

CTF.show: misc入门24-49

原创

[FW_ENJOEY](#) 于 2021-04-10 13:10:58 发布 591 收藏 3

分类专栏: [CTF.show](#) [CTF_MISC_Writeup](#)

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/qq_46230755/article/details/115311646

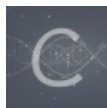
版权



[CTF.show](#) 同时被 2 个专栏收录

23 篇文章 4 订阅

订阅专栏



[CTF_MISC_Writeup](#)

24 篇文章 2 订阅

订阅专栏

上一次剩3题坐半天不会写。还是先把后面写了吧。

只是为了自己记录, 希望师傅们多提点建议多教教我。

文章目录

图片篇(文件结构)

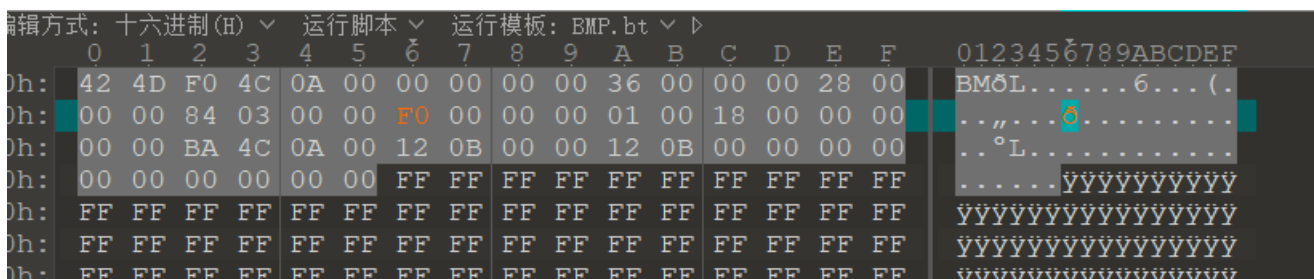
- misc24
- misc25
- misc26
- misc27
- misc28
- misc29
- misc30
- misc31
- misc32
- misc33
- misc34
- misc35
- misc36
- misc37
- misc38
- misc39
- misc40
- misc41
- misc42
- misc43
- misc44
- misc45
- misc46
- misc47
- misc48
- misc49

图片篇(文件结构)

misc24

提示: **flag**在图片上面。

010里面修改bmp的高度



ctfshow{dd7d8bc9e5e873eb7da3fa51d92ca4b7}



{there_is_no_flag_here}

https://blog.csdn.net/qq_46230755

ctfshow{dd7d8bc9e5e873eb7da3fa51d92ca4b7}

misc25

提示：flag在图片下面。

打开图片，调整图片高度，得到flag。

在这里插入图片描述

{there_is_no_flag_here}

ctfshow{494f611cc5842dd597f460874ce38f57}

利用脚本进行crc爆破

```

#coding=utf-8
import zlib
import struct
#读文件
file = 'C://Users/24471/Desktop/misc26.png' #注意, 1.png图片要和脚本在同一个文件夹下哦~
fr = open(file, 'rb').read()
data = bytearray(fr[12:29])
crc32key = eval(str(fr[29:33]).replace('\x', '').replace("b'", '0x').replace("'", ''))
#crc32key = 0xCBD6DF8A #补上0x, copy hex value
#data = bytearray(b'\x49\x48\x44\x52\x00\x00\x01\xF4\x00\x00\x01\xF1\x08\x06\x00\x00') #hex下copy grep hex
n = 4095 #理论上0xffffffff,但考虑到屏幕实际, 0x0fff就差不多了
for w in range(n):#高和宽一起爆破
    width = bytearray(struct.pack('>i', w))#q为8字节, i为4字节, h为2字节
    for h in range(n):
        height = bytearray(struct.pack('>i', h))
        for x in range(4):
            data[x+4] = width[x]
            data[x+8] = height[x]
            #print(data)
        crc32result = zlib.crc32(data)
        if crc32result == crc32key:
            print(width,height)
            #写文件
            newpic = bytearray(fr)
            for x in range(4):
                newpic[x+16] = width[x]
                newpic[x+20] = height[x]
            fw = open(file+'.png', 'wb')#保存副本
            fw.write(newpic)
            fw.close

```

得到一个新的png, 然后拖到010里面去查看

高度的hex为25e

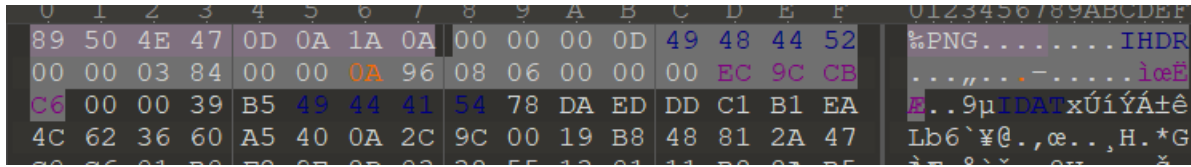
得到flag

```
ctfshow{94aef125e087a7ccf2e28e742efd704c}
```

misc26

提示: flag在图片下面。

和25一样，也是改高度，但是这次改的比较多



```
ctfshow{94aef1
+True height(hex) of this picture+
087a7ccf2e28e742efd704c}
```

得到flag，不过有提示，不完整。

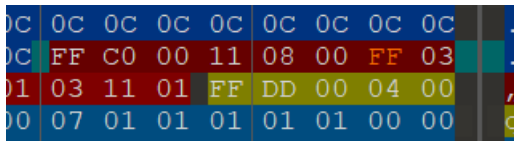
这里需要，找到图片的真实高度。

```
ctfshow{94aef0961087a7ccf2e28e742efd704c}
```

misc27

提示: flag在图片下面

图片是jpg，修改图片高度，把150的十六进制0096改成00FF得到flag



```
{there_is_no_flag_here}
```

```
ctfshow{5cc4f19eb01705b99bf41492430a1a14}
```

https://blog.csdn.net/qq_46230755

```
ctfshow{5cc4f19eb01705b99bf41492430a1a14}
```

misc28

提示: flag在图片下面。

图片是gif, 修改gif高度

一共要改两处

```

0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF
0000h: 47 49 46 38 39 61 84 03 FF 00 C4 00 00 00 00 00 GIF89a,,.ÿ.Ä.....
0010h: FF FF FF F4 F4 F4 E9 E9 E9 DD DD DD D1 D1 D1 C5 ÿÿôôééééÿÿÿÿÑÑÑÑ

```

```

0060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 21 F9 04 (!.....!ù.
0070h: 01 00 00 10 00 2C 00 00 00 00 84 03 FF 00 00 05 .....ÿ.
0080h: FF 60 20 8E 64 69 9E 68 AA AE 6C EB BE 70 2C CF ÿ` ždižh^@le³p,İ

