第二次循环获取到的是74，以此内推

最终跑出来是:**63746673686F777B6165366533656134386635313862376534326437646536663431326638333961 7D**

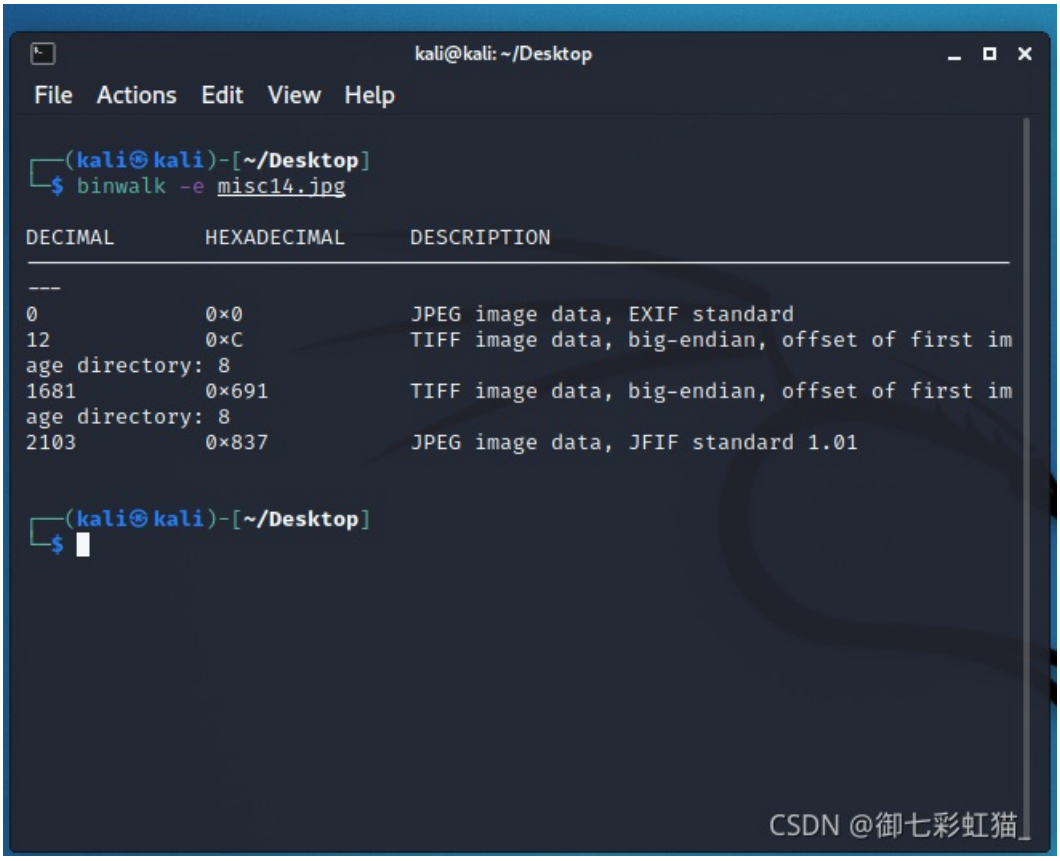
然后HEX解码最终结果是

> ctfshow{ae6e3ea48f518b7e42d7de6f412f839a}

misc14

看见题目提示是"在那张图里面"，想了一下应该不是IDAT了，应该是binwalk所以用binwalk或者foremost

但是呢，发现用binwalk -e 不行用foremost还是不行但是binwalk检测就是有文件如下图

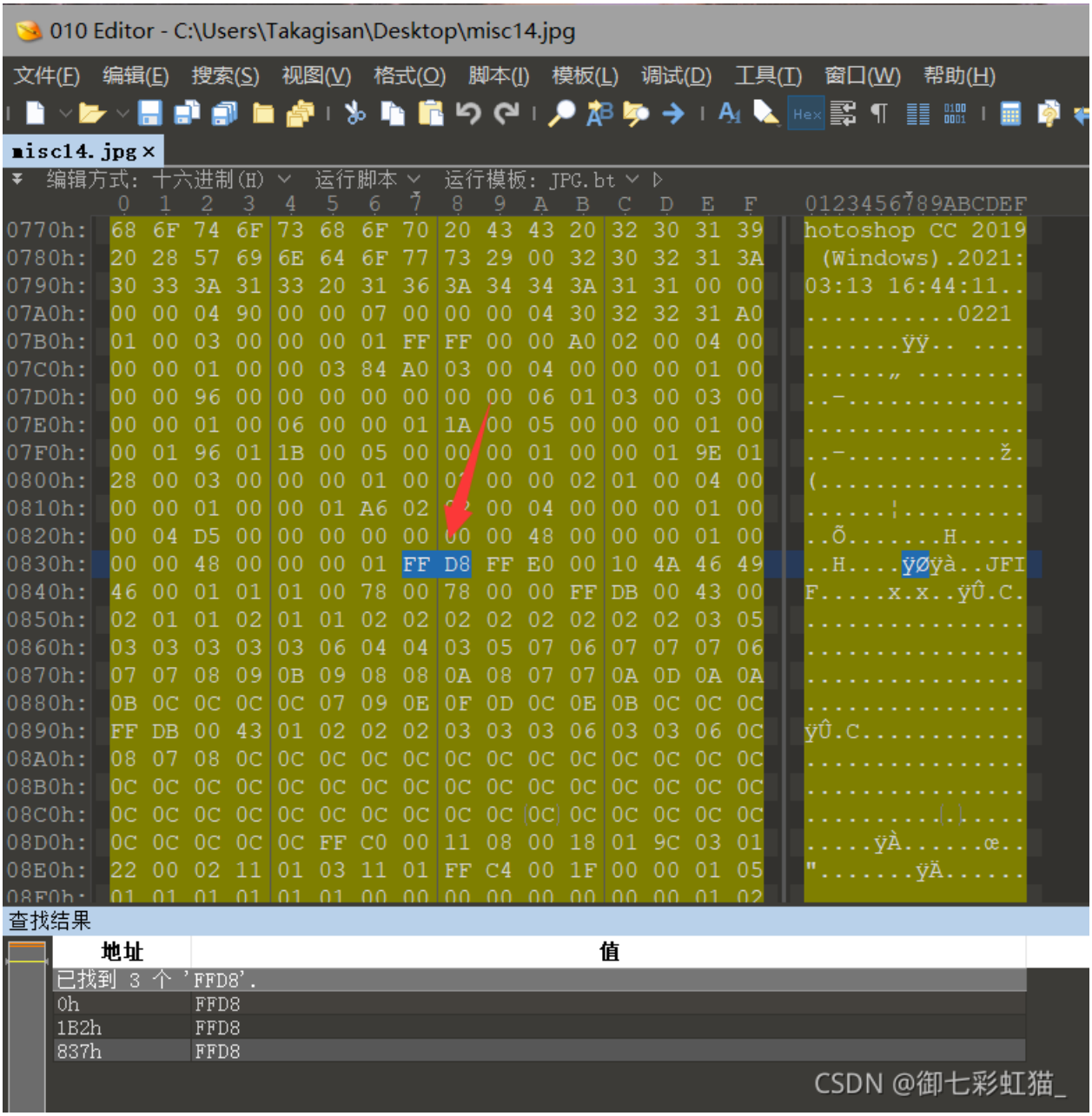


看样子应该是另外一张JPG图片了，手扣试试，JPG图头文件一般是FFD8

这里我使用的工具是010edit如下图所示，直接ctrl+f搜索16进制FFD8，看了一下

其实是有三个FFD8头的第一个头肯定是原本图片的头，第二个FFD8和右边的字符对比了一下没有JFIF开头盲猜不是

第三个就有了，直接复制FFD8到最后FFD9保存为另外一张jpg图片



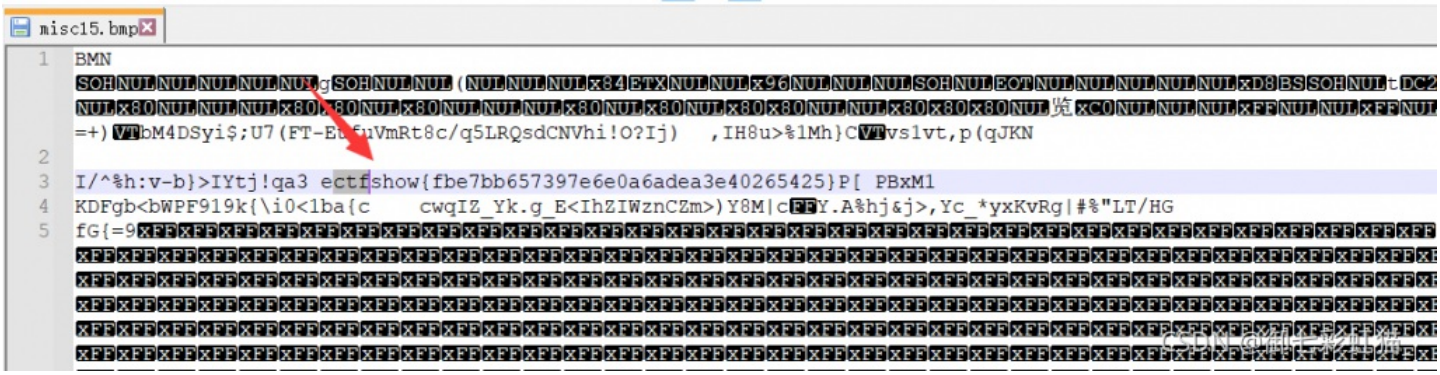
最终flag如下图

```
ctfshow{ce520f767fc465b0787cdb936363e694}
```

