ctfshow-Misc入门

原创

H3rmesk1t 于 2021-07-29 03:36:00 发布 1159 收藏 10

分类专栏： Misc 文章标签： Misc ctfshow

本文链接：https://blog.csdn.net/LYJ20010728/article/details/119193793

ctfshow-Misc入门

写在前面

> 后续提取图片中的flag均为脚本提取，部分flag提取出错需要人工再次核验哈~，flag提取演示

图片篇(基础操作)

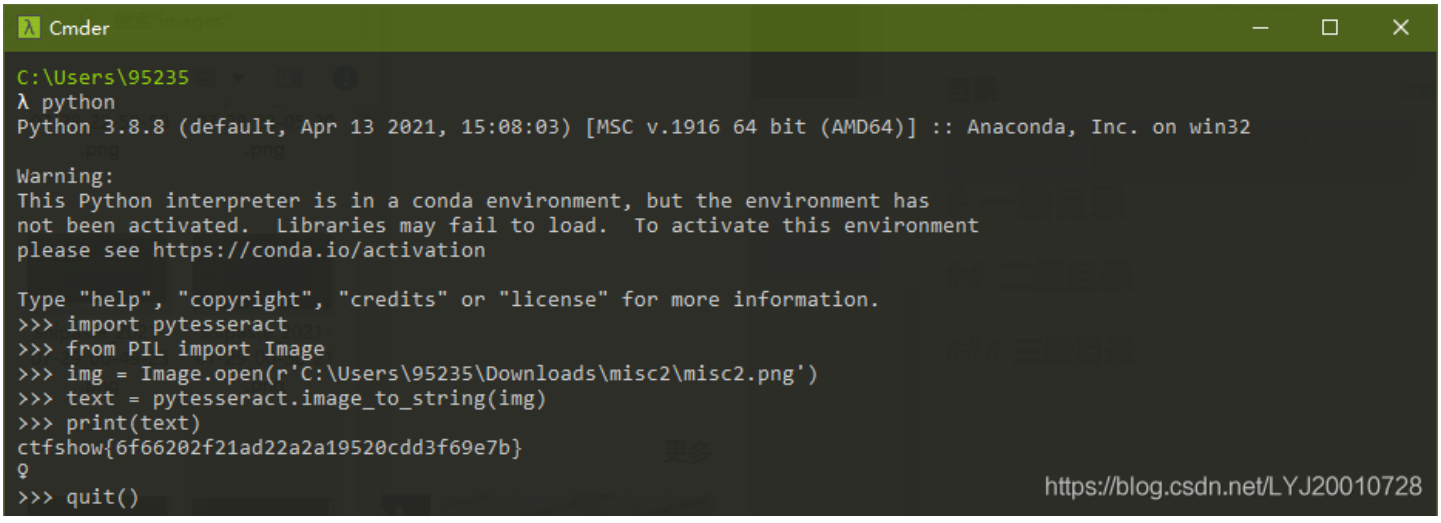