```

{there_is_no_flag_here}

ctfshow{59c8bc525426166b1c893fe12a387fd7}

https://blog.csdn.net/qq_46230755

ctfshow{59c8bc525426166b1c893fe12a387fd7}

misc29

与上一题一样。对gif全部帧进行替换。

```

87 4C 64 62 06 04 00 21 F9 04 01 32 00 02 00 2C
00 00 00 00 84 03 FF 00 87 00 00 00 FF FF FF FF
FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00

```

搜索全部的 96 00 换成 FF 00

{there_is_no_flag_here}

ctfshow{03ce5be6d60a4b3c7465ab9410801440}

https://blog.csdn.net/qq_46230755

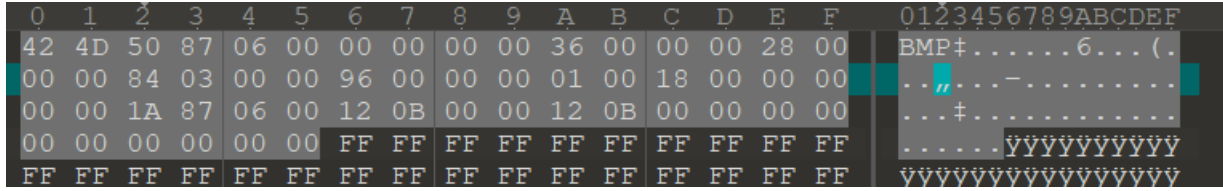
```
ctfshow{03ce5be6d60a4b3c7465ab9410801440}
```

misc30

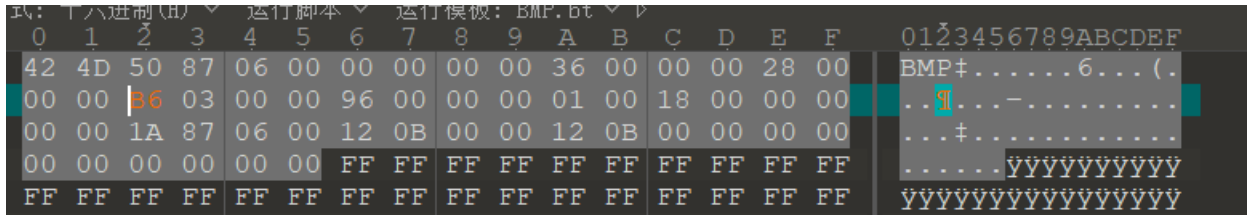
提示：正确的宽度是950。

把图片放入010，在宽度的位置进行修改成950的二进制 03 B6，注意要倒着写。

(我一开始也不知道的，是打开来发现宽度900写的是84 03，所以才想bmp是反着的)



改完得到flag

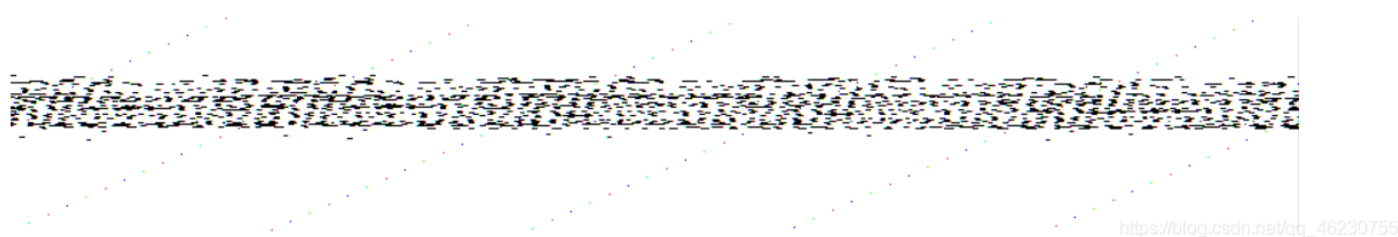


```
ctfshow{6db8536da312f6aeb42da2f45b5f213c}
```

```
ctfshow{6db8536da312f6aeb42da2f45b5f213c}
```

misc31

提示：高度是正确的，但正确的宽度是多少呢。



题目提示了改宽度
从v3师傅的博客抄的爆破bmp宽度的脚本

```

import struct
import zlib
f = open('./misc31.bmp', 'rb')
c = f.read()
width = c[18:22]
height = c[22:26]
# 爆破bmp宽度
for i in range(900,1100):
    f1 = open('./bpout/'+str(i)+'.bmp', 'wb')
    # print(struct.pack('>i',i)[::-1])
    img = c[:18]+struct.pack('>i',i)[::-1]+c[22:]
    f1.write(img)
    f1.close()

```

在1082.bmp找到正确的flag

ctfshow{fb09dcc9005fe3feefb73646b55efd5}

ctfshow{fb09dcc9005fe3feefb73646b55efd5}

misc32

提示：高度是正确的，但正确的宽度是多少呢

CRC爆破宽度

```

import zlib
import struct
image=open("misc32.png", "rb").read()
for i in range(4096):
    for j in range(4096):
        c=image[12:16]+struct.pack('>i',i)+struct.pack('>i',j)+image[24:29]
        CRC=0xE14A4C0B
        if zlib.crc32(c)==CRC:
            print(hex(i),hex(j))
            exit(0)

```

ctfshow{685082227bcf70d17d1b39a5c1195aa9}

ctfshow{685082227bcf70d17d1b39a5c1195aa9}

misc33

提示：出题人丧心病狂，把高度也改了

和上题一模一样的脚本，爆破宽高


```

import zlib
import struct
image=open("misc33.png","rb").read()
for i in range(4096):
    for j in range(4096):
        c=image[12:16]+struct.pack('>i',i)+struct.pack('>i',j)+image[24:29]
        CRC=0x5255A798
        if zlib.crc32(c)==CRC:
            print(hex(i),hex(j))
            exit(0)

```

<ctfshow{03070a10ec3a3282ba1e352f4e07b0a9}

```
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
 由于CRC也被修改了，所以我们对宽度进行爆破，然后自己找图片

```

import zlib
import struct
image=open("misc34.png","rb").read()
width=image[16:20]
height=image[20:24]
for i in range(900,1200):
    image1=open(str(i-900)+'.png','wb')
    change=image[:16]+struct.pack('>i',i)+image[20:]
    image1.write(change)
    image1.close()

```

ctfshow{03e102077e3e5de9dd9c04aba16ef014}

```
ctfshow{03e102077e3e5de9dd9c04aba16ef014}
```

misc35

提示：出题人负隅顽抗，但我们知道正确宽度肯定大于900
 先改高度为\x02\x96，