misc15

直接用winhex打开或者其他查看文本打开就能发现flag

我以为越到后面越难，想不到居然这么简单，绝了！

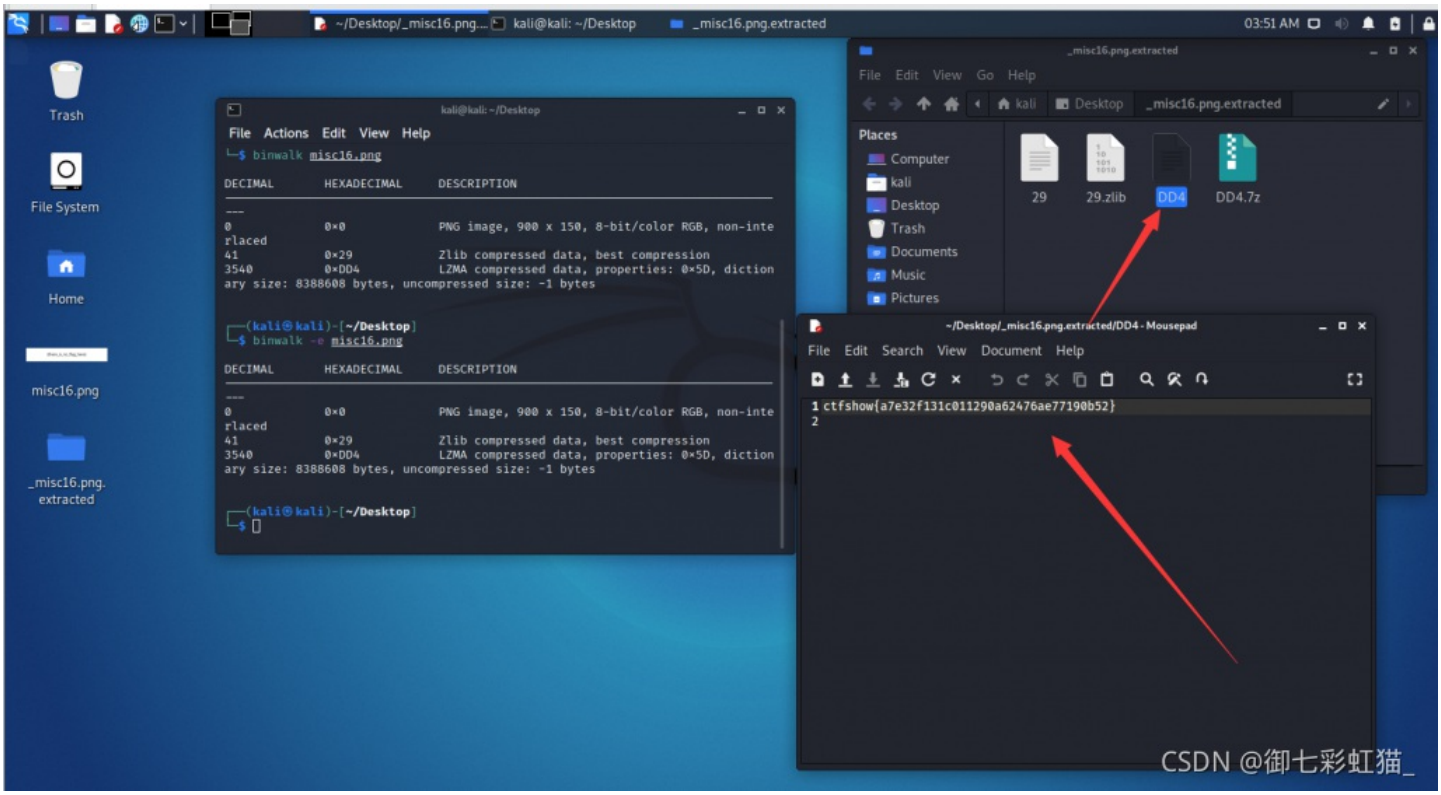


misc16

用Winhex打开没啥可发现的东西，用binwalk发现了东西参数`binwalk -e 文件`

分离出文件，在DD4里面，简单

flag如下图

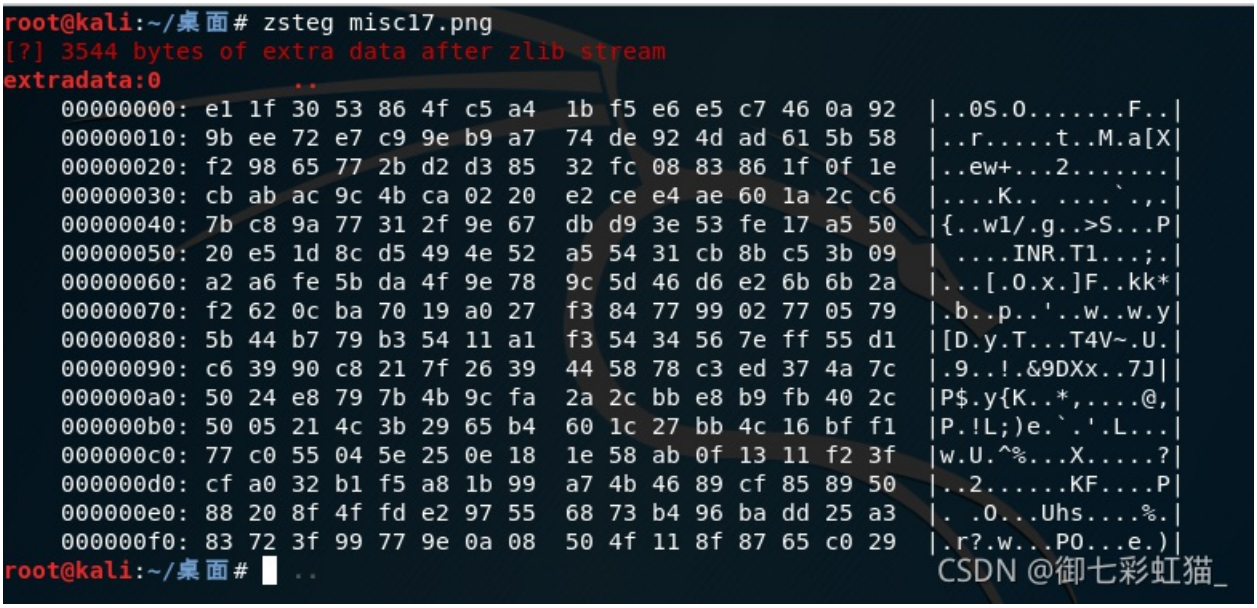


misc17

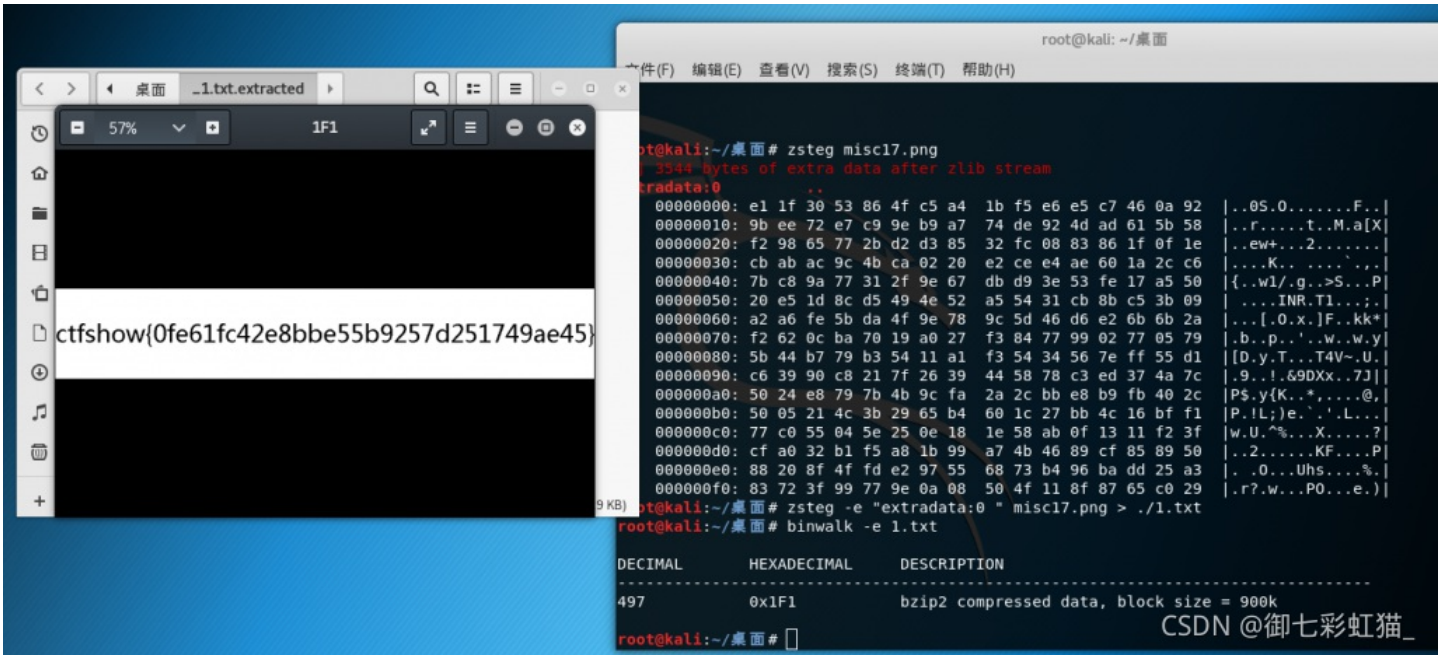
用winhex然后binwalk查看了没啥发现的，然后用到了一个新东西叫做zsteg

简单用法就是`zsteg 文件`

基本上能发现隐藏数据如下图，发现了隐藏数据在extradata:0



然后使用`zsteg -e " extradata:0 " misc17.png > ./1.txt`
就生成了一个带有数据的txt文件最后用binwalk -e分离出来即可,拿到flag



misc18

看题目提示的是 flag在标题、作者、照相机和镜头型号里

这里用到了一个新命令叫做exiftool，用来查看图片的各种信息，比如拍摄地点，什么软件，时间，作者，等等之类的

这个工具不是kali自带的工具，先要下载

输入命令:`apt-get install exiftool`

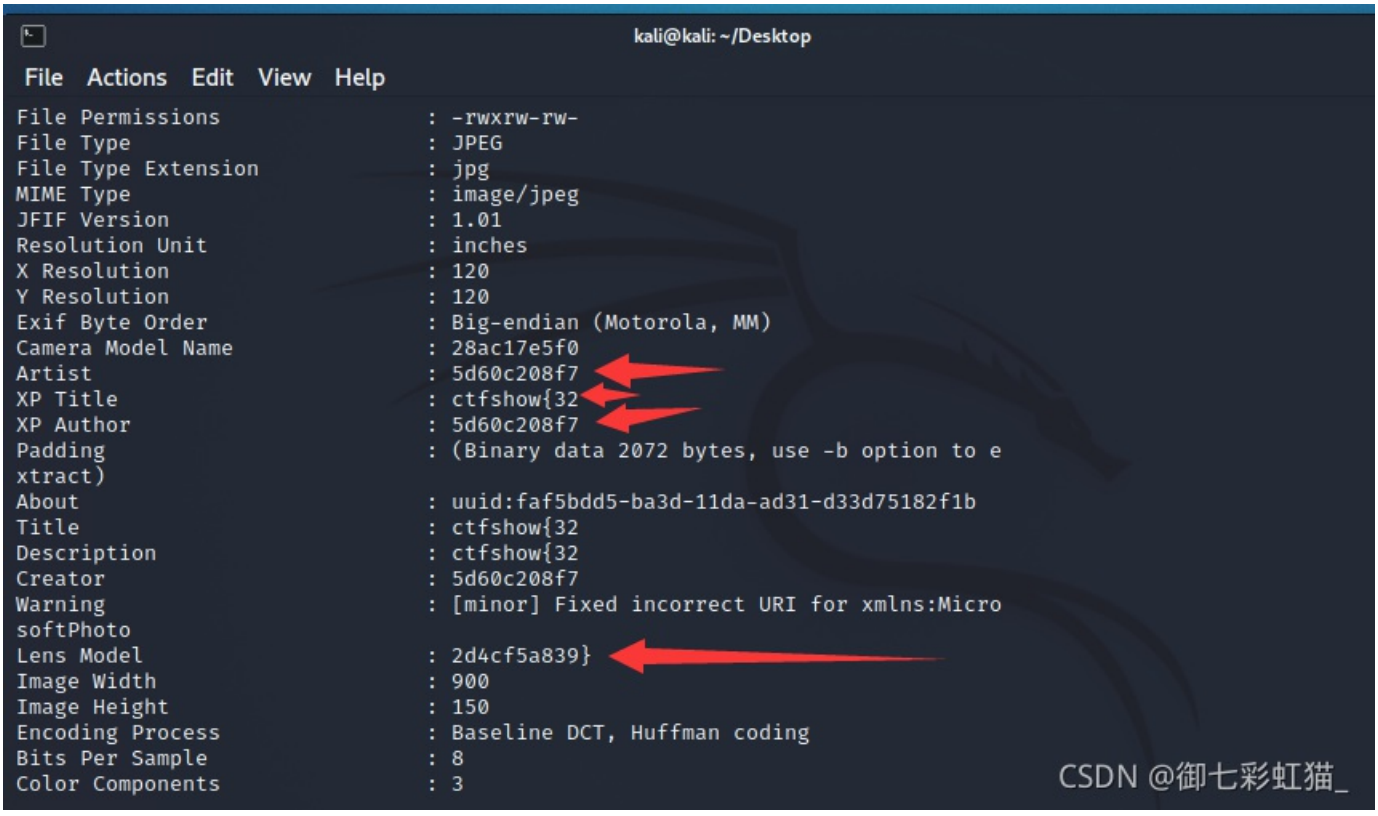
如果下载失败报错是没有依赖，如下图

```
root@kali:~/桌面# apt-get install exiftool
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
注意，选中 'libimage-exiftool-perl' 而非 'exiftool'
您也许需要运行“apt --fix-broken install”来修正上面的错误。
下列软件包有未满足的依赖关系：
fcitx-baidupinyin : 依赖：fcitx-bin 但是它将不会被安装
                    依赖：fcitx-data (>= 1:4.2.0) 但是它将不会被安装
                    依赖：fcitx-modules 但是它将不会被安装
                    依赖：qml-module-qtquick-controls (>= 5.5.1) 但是它将不会被安装
                    推荐：fcitx (>= 1:4.2.0) 但是它将不会被安装
E: 有未能满足的依赖关系。请尝试不指明软件包的名字来运行“apt --fix-broken install”(也可以指定一
法)。
root@kali:~/桌面#
```