misc1

> flag在下载的图片上

ctfshow{22f1fb91fc4169f1c9411ce632a0ed8d}

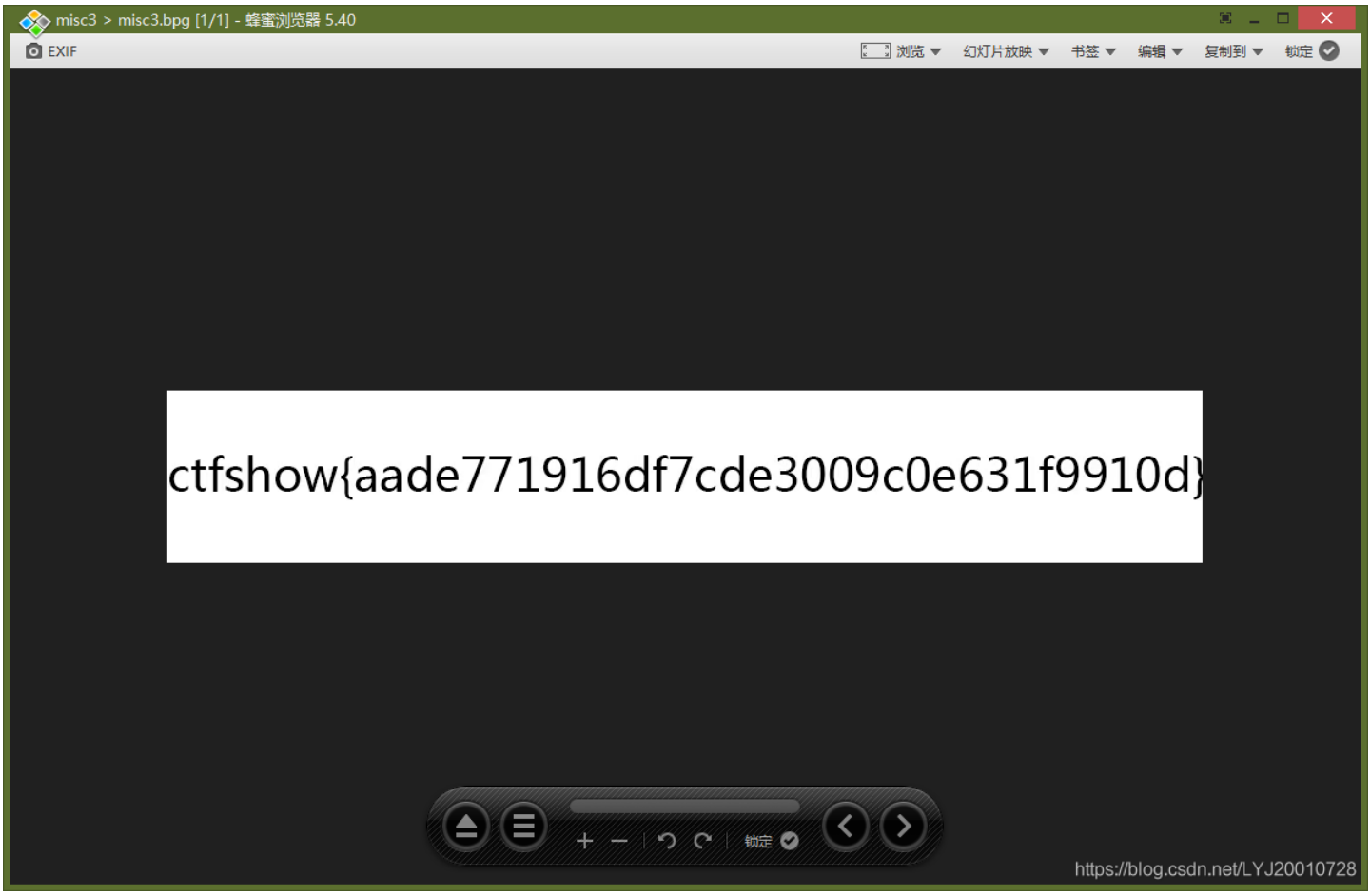
misc2

> 将后缀名改为 `.png` 即可在图片上看到flag



misc3

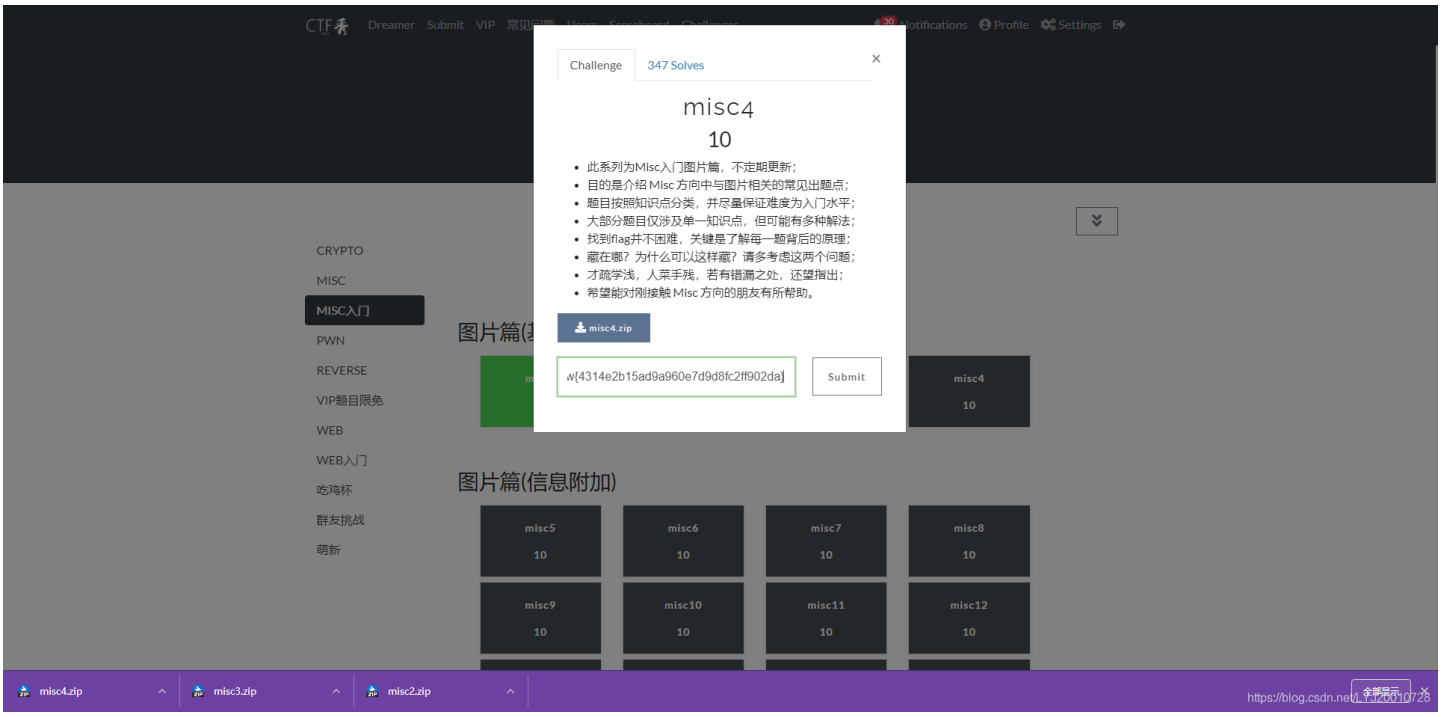
> 推荐一款图片浏览器 `Honeyview`，直接可以查看bpg格式的图片



misc4

> 用 HxD 依次查看文件头，将后缀名依次改为 `.png`、`.jpg`、`.bmp`、`.gif`、`.tif`、`.webp`，将内容拼接起来即可得到flag

名称	修改日期	类型	大小
1.png	2021/2/4 17:25	PNG 文件	6 KB
2.jpg	2021/2/4 17:25	JPG 文件	19 KB
3.bmp	2021/2/4 17:26	BMP 文件	396 KB
4.gif	2021/2/4 17:27	GIF 文件	2 KB
5.tif	2021/2/4 17:28	TIF 文件	418 KB
6.webp	2021/3/25 0:49	Image (webp) File	1 KB