```

import zlib
import struct
image=open("misc35.png","rb").read()
for i in range(900,1200):
    image1=open(str(i-900)+'.png','wb')
    change=image[:image.index(b'\x02\x96\x03\x84')+2]+struct.pack('>h',i)+image[image.index(b'\x02\x96\x03\x84')+4:]
    image1.write(change)
    image1.close()

```

ctfshow{ca35201ca9ed607e5a68f44ef573fbc3}

ctfshow{ca35201ca9ed607e5a68f44ef573fbc3}

misc36

提示：出题人坦白从宽，正确的宽度在920-950之间
先把gif的高度改了，记得要改两处。

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | 0123 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|
| 47 | 49 | 46 | 38 | 39 | 61 | 84 | 03 | 90 | 01 | 91 | 00 | 00 | 00 | 00 | 00 | GIF8 |
| FF | FF | FF | FF | FF | FF | 00 | 00 | 00 | 21 | F9 | 04 | 01 | 00 | 00 | 02 | YYYY |
| 00 | 2C | 00 | 00 | 00 | 00 | 84 | 03 | 90 | 01 | 00 | 02 | FF | 8C | 8F | A9 | ... |
| CB | ED | 0F | A3 | 9C | B4 | DA | 8B | B3 | DE | BC | FB | 0F | 86 | E2 | 48 | Ë1.þ |
| 96 | E6 | 89 | A6 | EA | CA | B6 | EE | 0B | C7 | F2 | 4C | D7 | F6 | 8D | E7 | -æ%! |
| FA | CE | F7 | FE | 0F | 0C | 0A | 87 | C4 | A2 | F1 | 88 | 4C | 2A | 97 | CC | úÎ÷þ |
| A6 | F3 | 09 | 8D | 4A | A7 | D4 | AA | F5 | 8A | CD | 6A | B7 | DC | AE | F7 | !ó.. |
| 0B | 0E | 8B | C7 | E4 | B2 | F9 | 8C | 4E | AB | D7 | EC | B6 | FB | 0D | 8F | ..<Q |
| CB | E7 | F4 | BA | FD | 8E | CF | EB | F7 | FC | BE | FF | 0F | 18 | 28 | 38 | Ëçô° |
| 48 | 58 | 68 | 78 | 88 | 98 | A8 | B8 | C8 | D8 | E8 | F8 | 08 | 19 | 29 | 39 | HXhx |
| 49 | 59 | 69 | 79 | 89 | 99 | A9 | B9 | C9 | D9 | E9 | F9 | 09 | 1A | 2A | 3A | IYiy |
| 4A | 5A | 6A | 7A | 8A | 9A | AA | BA | CA | DA | EA | FA | 0A | 1B | 2B | 3B | JZjz |
| 4B | 5B | 6B | 7B | 8B | 9B | AB | BB | CB | DB | EB | FB | 0B | 1C | 2C | 3C | K[k{ |
| 4C | 5C | 6C | 7C | 8C | 9C | AC | BC | CC | DC | EC | FC | 0C | 1D | 2D | 3D | L[l |
| 4D | 5D | 6D | 7D | 8D | 9D | AD | BD | CD | DD | ED | FD | 0D | 1E | 2E | 3E | M m} |
| 4E | 5E | 6E | 7E | 8E | 9E | AE | BE | CE | DE | EE | FE | 0E | 1F | 2F | 3F | N^n~ |
| 4F | 5F | 6F | 7F | 8F | 9F | AF | BF | CF | DF | EF | FF | 0F | 30 | A0 | C0 | O_o. |
| 81 | 04 | 0B | 1A | 3C | 88 | 30 | A1 | C2 | 85 | 0C | 1B | 3A | 7C | 08 | 31 | |
| A2 | C4 | 89 | 14 | 2B | 5A | BC | 88 | 31 | A3 | C6 | 8D | FF | 1C | 3B | 7A | φÄ%. . |
| FC | 08 | 32 | A4 | C8 | 91 | 24 | 4B | 9A | 3C | 89 | 32 | A5 | CA | 95 | 2C | ü.2π |
| 5B | BA | 7C | 09 | 33 | A6 | CC | 99 | 6E | 00 | D8 | 04 | 90 | E6 | A6 | 9E | [° . . |
| 9B | 36 | 11 | F0 | 64 | F2 | 93 | 66 | 41 | 9D | 17 | 82 | 9A | E1 | 89 | D3 | >6.ð |
| 0E | 52 | A2 | 01 | 8C | 26 | 71 | 1A | 85 | A9 | D0 | 56 | 50 | 27 | 54 | 15 | .Rç. |
| 83 | 54 | E9 | D2 | 9E | 06 | AE | 16 | F1 | DA | 04 | EC | 54 | 52 | 52 | 2D | fTéò |
| 88 | F5 | 92 | B5 | 4E | DA | A6 | 07 | CE | 06 | 71 | BB | A4 | EC | D8 | 52 | ^ð'µ |
| 67 | B9 | 2E | 80 | CB | 45 | 2E | 9C | B5 | 09 | F0 | FA | F0 | 8B | 04 | F0 | g¹.e |
| 5C | 4C | 7A | DB | 16 | 6E | 7A | B8 | 1A | 58 | C1 | 3B | 18 | 1B | 49 | 3C | \LzÛ |
| B8 | 53 | E2 | C5 | 90 | A7 | 51 | B6 | FB | B4 | 32 | 50 | CC | 91 | 45 | 4D | ,sâÄ |
| 3E | EC | 78 | D9 | E5 | A4 | 4A | 42 | 13 | D1 | DC | 99 | 30 | E7 | BE | A0 | >ìxÛ |
| 51 | 43 | 1B | BD | 79 | B5 | 14 | D7 | A9 | 2B | 7D | 96 | DD | 95 | B6 | 33 | QC.¿ |
| D8 | 71 | 75 | 1F | F1 | 5D | 3B | 92 | DC | AD | 4B | 7D | 4A | DD | 4A | A1 | øqu. |
| B8 | 03 | A3 | 62 | 95 | 7F | A8 | CB | 77 | 83 | 53 | E4 | 56 | 9D | 77 | 20 | .,fb |
| CE | 17 | 7A | F4 | 07 | D6 | 4D | 1B | C6 | 4C | 5D | 82 | F5 | BB | 4C | 9B | Ï.zð |
| 6E | 27 | 4E | 3A | E8 | A7 | E1 | D8 | 8E | 13 | 6D | 1E | 01 | 3E | 83 | A0 | o!O. |

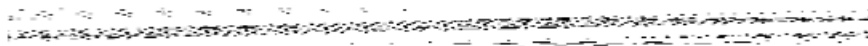
3989B500d01 E:\F...225\
misc36.gif D:\Py...main\
PNG.bt C:\Us...tory\
收藏的文件
最近的文件
misc32.png C:\Us...ktop\
data2 C:\Us...ktop\
data1 C:\Us...ktop\
hongbao2.jpg C:\Us...ktop\
1.ra C:\Us...ktop\
messi.zip C:\Us...essi\
encode.mp3 C:\Us...ktop\
1.png C:\Us...ktop\
00083771.png C:\Us...ktop\
me.zip C:\Us...ktop\
冰墩墩.png C:\Us...ktop\
misc29.gif C:\Us...ktop\
工作区 资源管理器

变量

| 名称 | |
|--------------------------|------|
| > struct RGB rgb[2] | #FFF |
| > struct RGB rgb[3] | #000 |
| ▼ struct DATA Data | |
| > struct GRAPHICCONT... | |
| ▼ struct IMAGEDESCRIP... | |
| UByte ImageSeper... | 44 |
| ushort ImageLeftPo... | 0 |
| ushort ImageTopPo... | 0 |
| ushort ImageWidth | 900 |
| ushort ImageHeight | 400 |
| > struct IMAGEDESCR... | |
| > struct IMAGEDATA Im... | |
| > struct TRAILER Trailer | |

https://blog.csdn.net/cq_40230755