输入: `sudo apt install libimage-exiftool-perl`

然后在输入:`apt-get install exiftool`一般能OK

直接使用exiftool 文件就能看见一堆信息，找到有价值的信息如下图



按照 标题、作者、照相机和镜头 排序
最终flag如下

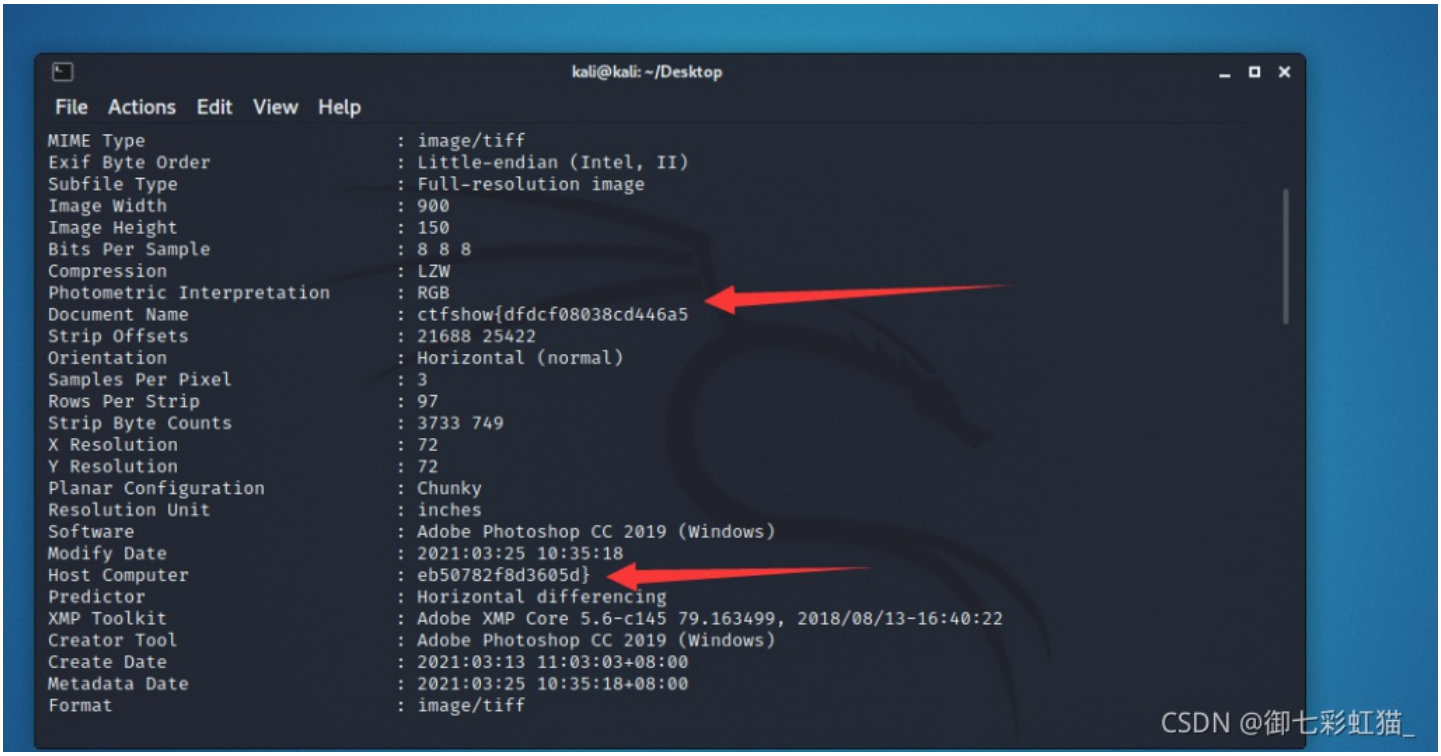
```
ctfshow{325d60c208f728ac17e5f02d4cf5a839}
```

misc19

题目提示的是 flag在主机上的文档名里

和上题一样使用exiftool 文件名

如下图所示



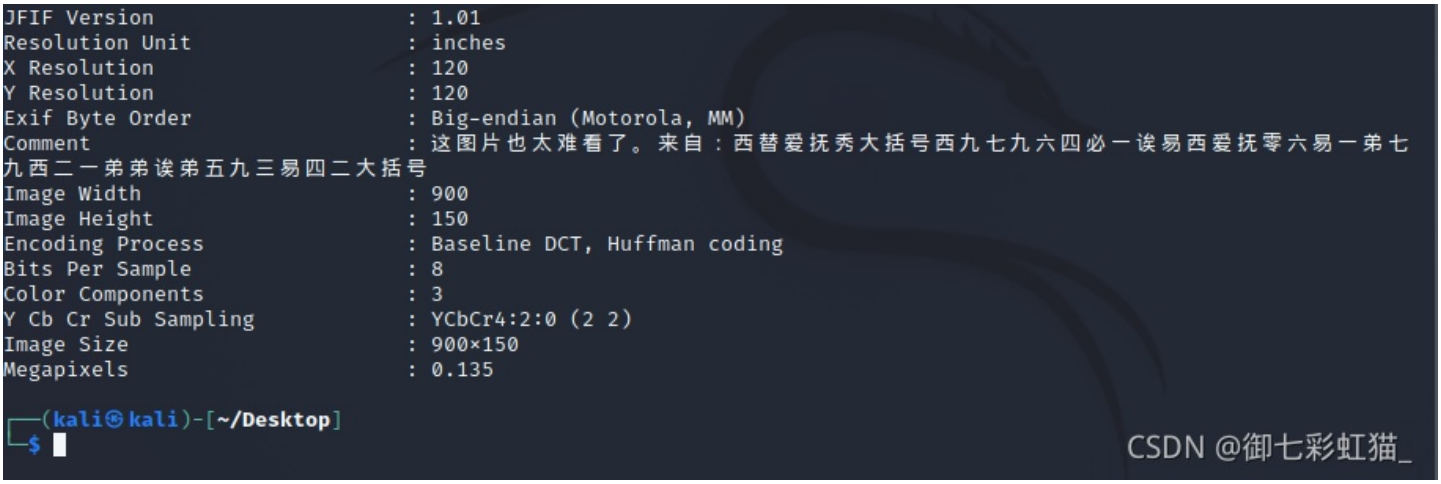
最终flag是如下

```
ctfshow{dfdcf08038cd446a5eb50782f8d3605d}
```

misc20

解题思路和misc19和18差不多

提示是flag在评论里面，我们发现如下图



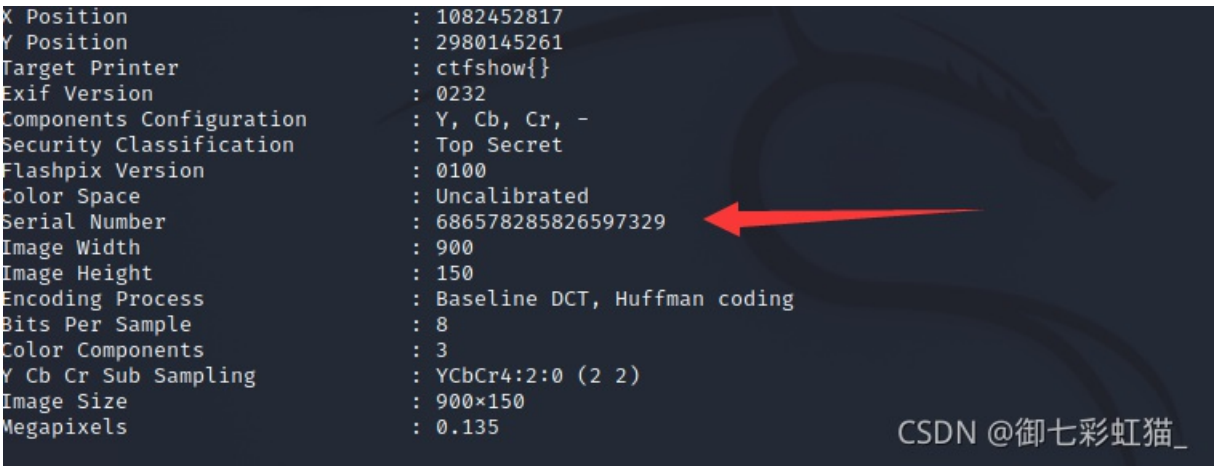
发现是谐音， 最终flag是

```
ctfshow{c97964b1aecf06e1d79c21ddad593e42}
```