图片篇(信息附加)

misc5

> 用 HxD 打开，拖到尾部即可发现flag

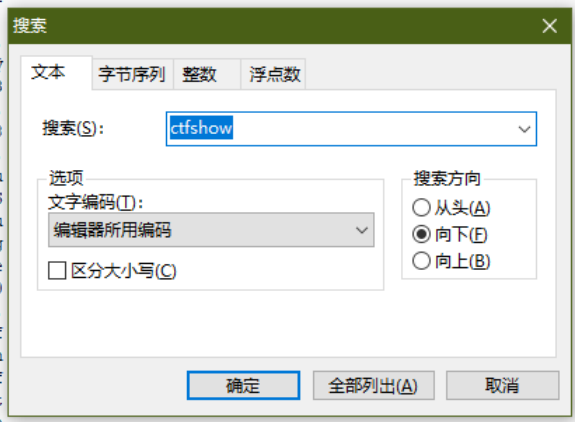
misc5.png

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  对应文本
00000D30  55 39 40 58 D7 5D 32 3D 7F 79 BE 3D 9D AC AC D2  U9@X×]2=.y¾=.¬¬Ò
00000D40  9D 40 EB 09 84 D0 19 E9 8C A1 B9 59 FC D2 42 F3  .@ë.„Ð.éŒ¡¹YüÒBó
00000D50  E9 A0 47 30 04 DE 43 3A 91 48 A3 D0 95 06 BD D3  é G0.ÞC:'H£Ð•.½Ó
00000D60  D2 D5 C1 6F C1 FC 58 DD B4 2C 5C FF 92 EF C7 E1  ÒÕÁoÁüXÝ´,\ÿ'ïÇá
00000D70  AE 67 A5 6B 20 16 02 21 74 4A 36 BD 7B CD BA 62  ®g¥k ..!tJ6½{Í°b
00000D80  00 FB F5 52 D8 FC E4 7A DD A9 A9 E5 15 BF 52 18  .ûõRØüäzÝ©©å.¿R.
00000D90  4C 03 84 A9 45 7B E1 A9 AC 88 99 45 A1 AB 94 9D  L.„©E{á©¬ˆ™E¡«".
00000DA0  00 00 9A 2B 94 1A 68 4A 69 02 80 B6 70 85 10 00  ..š+".hJi.€¶p….. 
00000DB0  68 2E 2D FA FF 8A BB 15 9E 0A E1 0B 83 00 6D E1  h.-úÿŠ».ž.á.ƒ.má
00000DC0  0A 21 00 F0 46 F3 70 7D 92 9B 5C A4 C2 15 41 80  .!.ðFóp}'›\¤Â.A€
00000DD0  B6 12 08 01 00 00 22 E5 96 51 00 00 80 48 09 84  ¶......"å–Q..€H.„
00000DE0  00 00 00 91 12 08 01 00 00 22 25 10 02 00 00 44  ...'......"%....D
00000DF0  4A 20 04 00 00 88 94 40 08 00 00 10 29 81 10 00  J ...ˆ"@....)...
00000E00  00 20 52 02 21 00 00 40 A4 04 42 00 00 80 48 09  . R.!..@¤.B..€H.
00000E10  84 00 00 00 91 12 08 01 00 00 22 25 10 02 00 00  „...'......"%....
00000E20  44 4A 20 04 00 00 88 94 40 08 00 00 10 29 81 10  DJ ...ˆ"@....)..
00000E30  00 00 20 52 02 21 00 00 40 A4 04 42 00 00 80 48  .. R.!..@¤.B..€H
00000E40  09 84 00 00 00 91 12 08 01 00 00 22 25 10 02 00  .„...'......"%...
00000E50  00 44 4A 20 04 00 00 88 94 40 08 00 00 10 29 81  .DJ ...ˆ"@....).
00000E60  10 00 00 20 52 02 21 00 00 40 A4 04 42 00 00 80  ... R.!..@¤.B..€
00000E70  48 09 84 00 00 00 91 12 08 01 00 00 22 25 10 02  H.„...'....."%..
00000E80  00 00 44 4A 20 04 00 00 88 94 40 08 00 00 10 29  ..DJ ...ˆ"@....)
00000E90  81 10 00 00 20 52 02 21 00 00 40 A4 04 42 00 00  .... R.!..@¤.B..
00000EA0  80 48 09 84 00 00 00 91 12 08 01 00 00 22 25 10  €H.„...'....."%.