```
import zlib
import struct
image=open("misc36.gif","rb").read()
for i in range(920,950):
    image1=open(str(i-900)+'_gif','wb')
    change=image[:image.index(b'\x00\x00\x84\x03\x90\x01')+2]+struct.pack('>h',i)[::-1]+image[image.index(b'\x00\x00\x84\x03\x90\x01')+4:]
    image1.write(change)
    image1.close()
```



ctfshow{1ebf739f832906d60f57436b8179166f}

https://blog.csdn.net/qq_46230755

ctfshow{1ebf739f832906d60f57436b8179166f}

misc37

提示: **flag在图片里**

flag存在其中的几帧, 直接看就可以。

ctfshow{

2056782c

d57b1326

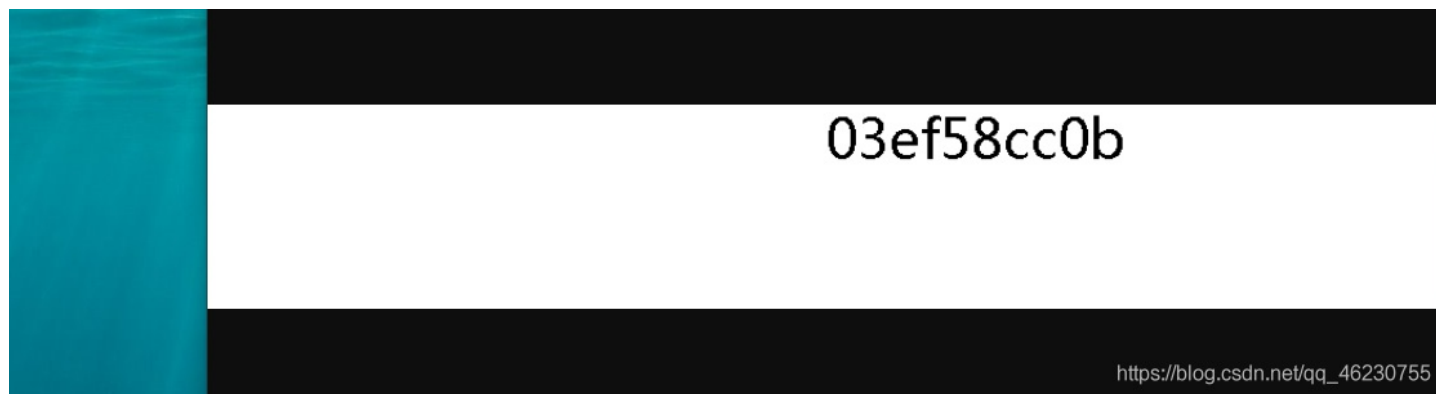
1dcbbe3d

6eecda17}

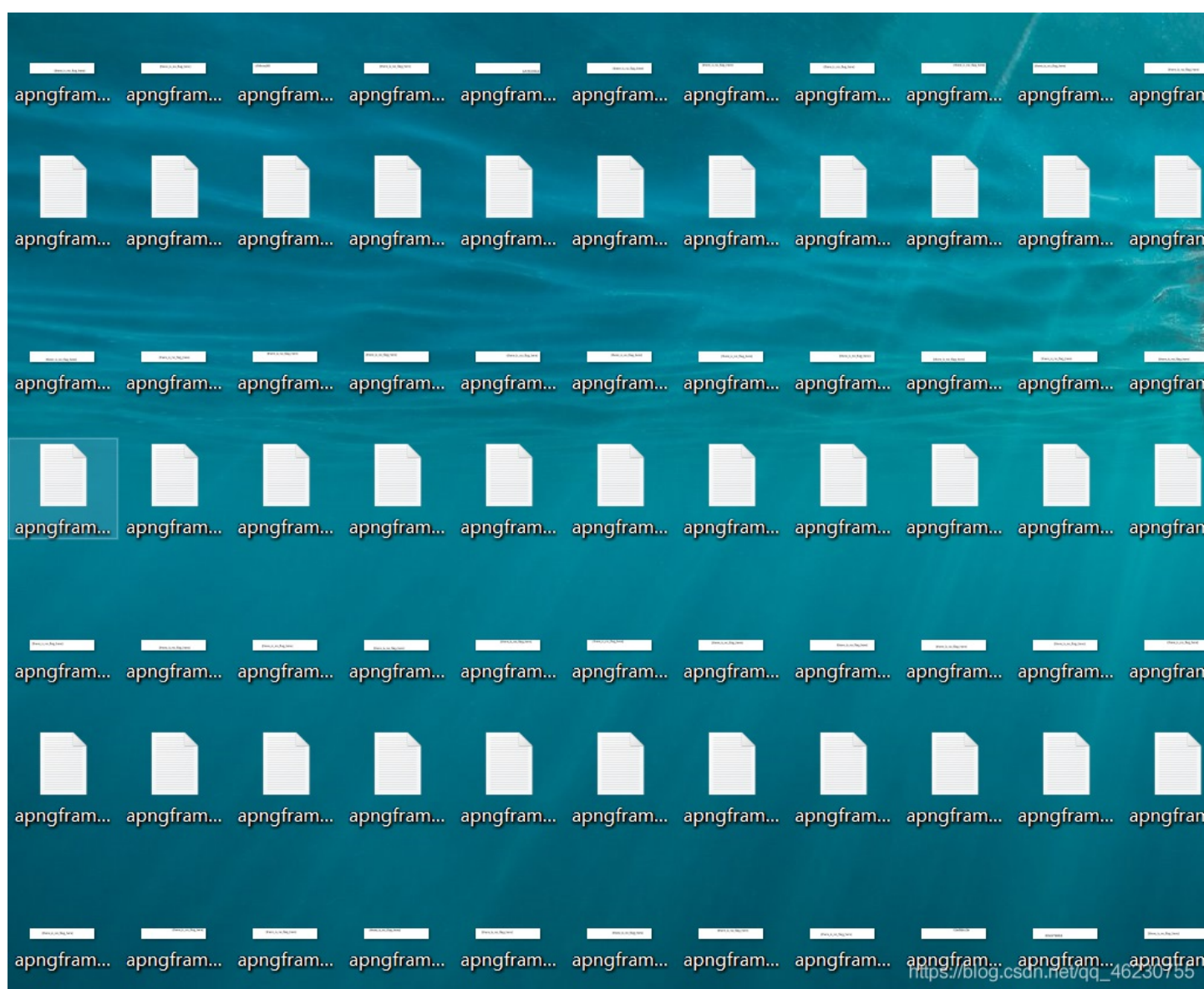
```
ctfshow{2056782cd57b13261dcbbe3d6eecda17}
```

[misc38](#)

提示: flag在图片里
是一个动态的png



用工具 APNG Disassembler 分离 (下载地址)



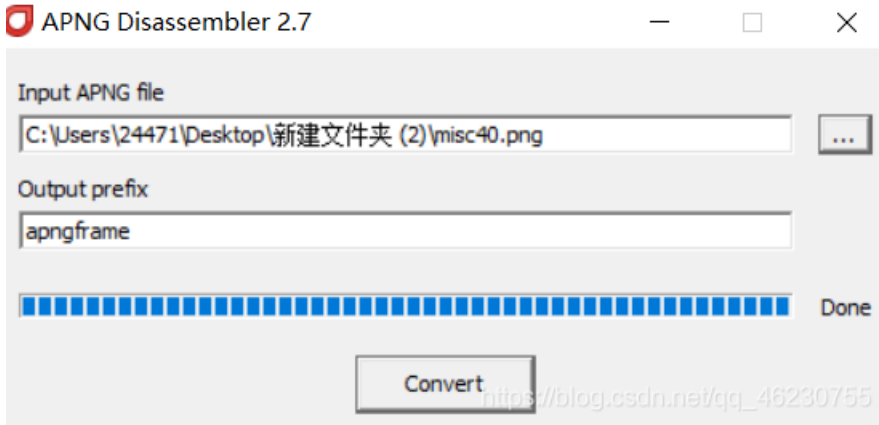
然后找就行