misc21

提示的是 flag在序号里 ， 应该和其他题是一样的，如下图

但是感觉和正常的flag不一样，看着像HEX编码，我们转字符试试

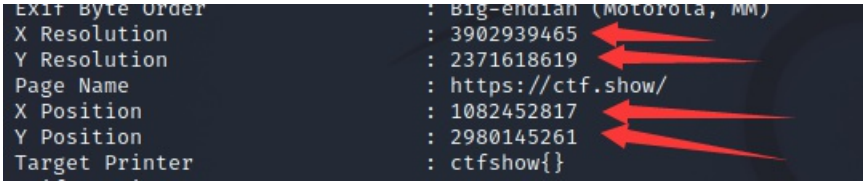


在线转换网站是□ 在线16进制转字符

转换之后如下图

提示是hex(x&y)意思是转码x和y

仔细找了找发现，发现有四个符合要求了的



在Python里面有个hex()函数可以自动将16进制转换成字符串

如下这个代码很简单，因为hex输出前面会自动输出0x

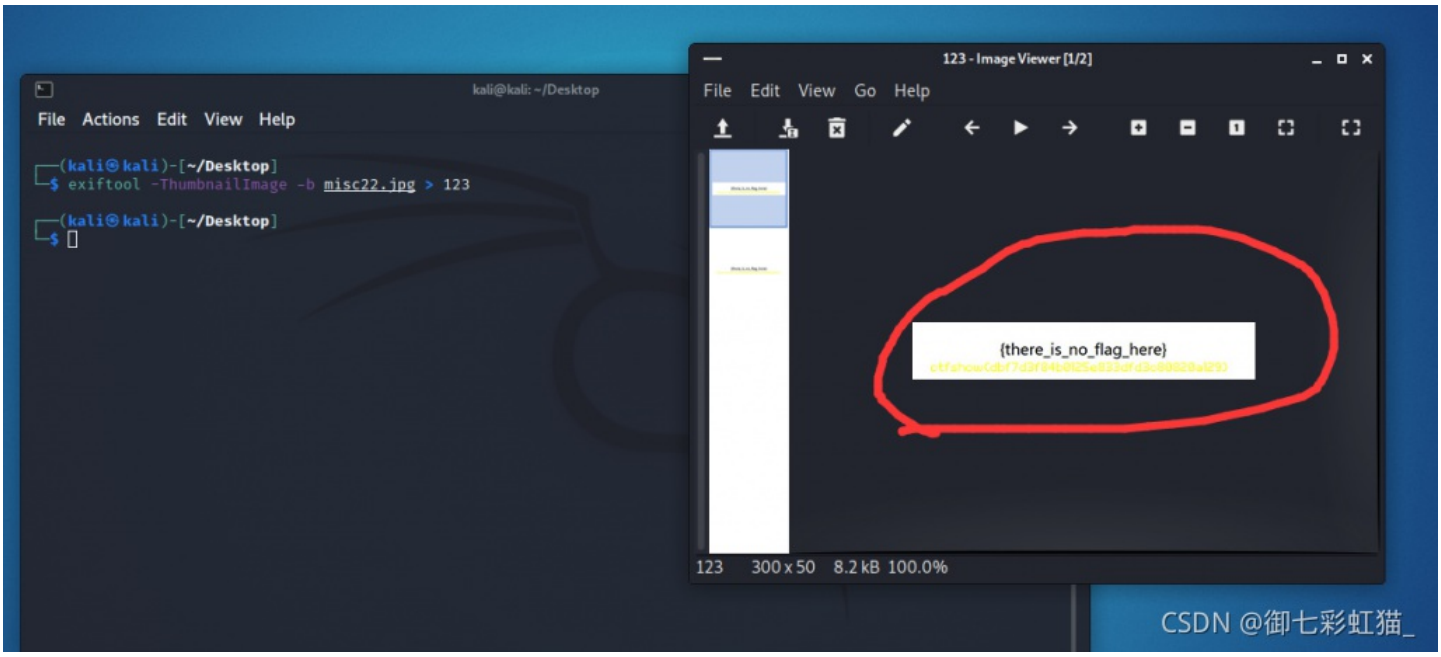
[2:]的作用是删除前面的0x,“+”是把字符连接起来

```
print('ctf{'+hex(3902939465)[2:]+hex(2371618619)[2:]+hex(1082452817)[2:]+hex(2980145261)[2:]+'}')
```

misc22

其他方法都不出来看了WP发现这题是 thumbnail 隐写

命令是:`exiftool -ThumbnailImage -b misc22.png > 123.txt`



我眼睛都快看瞎了这个黄色真的难看，然后最终FLAG是

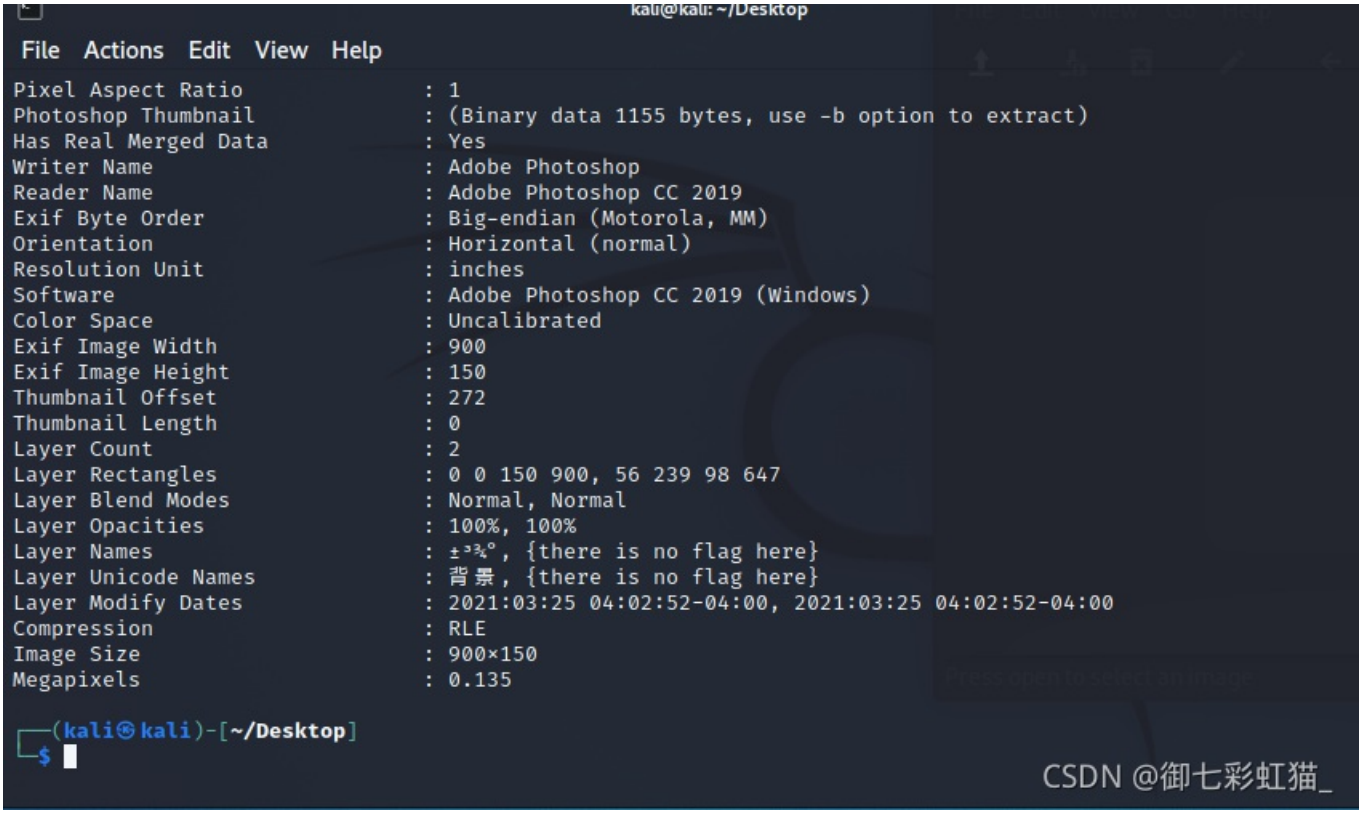
```
ctfshow{dbf7d3f84b0125e833dfd3o80820a129
```

misc23

下载下来文件发现是PS的.psd文件

Winhex和Binwalk都没啥用，直接用exiftool查看了一下如下图

东西太多了，懒得看



使用命令如下,查找里面是否有ctfshow的内容

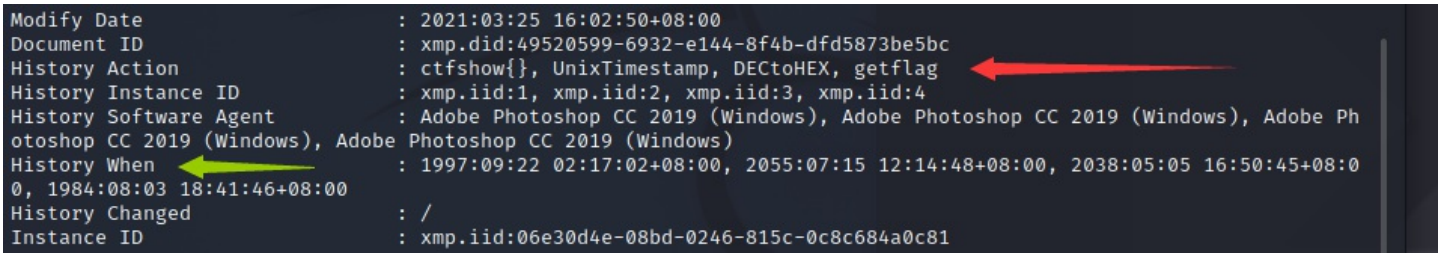
```
exiftool misc23.psd | grep ctfshow
```

还真有

显示是History Action这行，于是我找了一下找到了，然后还发现了一句话，如下图

红色箭头的那句话的意思是说，转换时间戳，然后来获取flag

绿色箭头是要转换的时间把这些转换成时间戳



在线时间戳转换网站□时间戳转换

转换过后最终得到的是

874865822 | 2699237688 | 2156662245 | 460377706

然后根据上面的经验把这些数字转换成字符串用Python的hex()函数，代码如下

```
print('ctf{'+hex(874865822)[2:]+hex(2699237688)[2:]+hex(2156662245)[2:]+hex(460377706)[2:]+'}')
```

然后最终的flag如下

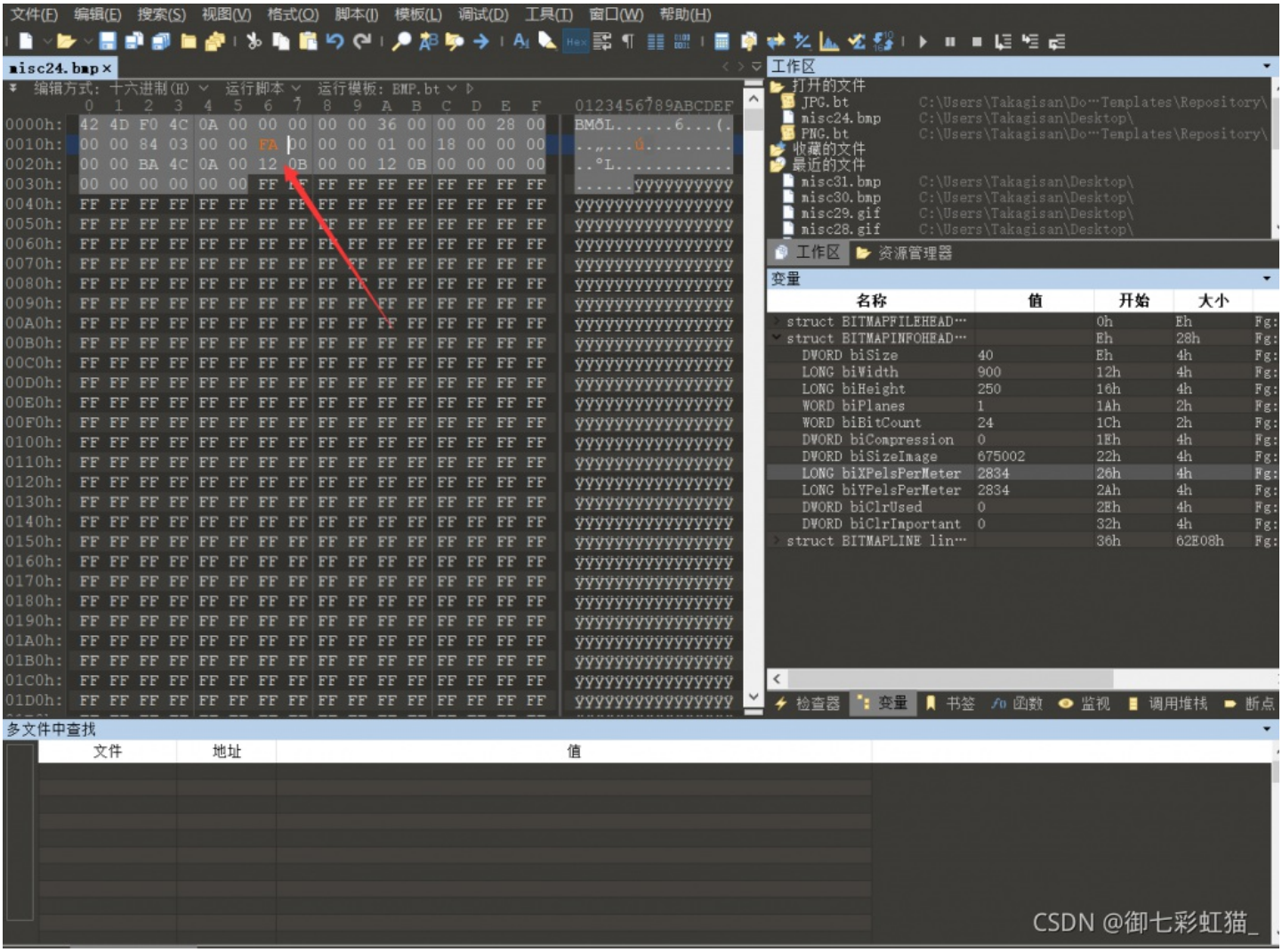
```
ctf{3425649ea0e31938808c0de51b70ce6a}
```