00000EB0  02 00 00 44 4A 20 04 00 00 88 94 40 08 00 00 10  ...DJ ...ˆ"@....
00000EC0  29 81 10 00 00 20 52 02 21 00 00 40 A4 04 42 00  ).... R.!..@¤.B.
00000ED0  00 80 48 09 84 00 00 00 91 12 08 01 00 00 22 25  .€H.„...'....."%
00000EE0  10 02 00 00 44 4A 20 04 00 00 88 94 40 08 00 00  ....DJ ...ˆ"@...
00000EF0  10 29 81 10 00 00 20 52 02 21 00 00 40 A4 04 42  .).... R.!..@¤.B
00000F00  00 00 80 48 09 84 00 00 00 91 12 08 01 00 00 22  ..€H.„...'....."
00000F10  25 10 02 00 00 44 4A 20 04 00 00 88 52 08 FF 07  %....DJ ...ˆR.ÿ.
00000F20  33 3E 20 BA 99 89 97 04 00 00 00 00 49 45 4E 44  3> °™‰—.....IEND
00000F30  AE 42 60 82 63 74 66 73 68 6F 77 7B 32 61 34 37  ®B`‚ctfshow{2a47
00000F40  36 62 34 30 31 31 38 30 35 66 31 61 38 65 34 62  6b4011805f1a8e4b
00000F50  39 30 36 63 38 66 38 34 30 38 33 65 7D           906c8f84083e}
```


misc6

用 `HxD` 打开，搜索关键词 `ctfshow` 即可发现flag

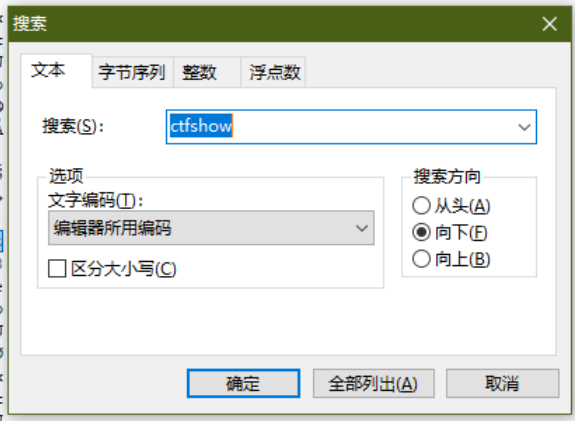
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  对应文本
00000560  FE 72 32 49 29 0E 46 26 2E 4D 07 1F 26 9A EE A0  þr2I).F&.M..&šî
00000570  C4 D5 63 43 99 ED 32 DF 63 C1 6F B5 12 BA EB AA  ÄÕcC™í2ßcÁoµ.ºë ª
00000580  B6 D5 53 43 2B 60 0D 63 1A 00 6B 5A 04 35 AD 68  ¶ÕSC+`.c..kZ.5.h
00000590  FA 2D 6A 92 49 29 03 30 70 AB CA 7E 65 78 F5 33  ú-j'I).0p«Ê~exõ3
000005A0  2A D1 B6 CC 86 B1 A2 C7 0F 6F B5 F6 81 BD DF 41  *Ñ¶Ì†±¢Ç.oµö.½ßA
000005B0  9F E6 29 DF 45 59 15 3A 9B 9B BA B7 44 89 23 83  Ÿæ)ßEY.:››º·D‰#ƒ
000005C0  B8 7B 9B 0E 44 49 25 35 A8 E9 BD 3B 1A DF 5B 1F  ¸{›.DI%5¨é½;.ß[.
000005D0  16 9A 6D D8 2A F5 2B AD AD 77 A6 D0 D6 B2 AD EC  .šmØ*õ+..w¦ÐÖ².ì
000005E0  68 77 A4 C6 D7 5F E8 FF 00 E0 D5 94 92 49 4C 5D  hw¤Æ×_èÿ.àÕ"'IL]
000005F0  5D 6E 73 5E E6 87 39 84 96 38 89 2D 24 6D 3B 7F  ]ns^æ‡9„–8‰-$m;.
00000600  77 DA 54 92 49 25 29 24 92 49 4A 49 24 92 53 FF  wÚT'I%)$'IJI$'Sÿ
00000610  D9 FF ED 0D F2 50 68 6F 74 6F 73 68 6F 70 20 33  Ùÿí.òPhotoshop 3