```
ctfshow{48b722b570c603ef58cc0b83bbf7680d}
```

misc39

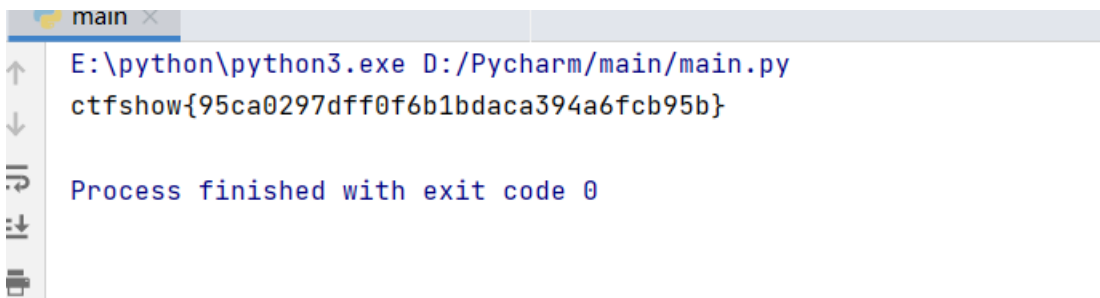
提示: flag就像歌, 有长有短仿佛岁月悠悠

一样用到上题的软件



读取每一个txt文件然后去掉干扰从28开始。

```
flag=''
for i in range(28,69):
    f= open('C:/Users/24471/Desktop/新建文件夹 (2)/apngframe'+str(i)+'.txt')
    j=f.read()
    flag+=chr(int(j.split('/')[0].split('=')[1]))
print(flag)
```

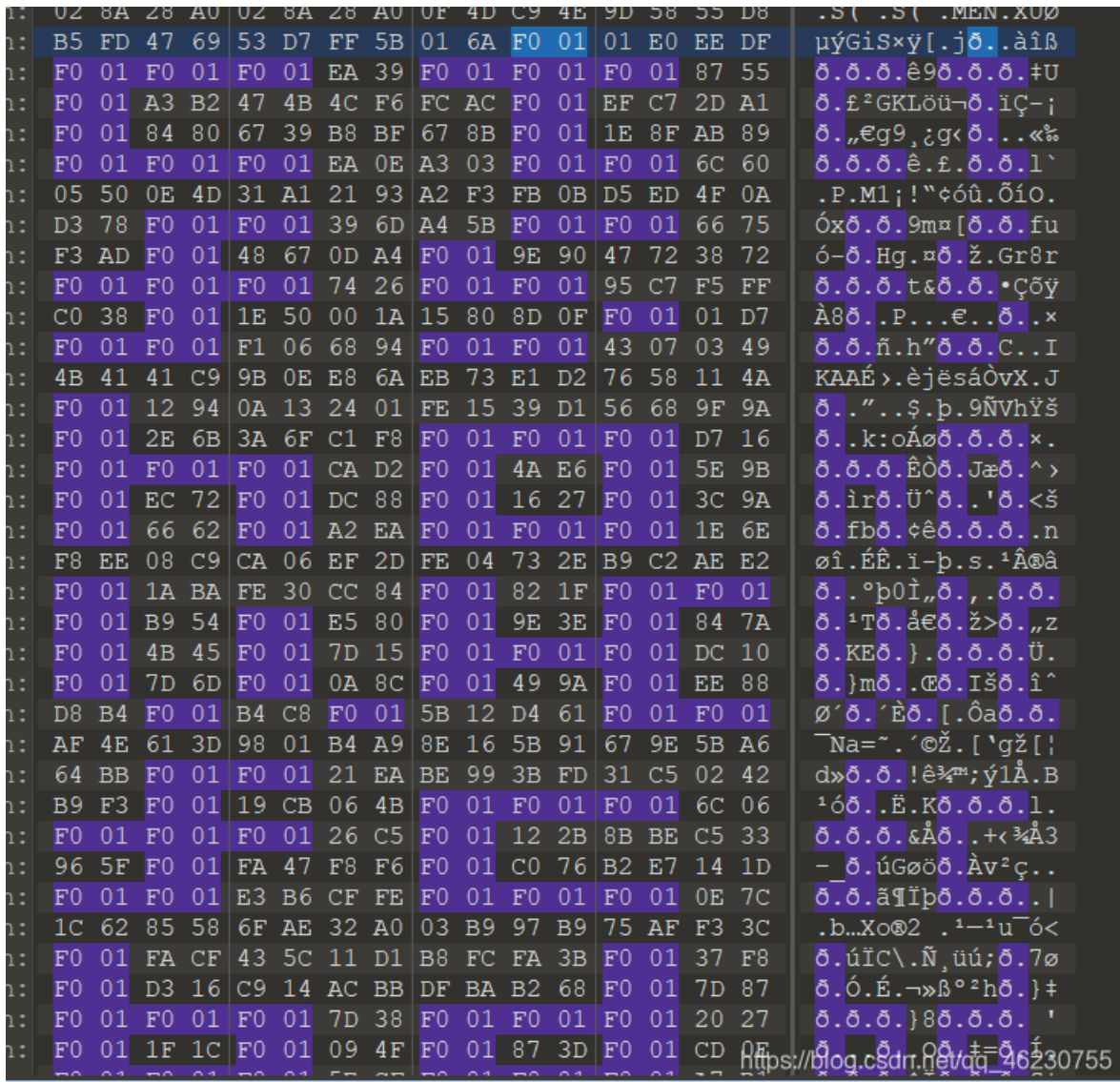