misc24

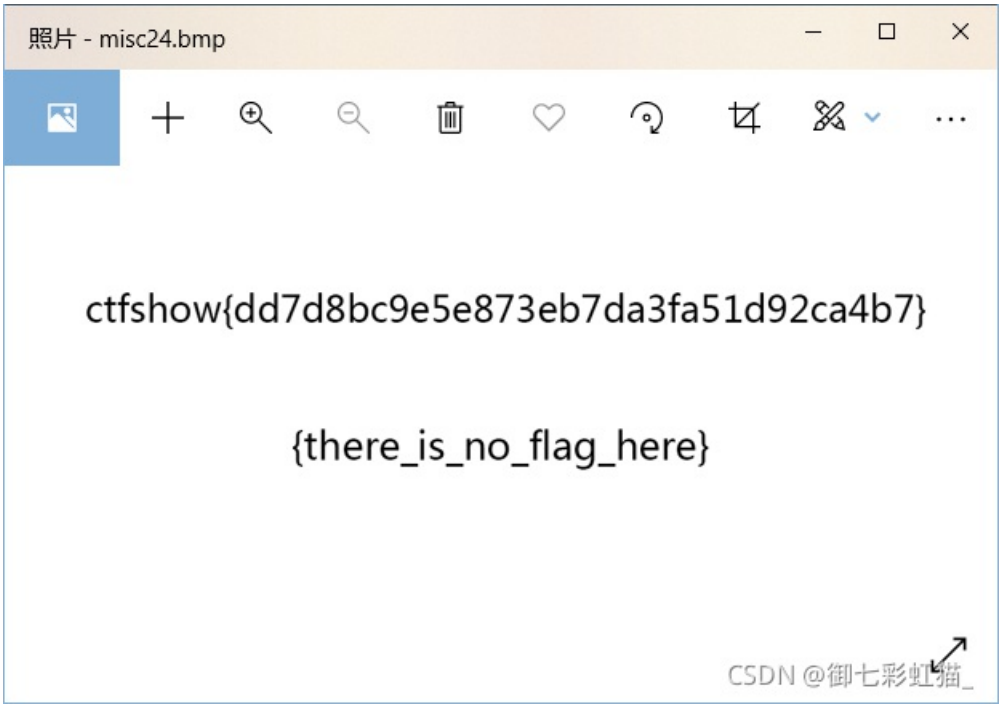
提示是:flag在图片上面

使用010edit来修改高度

可以使用python的hex函数来转换FA对应的是250的数字



最终flag: `ctfshow{dd7d8bc9e5e873eb7da3fa51d92ca4b7}`

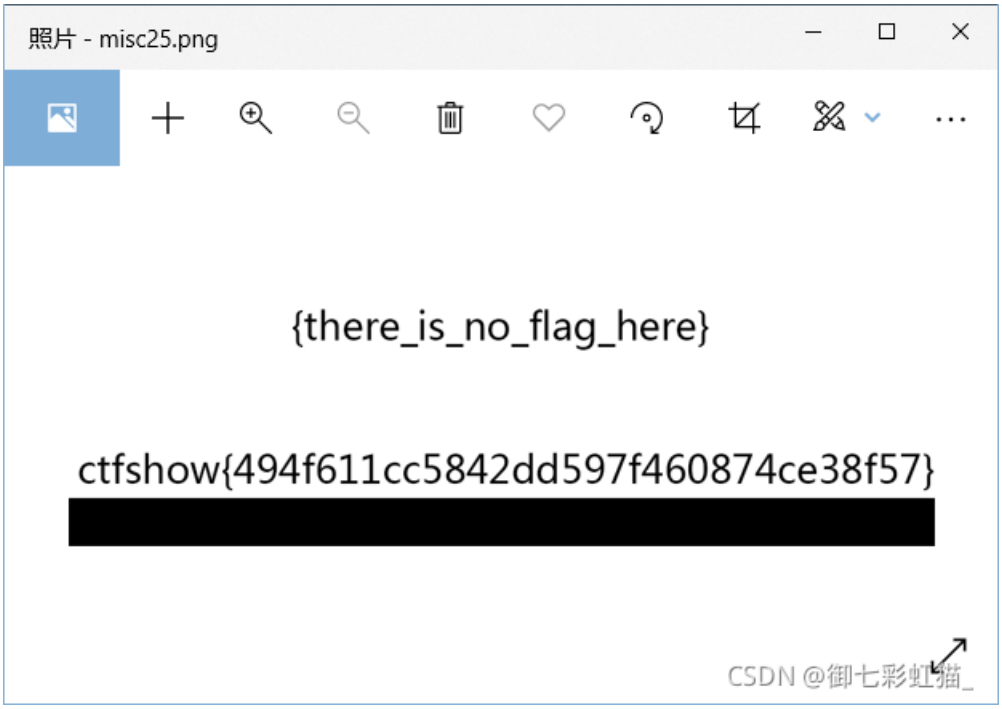


misc25

同Misc24

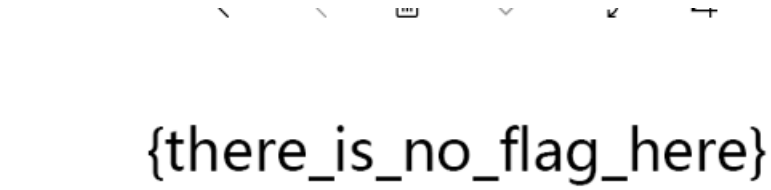
如下图最终flag

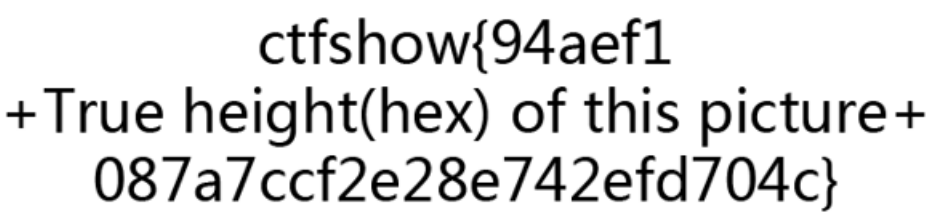
```
ctfshow{494f611cc5842dd597f460874ce38f57}
```



misc26

根据提示，就无脑把高度拉高就对了，但是要真正的高度





真正的高度需要我们跑脚本

真正的高度是606，16进制是25e
代码如下

```
import binascii
import struct

crcbp = open("misc26.png", "rb").read()
for i in range(2000):
    for j in range(2000):
        data = crcbp[12:16] + struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0xEC9CCBC6):
            print(i, j)
            print('hex:', hex(i), hex(j))
```

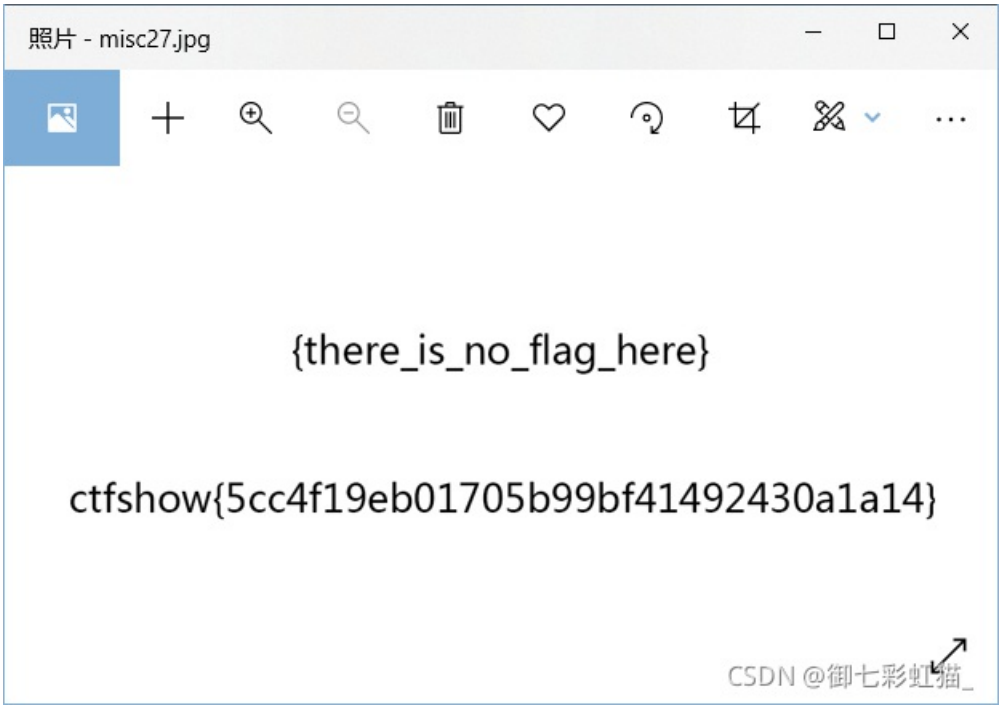
最终flag是:`ctfshow{94aef125e087a7ccf2e28e742efd704c}`