00000620  2E 30 00 38 42 49 4D 04 25 00 00 00 00 00 10 00  .0.8BIM.%.......
00000630  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 38  ...............8
00000640  42 49 4D 04 3A 00 00 00 00 00 D7 00 00 00 10 00  BIM.:.....×.....
00000650  00 00 01 00 00 00 00 00 0B 70 72 69 6E 74 4F 75  .........printOu
00000660  74 70 75 74 00 00 00 05 00 00 00 00 50 73 74 53  tput........PstS
00000670  62 6F 6F 6C 01 00 00 00 00 00 49 6E 74 65 65 6E  bool.....Inteenu
00000680  6D 00 00 00 00 49 6E 74 65 00 00 00 49 6D 67     m....Inte....Img
00000690  20 00 00 00 0F 63 74 66 73 68 6F 77 7B 64 35 65   ....ctfshow{d5e
000006A0  39 33 37 61 65 66 62 30 39 31 64 33 38 65 37 30  937aefb091d38e70
000006B0  64 39 32 37 62 38 30 65 31 65 32 65 61 7D 00 01  d927b80ele2ea}..
000006C0  00 00 00 00 00 0F 70 72 69 6E 74 50 72 6F 6F 66  ......printProof
000006D0  53 65 74 75 70 4F 62 6A 63 00 00 00 05 68 21 68  SetupObjc....h!h
000006E0  37 8B BE 7F 6E 00 00 00 00 0A 70 72 6F 6F 66  7‹¾.n.....proof
000006F0  53 65 74 75 70 00 00 00 01 00 00 00 00 42 6C 74  Setup........Blt
00000700  6E 65 6E 75 6D 00 00 00 0C 62 75 69 6C 74 69 6E  nenum....builtin
00000710  50 72 6F 6F 66 00 00 00 09 70 72 6F 6F 66 43 4D  Proof....proofCM
00000720  59 4B 00 38 42 49 4D 04 3B 00 00 00 00 02 2D 00  YK.8BIM.;.....-.
00000730  00 00 10 00 00 00 01 00 00 00 00 00 12 70 72 69  .............pri
00000740  6E 74 4F 75 74 70 75 74 4F 70 74 69 6F 6E 73 00  ntOutputOptions.
00000750  00 00 17 00 00 00 00 43 70 74 6E 62 6F 6F 6C 00  .......Cptnbool.
00000760  00 00 00 00 43 6C 62 72 62 6F 6F 6C 00 00 00 00  ....Clbrbool....
00000770  00 52 67 73 4D 62 6F 6F 6C 00 00 00 00 43 72 72  .RgsMbool.....Cr
00000780  6E 43 62 6F 6F 6C 00 00 00 00 43 6E 74 43 62 72  nCbool.....CntCb
00000790  6F 6F 6C 00 00 00 00 00 4C 62 6C 73 62 6F 6F 6C  ool.....Lblsbool
000007A0  00 00 00 00 00 4E 67 74 76 62 6F 6F 6C 00 00 00  .....Ngtvbool...
000007B0  00 00 45 6D 6C 44 62 6F 6F 6C 00 00 00 00 00 49  ..EmlDbool.....I
```