```
ctfshow{95ca0297dff0f6b1bdaca394a6fcb95b}
```

misc41

提示: H4ppy Apr11 F001's D4y! 愚人节到了, 一群笨蛋往南飞, 一会儿排成S字, 一会儿排成B字。

在010里面搜索F0 01 全部高亮



ctfshow{fcbd427caf4a52f1147ab44346cd1cdd}

misc42

提示: flag有多长? 2cm.....不好意思打错了, 41位

我猜这题在黑套宝。

| | | | |
|-----|-----------|----------|----------------|
| 99 | d639e... | critical | PNG image data |
| 116 | af63a2... | critical | PNG image data |
| 102 | d7127... | critical | PNG image data |
| 115 | b5296... | critical | PNG image data |
| 104 | dce9d... | critical | PNG image data |
| 111 | 302ca... | critical | PNG image data |
| 119 | 927d6... | critical | PNG image data |
| 123 | 6ef517... | critical | PNG image data |
| 48 | 98574... | critical | PNG image data |
| 55 | 866b9... | critical | PNG image data |
| 56 | b7453... | critical | PNG image data |
| 99 | 4fb61... | critical | PNG image data |
| 98 | 5a119f... | critical | PNG image data |
| 100 | 657dd... | critical | PNG image data |
| 48 | 285d6... | critical | PNG image data |
| 102 | 004bb... | critical | PNG image data |
| 57 | 295cc... | critical | PNG image data |
| 99 | f766e2... | critical | PNG image data |
| 56 | 43c63... | critical | PNG image data |
| 100 | db791... | critical | PNG image data |
| 51 | 593c0... | critical | PNG image data |
| 102 | 83742... | critical | PNG image data |

https://blog.csdn.net/qq_45230755

放入tweakpng然后进行长度的提取, 批量转

```
str=[229,152,191,229,152,191,49,99,116,102,115,104,111,119,123,48,55,56,99,98,100,48,102,57,99,56,100,51,102,50,49,53,56,101,55,48,53,50,57,102,56,57,49,51,99,54,53,125]
flag=''
for i in str:
    flag+=chr(i)
print(flag)
```

```
main
E:\python\python3.exe D:/Pycharm/main/main.py
å ¿å ¿1ctfshow{078cbd0f9c8d3f2158e70529f8913c65}

Process finished with exit code 0
```

```
ctfshow{078cbd0f9c8d3f2158e70529f8913c65}
```

misc43

提示：错误中隐藏着通往正确答案的道路

猜测与crc错误有关。

利用pngdebugger提取

```
0x00000021      chunk-length=0x00000180 (384)
0x00000025      chunk-type=' IDAT'
0x000001A9      crc-code=0xE59387E5
>> (CRC CHECK)  crc-computed=0x8385F691      =>      CRC FAILED

0x000001AD      chunk-length=0x00000180 (384)
0x000001B1      chunk-type=' IDAT'
0x00000335      crc-code=0x93A62E63
>> (CRC CHECK)  crc-computed=0x42434298      =>      CRC FAILED

0x00000339      chunk-length=0x00000180 (384)
0x0000033D      chunk-type=' IDAT'
0x000004C1      crc-code=0x74667368
>> (CRC CHECK)  crc-computed=0x4462C3A1      =>      CRC FAILED

0x000004C5      chunk-length=0x00000180 (384)
0x000004C9      chunk-type=' IDAT'
0x0000064D      crc-code=0x6F777B36
>> (CRC CHECK)  crc-computed=0x397611E1      =>      CRC FAILED

0x00000651      chunk-length=0x00000180 (384)
0x00000655      chunk-type=' IDAT'
0x000007D9      crc-code=0x65623235
>> (CRC CHECK)  crc-computed=0x4E02AFA2      =>      CRC FAILED
```

https://blog.csdn.net/qq_46230755

把错误的code提出来用hex转字符

```
import binascii
str='E59387E593A62E63746673686F777B36656232353839666666663565333930666536623837353034646263303839327D'
print(binascii.a2b_hex(str))
```

```
E:\python\python3.exe D:/Pycharm/main/main.py
b'\xe5\x93\x87\xe5\x93\xa6.ctfshow{6eb2589ffff5e390fe6b87504dbc0892}'

ctfshow{6eb2589ffff5e390fe6b87504dbc0892}
```

misc44

提示：错误中还隐藏着坑

利用pngdebugger进行解析

```
PNGDebugger.exe misc44.png >1.txt
```

```
1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
-----
file-path=misc44.png
file-size=400414 bytes

0x00000000      png-signature=0x89504E470D0A1A0A

0x00000008      chunk-length=0x0000000D   (13)
0x0000000C      chunk-type='IHDR'
0x0000001D      crc-code=0x09DAD161
>> (CRC CHECK) crc-computed=0x09DAD161   =>      CRC OK!

0x00000021      chunk-length=0x00000480   (1152)
0x00000025      chunk-type='IDAT'
0x0000004A9     crc-code=0x8F84D7BB
>> (CRC CHECK) crc-computed=0x8F84D7BB   =>      CRC OK!

0x0000004AD     chunk-length=0x00000480   (1152)
0x0000004B1     chunk-type='IDAT'
0x000000935     crc-code=0xAAAF3DA3
>> (CRC CHECK) crc-computed=0xAAAF3DA3   =>      CRC OK!