misc27

和misc24一样

最终flag为

```
ctfshow{5cc4f19eb01705b99bf41492430a1a14}
```

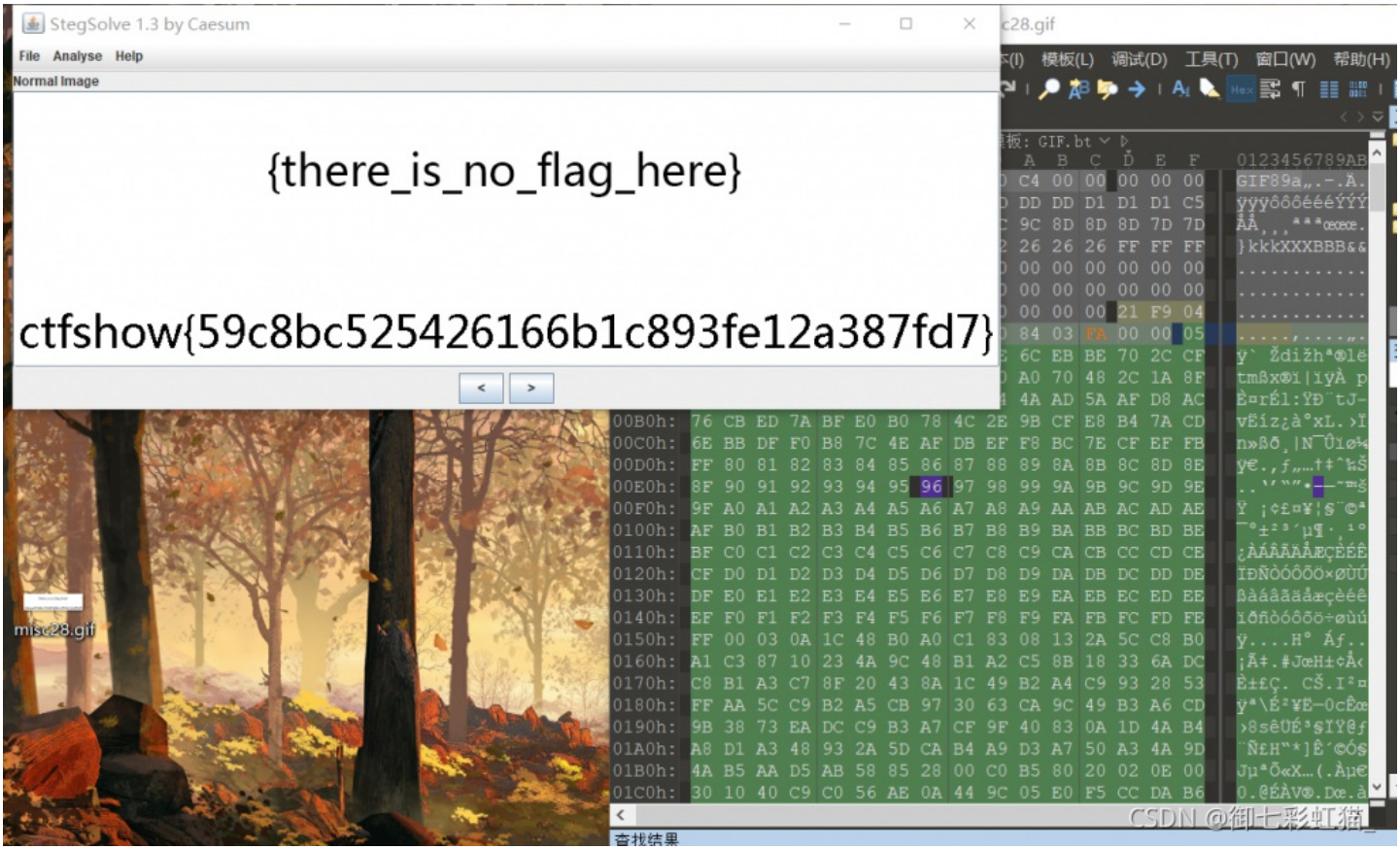


misc28

和上面一样,可能不一样的就是需要用预览图打开或者StegSolve打开

最终flag是

```
ctfshow{59c8bc525426166b1c893fe12a387fd7}
```

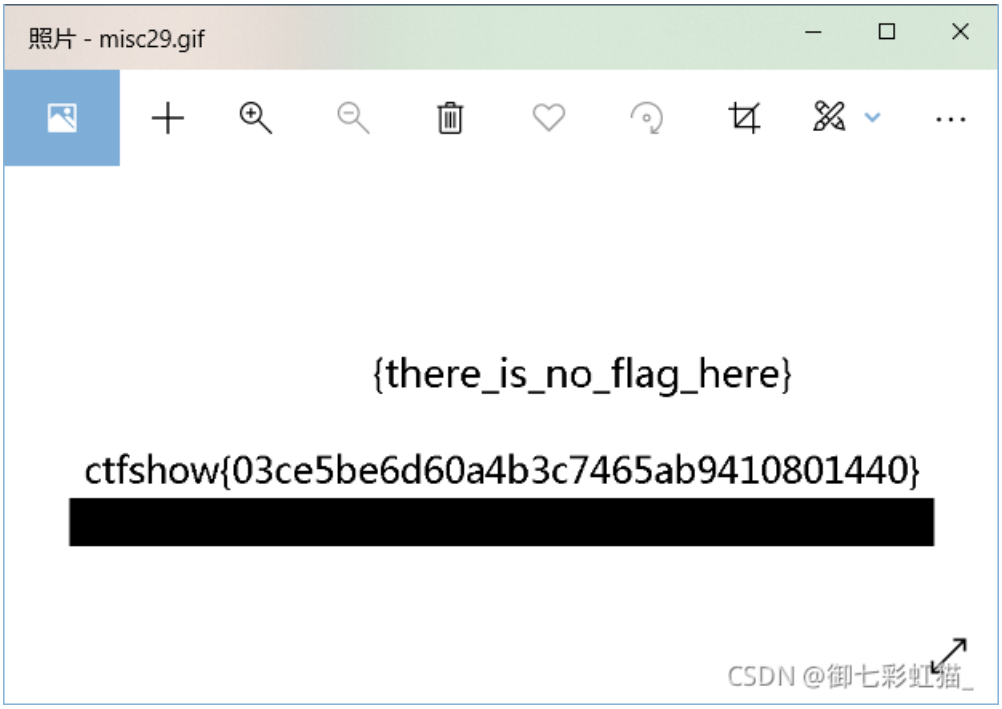


misc29

是个GIF，和上面一样修改高度

只不过我是把每一帧的高度都修改了,因为一帧一帧改了又去看太麻烦了

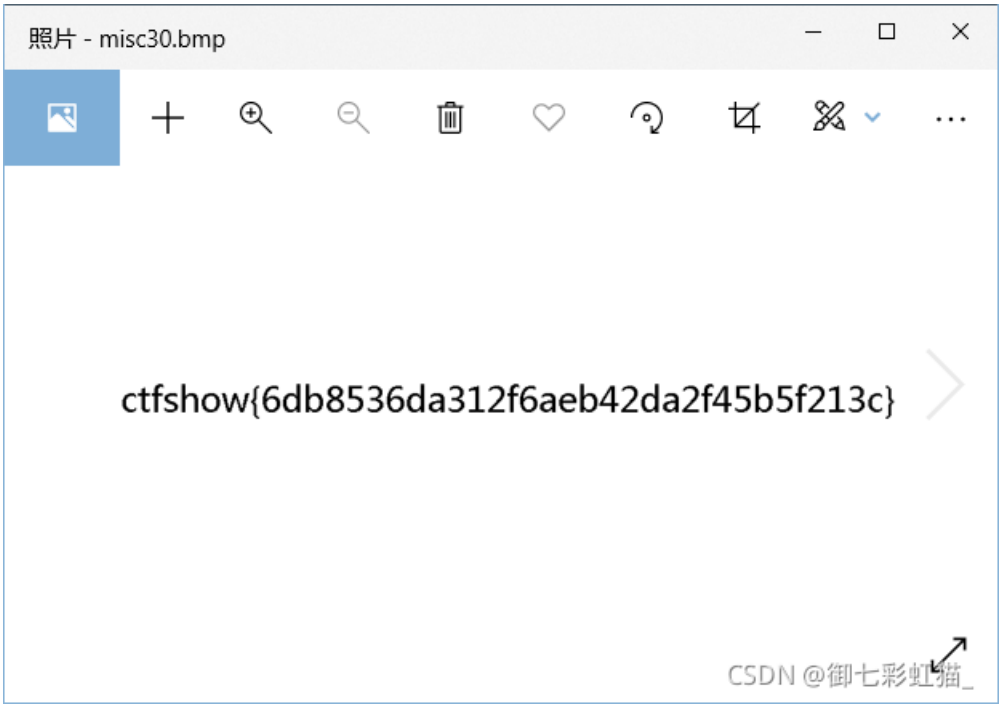
得到flag: `ctfshow{03ce5be6d60a4b3c7465ab9410801440}`