misc7

用 `HxD` 打开，搜索关键词 `ctfshow` 即可发现flag

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  对应文本
00007D20   55 D8 AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55  UØ«±Wb®Å]Š».v*ìU
00007D30   D8 AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8  Ø«±Wb®Å]Š».v*ìUØ
00007D40   AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB  «±Wb®Å]Š».v*ìUØ«
00007D50   B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1  ±Wb®Å]Š».v*ìUØ«±
00007D60   57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57  Wb®Å]Š».v*ìUØ«±W
00007D70   62 AE C5 5D 8A BF FF D1 FB F9 8A BB 15 76 2A EC  b®Å]Š¿ÿÑûùŠ».v*ì
00007D80   55 D8 AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55  UØ«±Wb®Å]Š».v*ìU
00007D90   D8 AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8  Ø«±Wb®Å]Š».v*ìUØ
00007DA0   AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB  «±Wb®Å]Š».v*ìUØ«
00007DB0   B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1  ±Wb®Å]Š».v*ìUØ«±
00007DC0   57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57  Wb®Å]Š».v*ìUØ«±W
00007DD0   62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62  b®Å]Š».v*ìUØ«±Wb
00007DE0   AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE  ®Å]Š».v*ìUØ«±Wb®
00007DF0   C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5  Å]Š».v*ìUØ«±Wb®Å
00007E00   5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D  ]Š».v*ìUØ«±Wb®Å]
00007E10   8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A  Š».v*ìUØ«±Wb®Å]Š
00007E20   BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BB  ».v*ìUØ«±Wb®Å]Š»
00007E30   15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BB 15  .v*ìUØ«±Wb®Å]Š».
00007E40   76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BF 63 74  v*ìUØ«±Wb®Å]Š¿ct
00007E50   66 73 68 6F 77 7B 63 35 65 37 37 63 39 63 32 38  fshow{c5e77c9c28
00007E60   39 32 37 35 65 33 66 33 30 37 33 36 32 65 31 65  9275e3f307362e1e
00007E70   64 38 36 62 62 37 7D 76 2A EC 55 D8 AB B1 57 62  d86bb7}v*ìUØ«±Wb
00007E80   AE C5 5D 8A BF FF D5 FB F9 8A BB 15 76 2A EC 55  ®Å]Š¿ÿÕûùŠ».v*ìU
00007E90   D8 AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8  Ø«±Wb®Å]Š».v*ìUØ
00007EA0   AB B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB  «±Wb®Å]Š».v*ìUØ«
00007EB0   B1 57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1  ±Wb®Å]Š».v*ìUØ«±
00007EC0   57 62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57  Wb®Å]Š».v*ìUØ«±W
00007ED0   62 AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62  b®Å]Š».v*ìUØ«±Wb
00007EE0   AE C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE  ®Å]Š».v*ìUØ«±Wb®
00007EF0   C5 5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5  Å]Š».v*ìUØ«±Wb®Å
00007F00   5D 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D  ]Š».v*ìUØ«±Wb®Å]
00007F10   8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A  Š».v*ìUØ«±Wb®Å]Š
00007F20   BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BB  ».v*ìUØ«±Wb®Å]Š»
00007F30   15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BB 15  .v*ìUØ«±Wb®Å]Š».
00007F40   76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BB 15 76  v*ìUØ«±Wb®Å]Š».v
00007F50   2A EC 55 D8 AB B1 57 62 AE C5 5D 8A BF FF D6 FB  *ìUØ«±Wb®Å]Š¿ÿÖû
00007F60   F9 8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D  ùŠ».v*ìUØ«±Wb®Å]
00007F70   8A BB 15 76 2A EC 55 D8 AB B1 57 62 AE C5 5D 8A  Š».v*ìUØ«±Wb®Å]Š
```



misc8

`binwalk` 查看图片发现隐藏图片，利用 `foremost` 提取出来

```
┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk misc8.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 900 x 150, 8-bit/color RGBA, non-interlaced
91            0x5B            Zlib compressed data, compressed
3892          0xF34           PNG image, 900 x 150, 8-bit/color RGB, non-interlaced
3954          0xF72           Zlib compressed data, default compression


┌──(kali㉿kali)-[~/Desktop]
└─$ foremost misc8.png
Processing: misc8.png
|*|

┌──(kali㉿kali)-[~/Desktop]
└─$
```



```
Cmder

C:\Users\95235
λ pytho
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc8\00000007.png')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
ctfshow{1df0a9a3f709a2605803664b55783687}
♀
>>> quit()
```


misc9

> 用 `zsteg` 查看图片，发现flag