0x000000939     chunk-length=0x00000480   (1152)
https://blog.csdn.net/qq_46230755
```

提取正确的CRC OK!为1，错误的CRC FAILED为0，删除无用的前10行和后四行。

```
f=open("1.txt")
str=''
while 1:
    s=f.readline()
    if s:
        if 'OK!' in s:
            str+='1'
        elif 'FAILED' in s:
            str+='0'
    else:
        break
print(str)
print(len((str)))
for i in range(len(str)//8):
    print(chr(int(str[i*8:(i+1)*8],2)),end="")
```

```
main ^
E:\python\python.exe D:/Pycharm/main/main.py
1111111111111111101100011011101000110011001110011011010000110111101110111011
344
ÿÿctfshow{cc1af32bf96308fc1263231be783f69e}
进程已结束。退出代码为 0
```

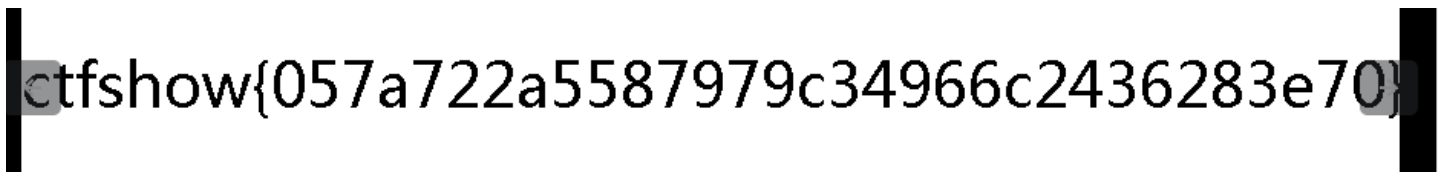
```
ctfshow{cc1af32bf96308fc1263231be783f69e}
```

misc45

提示: 有时候也需要换一换思维格式

因为bmp与png的读取像素的方式不一样,所以我们可以将其先进行转换格式

然后 `binwalk misc45 -e`



```
ctfshow{057a722a5587979c34966c2436283e70}
```

misc46

提示: 你见过扶乩吗

利用identify进行提取

```
identify misc46.gif > 1.txt
```

再利用一个画图的模块画图。

根据的是每一帧的偏移坐标

以下脚本参考V3师傅博客:

<https://blog.csdn.net/xzczhf/article/details/115434194?spm=1001.2014.3001.5501>

```
from PIL import Image
import matplotlib.pyplot as plt
f=open('1.txt')
pp=[]
while 1:
    c=f.readline()
    if c:
        s=eval(c.split('+')[1]+'+'+c.split('+')[2][:2])
        pp.append(s)
    else:
        break
img =Image.new('RGB',(400,70),(255,255,255))
for i in pp:
    new = Image.new('RGB',(1,1),(0,0,0))
    img.paste(new,i)
img.save('1.png')
```

```
ctfshow{05906b3be8742a13a93898186bc5802f}
```

```
ctfshow{05906b3be8742a13a93898186bc5802f}
```

misc47

提示: 没见过扶乩, 那你知道笔仙吗

是png文件, 利用浏览器打开

```
{there_is_no_flag_here}ag_here}!}
```

看别的师傅博客发现需要提取其偏移位。

然后编写脚本进行提取。

| | | | | | |
|---------------------------|-------------------|------|------|-----|-----|
| > struct PNG CHUNK I... | 900 x 150 (x8) | 10h | Dh | Fg: | Bg: |
| uint32 crc | 9DAD161h | 1Dh | 4h | Fg: | Bg: |
| > struct PNG CHUNK chu... | acTL (Ancillar... | 21h | 14h | Fg: | Bg: |
| > struct PNG CHUNK chu... | fcTL (Ancillar... | 35h | 26h | Fg: | Bg: |
| > struct PNG CHUNK chu... | IDAT (Critical... | 5Bh | 2F1h | Fg: | Bg: |
| > struct PNG CHUNK chu... | fcTL (Ancillar... | 34Ch | 26h | Fg: | Bg: |
| > struct PNG CHUNK chu... | fdAT (Ancillar... | 372h | 878h | Fg: | Bg: |
| > struct PNG CHUNK chu... | fcTL (Ancillar... | BEAh | 26h | Fg: | Bg: |
| > struct PNG CHUNK chu... | fdAT (Ancillar... | C10h | 878h | Fg: | Bg: |

v3师傅博客

```
import struct
from PIL import Image
import matplotlib.pyplot as plt
f=open('misc47.png','rb')
c=f.read()
c=c[c.index(bytes.fromhex('6663544C00000001')):]
pp=[]
for i in range(1,1124,2):
    start=c.index(bytes.fromhex('6663544C0000')+struct.pack('>h',i))
    fc=c[start:start+30]
    pp.append(struct.unpack('>h',fc[18:20])+struct.unpack('>h',fc[22:24]))

img =Image.new('RGB',(400,70),(255,255,255))
for i in pp:
    new = Image.new('RGB',(1,1),(0,0,0))
    img.paste(new,i)
img.save('1.png')
```

ctfshow{6d51f85b45a0061754a2776a32cf26c4}

ctfshow{6d51f85b45a0061754a2776a32cf26c4}

misc48

提示：附件的第 (Di) 七 (Qi) 题 (Ti) 中有提示。本题略脑洞，可跳过

| | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|
| 5 | 06 | 06 | 07 | 07 | 08 | 07 | 07 | 06 | 09 | 09 | 0A | 0A | 09 | 09 | 0C | |
| C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 01 | 03 | |
| 3 | 03 | 05 | 04 | 05 | 09 | 06 | 06 | 09 | 0D | 0A | 09 | 0A | 0D | 0F | 0E | |
| E | 0E | 0E | 0F | 0F | 0C | 0C | 0C | 0C | 0C | 0F | 0F | 0C | 0C | 0C | 0C | |
| 3 | 6F | 75 | 6E | 74 | 0C | 46 | 46 | 0C | 26 | 0C | 6D | 69 | 6E | 75 | 73 | count.FF.&.minus |
| C | 31 | 0C | 63 | 74 | 66 | 73 | 68 | 6F | 77 | 7B | 33 | 32 | 7D | 0C | FF | .1.ctfshow{32}.y |
| 0 | 00 | 11 | 08 | 00 | 96 | 03 | 84 | 03 | 01 | 11 | 00 | 02 | 11 | 01 | 03 | À.....-..... |

提示我们计算FF的数量。

八神师傅说是统计每两个有意义块之间的FF的数量再减一，因为每两个有意义块之间插入数据好像是不太影响的，至于要减一是因为这段中的最后一个FF是下一个有意义块的开头，取前32个段即可，会发现每个段长度都不超过16，所以直接转为16进制就是flag

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| FF | D8 | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | EE |
| 00 | 0E | 41 | 64 | 6F | 62 | 65 | 00 | 64 | 40 | 00 | 00 | 00 | 01 | FF | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | DB | 00 | 84 | 00 | 02 | 02 |
| 02 | 02 | 02 | 02 | 02 | 02 | 02 | 02 | 03 | 02 | 02 | 02 | 03 | 04 | 03 | 02 |
| 02 | 03 | 04 | 05 | 04 | 04 | 04 | 04 | 04 | 05 | 06 | 05 | 05 | 05 | 05 | 05 |
| 05 | 06 | 06 | 07 | 07 | 08 | 07 | 07 | 06 | 09 | 09 | 0A | 0A | 09 | 09 | 0C |
| 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 0C | 01 | 03 |
| 03 | 03 | 05 | 04 | 05 | 09 | 06 | 06 | 09 | 0D | 0A | 09 | 0A | 0D | 0F | 0E |
| 0E | 0E | 0E | 0F | 0F | 0C | 0C | 0C | 0C | 0C | 0F | 0F | 0C | 0C | 0C | 0C |
| 63 | 6F | 75 | 6E | 74 | 0C | 46 | 46 | 0C | 26 | 0C | 6D | 69 | 6E | 75 | 73 |
| 0C | 31 | 0C | 63 | 74 | 66 | 73 | 68 | 6F | 77 | 7B | 33 | 32 | 7D | 0C | FF |
| C0 | 00 | 11 | 08 | 00 | 96 | 03 | 84 | 03 | 01 | 11 | 00 | 02 | 11 | 01 | 03 |
| 11 | 01 | FF | FF | FF | FF | FF | FF | FF | FF | DD | 00 | 04 | 00 | 71 | FF |
| FF | FF | FF | FF | FF | FF | FF | FF | FF | FF | C4 | 01 | A2 | 00 | 00 | 00 |
| 07 | 01 | 01 | 01 | 01 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 05 |
| 03 | 02 | 06 | 01 | 00 | 07 | 08 | 09 | 0A | 0B | 01 | 00 | 02 | 02 | 03 | 01 |

手动算一下得到

0 12 11 0 7 10 13 13 9 0 9 13 0 13 6 0 10 9 2 1 0 1 10 8 11 5 12 7 2 2 3 10