misc30

根据提示把宽度改成950就可以了,还是用010edit软件

得到flag: `ctfshow{6db8536da312f6aeb42da2f45b5f213c}`

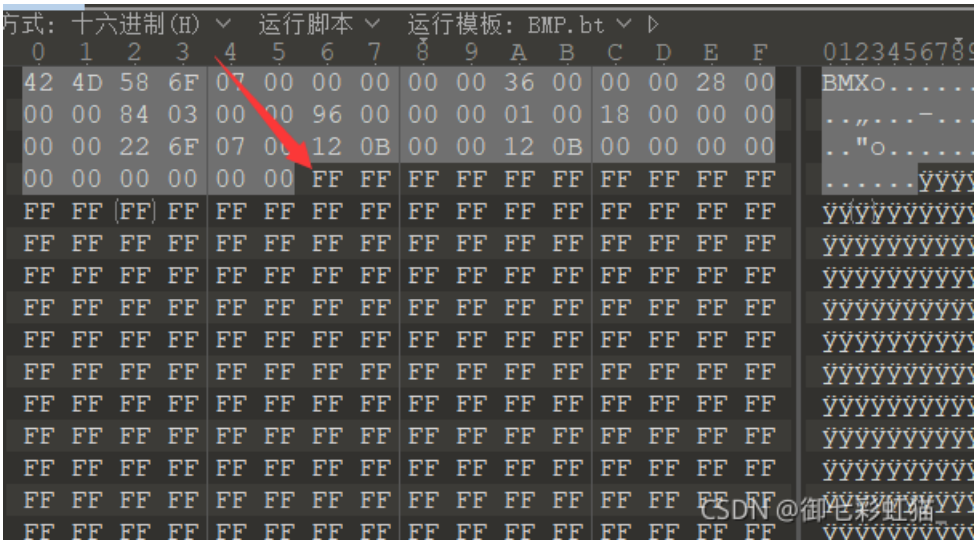


misc31

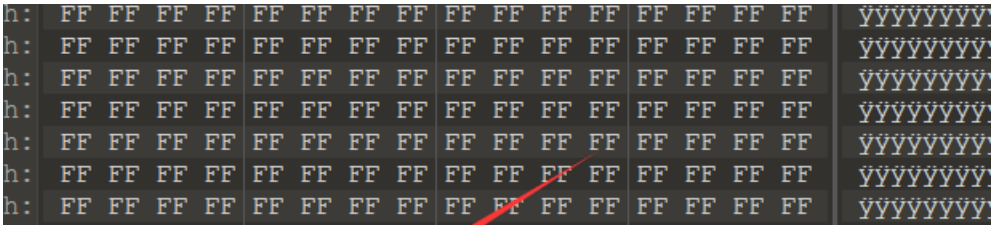
这种需要计算出高度，先要了解bmp文件格式

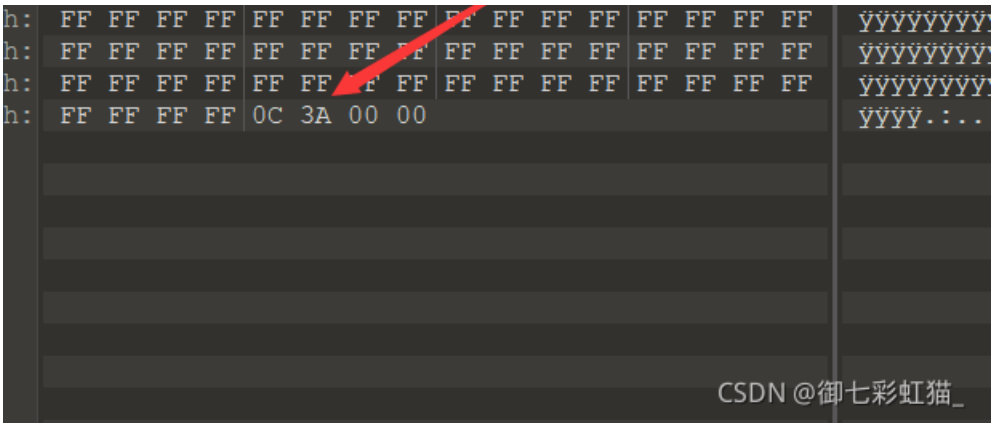
参考文章

图片放入010edit，从第一个ff开始

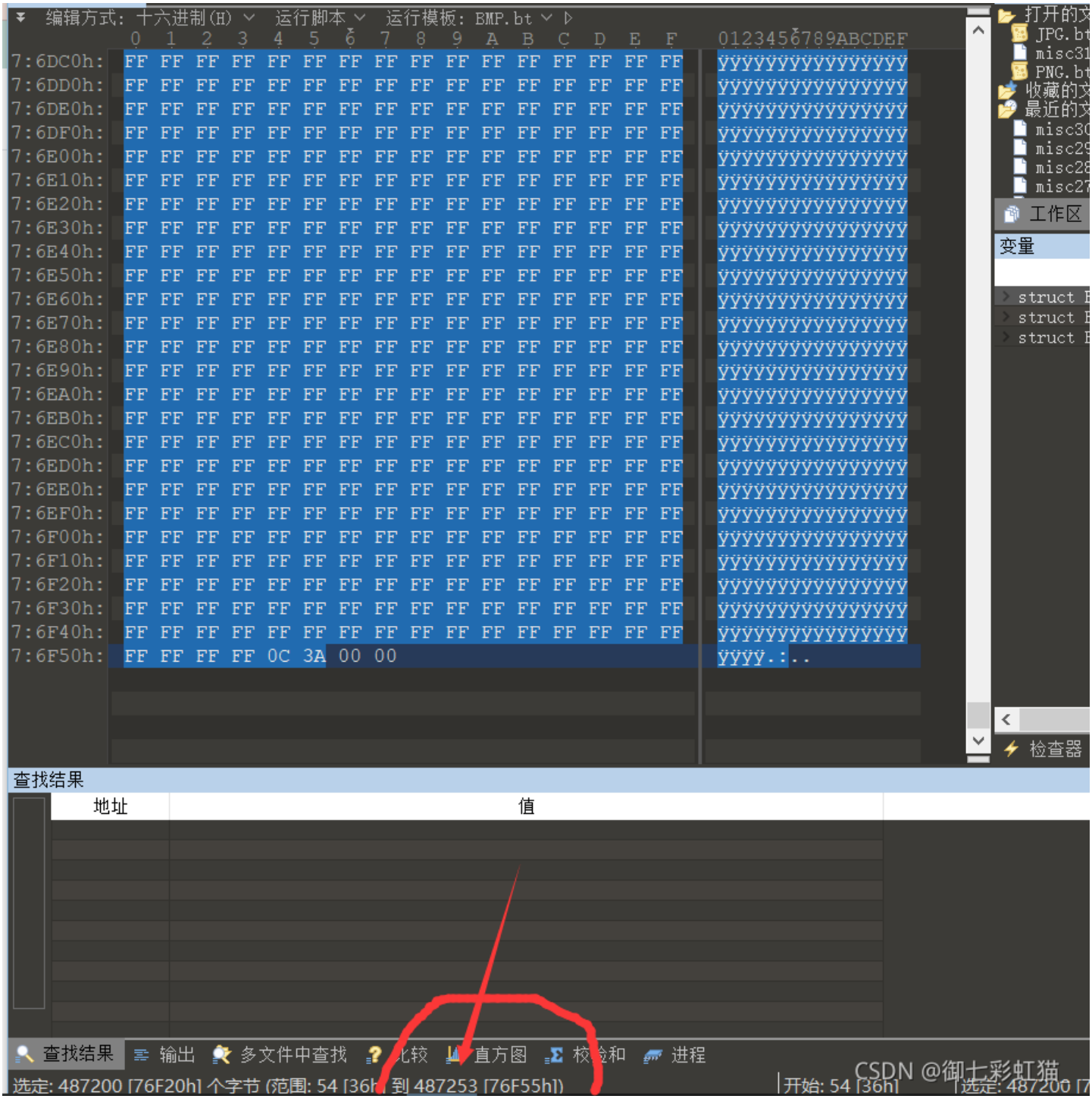


然后一直复制到最后除了00 00不要复制，因为那个是用来补位的

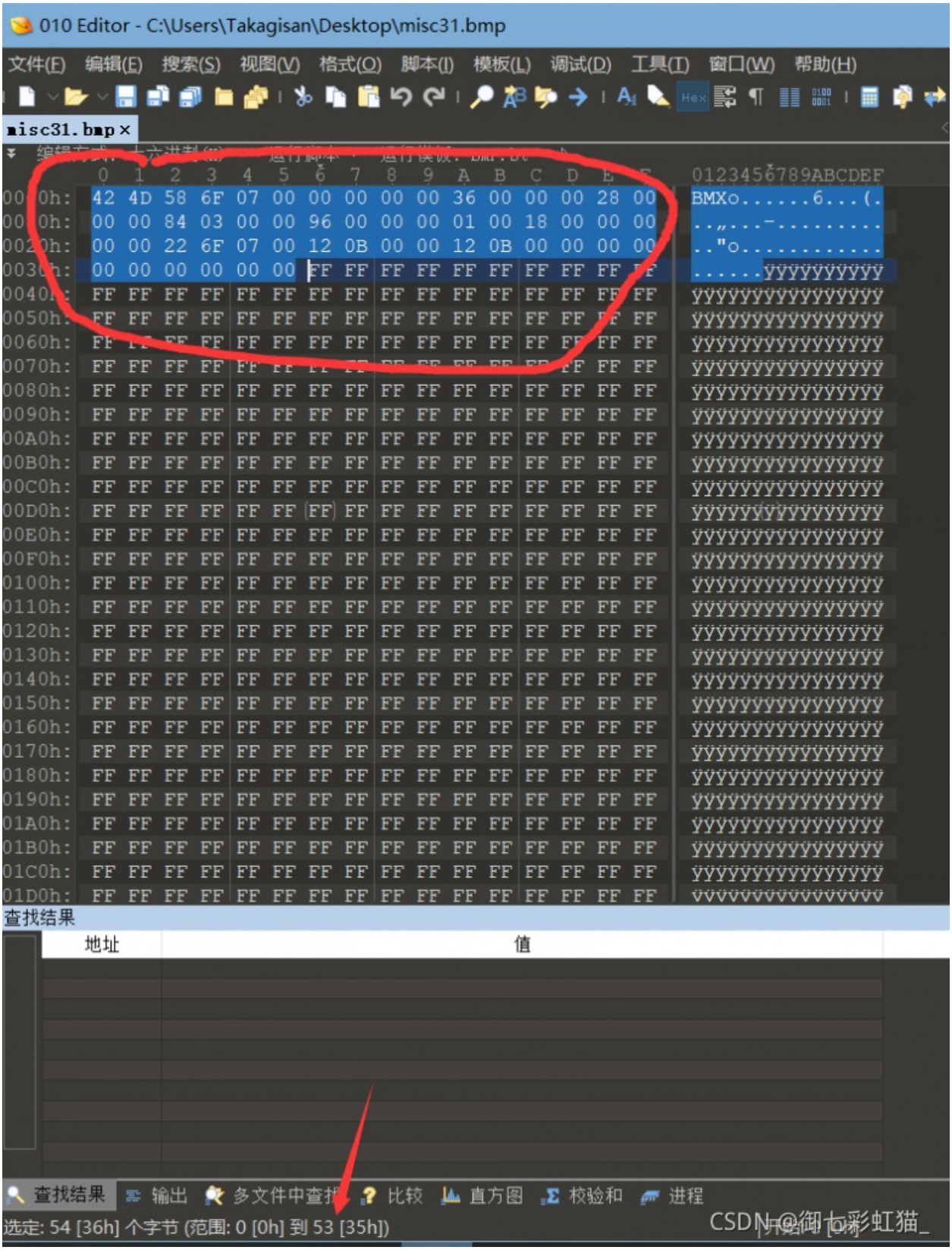




所以全部加起来是487253个字节



然后得到487253是全部数据，减去文件头的数据53个

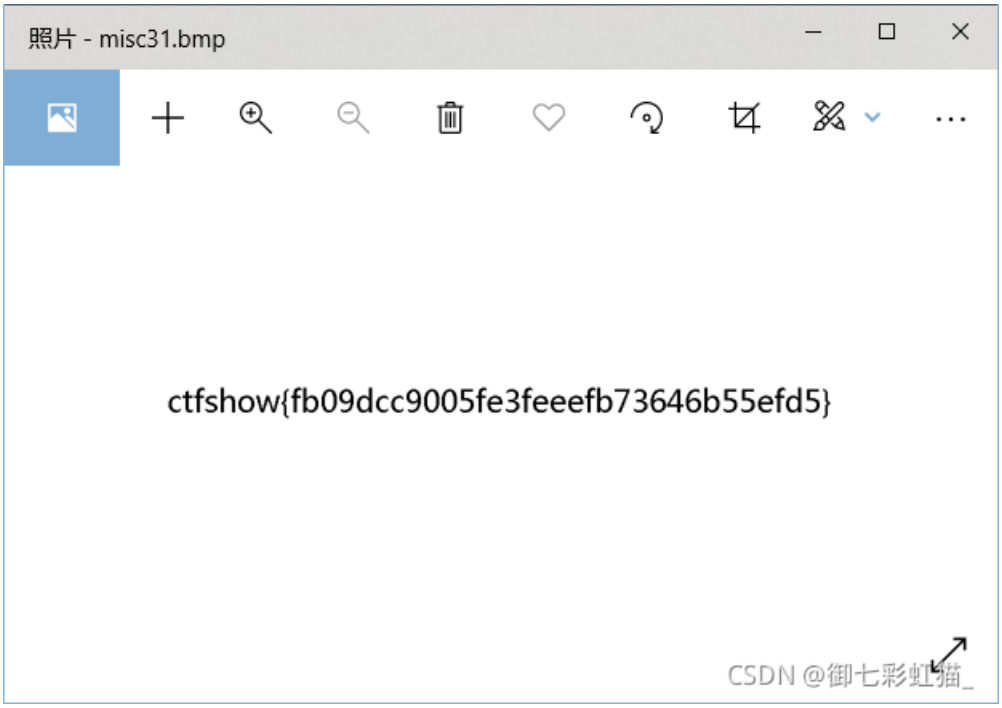


所以用（487253-53）/3/150，舍去余数就是1082

除以三是获取的像素值，在来除以高度150就能得到真正的宽度

修改宽度为1082

得到flag: `ctfshow{fb09dcc9005fe3feeefb73646b55efd5}`



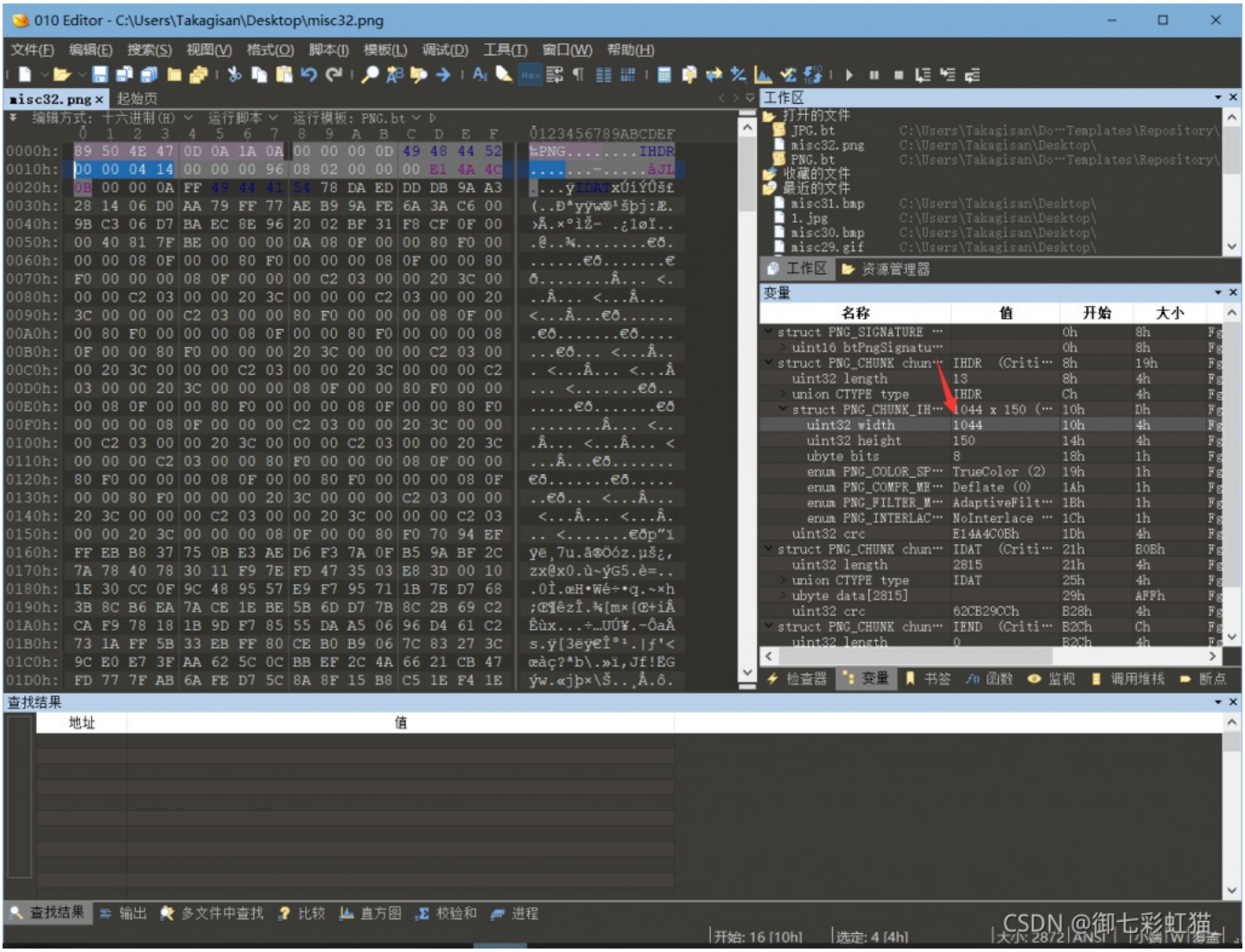
misc32

获取CRC E14A4C0B

和misc26一样用脚本跑这个png文件

跑出来是宽度是1044

然后在010edit修改宽度为1044即可




```
D:\OneDrive\Python\venv\Scripts\python.exe D:/OneDrive/Python/CTF/image/BaopoimageChangKuan.py
1044 150
hex: 0x414 0x96
```