```
s=[0,12,11,0,7,10,13,13,9,0,9,13,0,13,6,0,10,9,2,1,0,1,10,8,11,5,12,7,2,2,3,10]
f='0123456789abcdef'
flag='ctfshow{'
for i in range(len(s)):
    flag+=f[s[i]]
flag+='}'
print(flag)
```

```
ctfshow{0cb07add909d0d60a92101a8b5c7223a}
```

misc49

提示：它们一来就是十六种。本题略脑洞，可跳过
丢入010

```
9 44 3D 34 00 FF E6 00 13 47 6F 50 72 6F 00 eID=4.ÿæ..GoPro.
4 5A 4F 4D 20 3D 20 59 3E 00 FF E7 00 10 48 <DZOM = Y>.ÿç..H
1 77 65 69 00 4D 61 74 65 00 38 00 FF E1 00 uawei.Mate.8.ÿá.
5 78 69 66 00 00 4D 4D 00 2A 00 00 00 08 00 :Exif..MM.*.....
1 10 00 01 00 00 00 01 01 00 00 00 51 11 00 .Q.....Q..
```

看到一堆的设备

问了下八神师傅，原来E后面的那位就是flag，最后按照这些块的顺序把E后面那位合起来就行了，最后如下

```
          C D E F 0123456789ABCDEF
FF D8 FF E0 00 10 4A 46 49 46 00 01 01 01 00 C0 ÿøÿà..JFIF.....Ä
00 C0 00 00 FF EC 00 11 44 75 63 6B 79 00 01 00 .Ä..ÿì..Ducky...
04 00 00 00 50 00 00 FF E6 00 13 47 6F 50 72 6F ....P..ÿæ..GoPro
00 3C 44 5A 4F 4D 20 3D 20 59 3E 00 FF E1 00 3A .<DZOM = Y>.ÿá.:
45 78 69 66 00 00 4D 4D 00 2A 00 00 00 08 00 03 Exif..MM.*.....
51 10 00 01 00 00 00 01 01 00 00 00 51 11 00 04 Q.....Q...
00 00 00 01 00 00 00 00 51 12 00 04 00 00 00 01 .....Q.....
00 00 00 00 00 00 00 00 FF E8 00 1C 53 50 49 46 .....ÿè..SPIF
46 56 65 72 73 69 6F 6E 32 00 50 72 6F 66 69 6C FVersion2.Profil
65 49 44 3D 34 00 FF E6 00 13 47 6F 50 72 6F 00 eID=4.ÿæ..GoPro.
3C 44 5A 4F 4D 20 3D 20 59 3E 00 FF E7 00 10 48 <DZOM = Y>.ÿç..H
75 61 77 65 69 00 4D 61 74 65 00 38 00 FF E1 00 uawei.Mate.8.ÿá.
3A 45 78 69 66 00 00 4D 4D 00 2A 00 00 00 08 00 :Exif..MM.*.....
03 51 10 00 01 00 00 00 01 01 00 00 00 51 11 00 .Q.....Q..
04 00 00 00 01 00 00 00 00 51 12 00 04 00 00 00 01 .....Q.....
01 00 00 00 00 00 00 00 00 FF EA 00 28 50 68 6F .....ÿê.(Pho
74 6F 53 74 75 64 69 6F E5 A5 97 E5 A8 83 E7 BC toStudioÿ-ÿ`fç¼
9D E5 90 88 EF BC 8C E7 8B 97 E9 83 BD E4 B8 8D .ÿ.^iÿEç<-éfÿä,.
E5 81 9A FF E1 00 3A 45 78 69 66 00 00 4D 4D 00 ÿ.ÿÿá.:Exif..MM.
2A 00 00 00 08 00 03 51 10 00 01 00 00 00 01 01 *......Q.....
00 00 00 51 11 00 04 00 00 00 01 00 00 00 00 51 ...Q.....Q
12 00 04 00 00 00 01 00 00 00 00 00 00 00 00 FF .....ÿ
E5 00 11 53 75 6D 73 75 6E 67 00 42 6F 6D 62 00 ÿ..Sumsung.Bomb.
37 00 FF E3 00 13 4D 45 54 41 00 00 4B 6F 64 61 7.ÿã..META..Koda
6B 00 54 2D 30 33 00 FF EF 00 10 47 72 61 70 68 k.T-03.ÿì..Graph
43 6F 6E 76 00 EA 02 71 00 FF E5 00 11 53 75 6D Conv.ÿ.ÿ.ÿá..Sum
73 75 6E 67 00 42 6F 6D 62 00 37 00 FF ED 00 https://blogosgn.com/q_46280755
50 69 6E 74 6F 73 69 6F 70 20 22 2F 20 00 20 42 Photoshop 2.0.0P
```

```
ctfshow{0c618671a153f5da3948fdb2a2238e44}
```

参考大量的V3博客

v3