代码如下

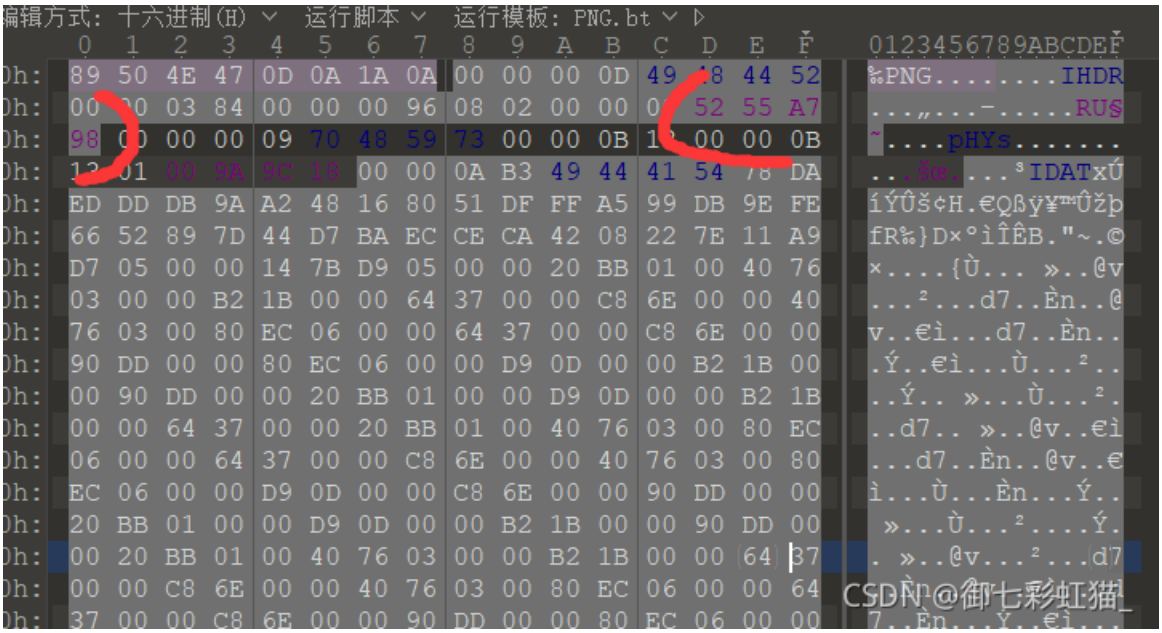
```python
import binascii
import struct

crcbp = open("misc32.png", "rb").read()
for i in range(2000):
    for j in range(2000):
        data = crcbp[12:16] + struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0xE14A4C0B):
            print(i, j)
            print('hex:', hex(i), hex(j))
```

得到flag

misc33

获取图片CRC

0x5255A798




```
crc32 = binascii.crc32(data) & 0xffff
if(crc32 == 0x5255A798):
    print(i, j)
```

还是一样用Python老跑出他的高度和宽度

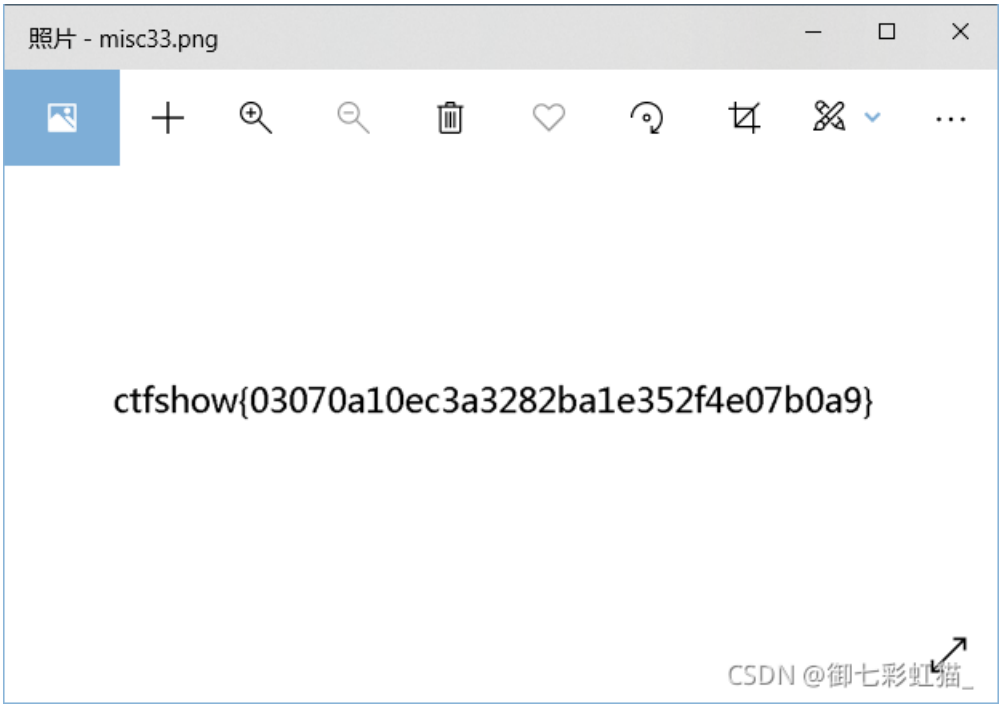
```
D:\ruanjian\PythonAnaconda\python.exe D:/OneDrive/Python/CTF/image/真正的高度.py
978 142
hex: 0x3d2 0x8e
```

下面是Python代码

```
import binascii
import struct

crcbp = open("misc33.png", "rb").read()
for i in range(2000):
    for j in range(2000):
        data = crcbp[12:16] + struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        if(crc32 == 0xE14A4C0B):
            print(i, j)
            print('hex:', hex(i), hex(j))
```

把宽度修改成978，宽度修改成142就可以了

获得flag: `ctfshow{03070a10ec3a3282ba1e352f4e07b0a9}`



misc34

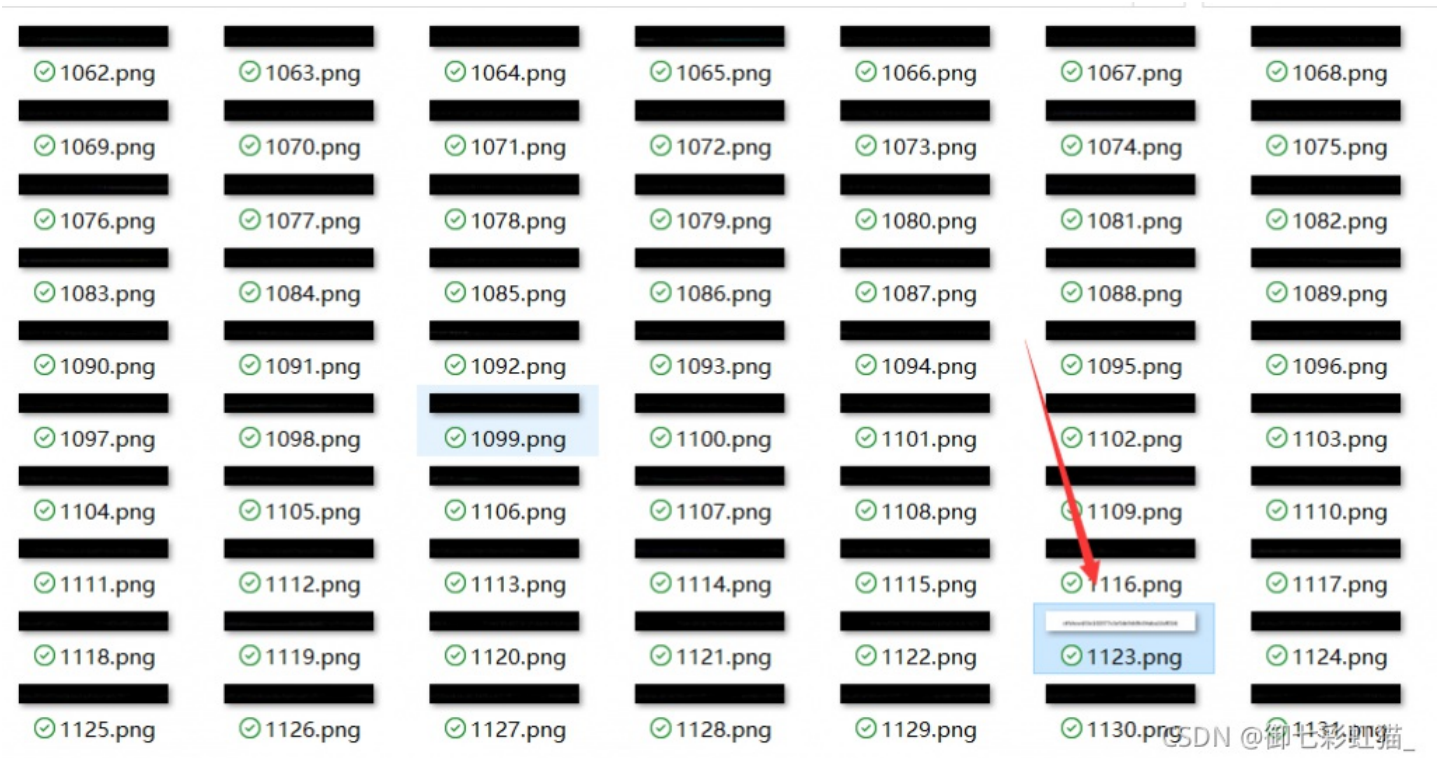
提示: 出题人狗急跳墙，把IHDR块的CRC也改了，但我们知道正确宽度肯定大于900

使用脚本跑

代码如下

```
import zlib
import struct
filename = "misc34.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    #w = all_b[16:20]
    #h = all_b[20:24]
    for i in range(901,1200):
        name = str(i) + ".png"
        f1 = open(name,"wb")
        im = all_b[:16]+struct.pack('>i',i)+all_b[20:]
        f1.write(im)
        f1.close()
```

跑出来一堆图片，一看，一堆黑色图片里面有一个白色



得到flag: `ctfshow{03e102077e3e5de9dd9c04aba16ef014}`

ctfshow{03e102077e3e5de9dd9c04aba16ef014}