```
┌──(kali㉿kali)-[~/Desktop]
└─$ zsteg misc9.png
meta XML:com.adobe.xmp .. text: "<?xpacket begin=\"\" id=\"W5M0MpCehiHzreSzNTczkc9d\"?> <x:xmpmeta xmlns:x=\"adobe:ns:meta/\" x:
org/1999/02/22-rdf-syntax-ns#\"> <rdf:Description rdf:about=\"\" xmlns:xmp=\"http://ns.adobe.com/xap/1.0/\" xmlns:dc=\"http://p
be.com/xap/1.0/mm/\" xmlns:stEvt=\"http://ns.adobe.com/xap/1.0/sType/ResourceEvent#\" xmp:CreatorTool=\"Adobe Photoshop CC 2019
tadataDate=\"2021-02-24T17:32:07+08:00\" dc:format=\"image/png\" photoshop:ColorMode=\"3\" photoshop:ICCProfile=\"sRGB IEC61966
0ec0-4640-87cb-795628d37d8d\" xmpMM:OriginalDocumentID=\"xmp.did:f9094e65-0ec0-4640-87cb-795628d37d8d\"> <xmpMM:History> <rdf:S
when=\"2021-02-24T17:22:36+08:00\" stEvt:softwareAgent=\"Adobe Photoshop CC 2019 (Windows)\"/> </rdf:Seq> </xmpMM:History> </rd
    00000000: 3c 3f 78 70 61 63 6b 65  74 20 62 65 67 69 6e 3d  |<?xpacket begin=|
    00000010: 22 feff 22 20 69 64 3d 22  57 35 4d 30 4d 70 43 65  |"." id="W5M0MpCe|
    00000020: 68 69 48 7a 72 65 53 7a  4e 54 63 7a 6b 63 39 64  |hiHzreSzNTczkc9d|
    00000030: 22 3f 3e 20 3c 78 3a 78  6d 70 6d 65 74 61 20 78  |"?> <x:xmpmeta x|
    00000040: 6d 6c 6e 73 3a 78 3d 22  61 64 6f 62 65 3a 6e 73  |mlns:x="adobe:ns|
    00000050: 3a 6d 65 74 61 2f 22 20  78 3a 78 6d 70 74 6b 3d  |:meta/" x:xmptk=|
    00000060: 22 41 64 6f 62 65 20 58  4d 50 20 43 6f 72 65 20  |"Adobe XMP Core |
    00000070: 35 2e 36 2d 63 31 34 35  20 37 39 2e 31 36 33 34  |5.6-c145 79.1634|
    00000080: 39 39 2c 20 32 30 31 38  2f 30 38 2f 31 33 2d 31  |99, 2018/08/13-1|
    00000090: 36 3a 34 30 3a 32 32 20  20 20 20 20 20 20 20 22  |6:40:22        "|
    000000a0: 3e 20 3c 72 64 66 3a 52  44 46 20 78 6d 6c 6e 73  |> <rdf:RDF xmlns|
    000000b0: 3a 72 64 66 3d 22 68 74  74 70 3a 2f 2f 77 77 77  |:rdf="http://www|
    000000c0: 2e 77 33 2e 6f 72 67 2f  31 39 39 39 2f 30 32 2f  |.w3.org/1999/02/|
    000000d0: 32 32 2d 72 64 66 2d 73  79 6e 74 61 78 2d 6e 73  |22-rdf-syntax-ns|
    000000e0: 23 22 3e 20 3c 72 64 66  3a 44 65 73 63 72 69 70  |#"> <rdf:Descrip|
    000000f0: 74 69 6f 6e 20 72 64 66  3a 61 62 6f 75 74 3d 22  |tion rdf:about="|
meta Warning        .. text: "ctfshow{5c5e819508a3ab1fd823f11e83e93c75}"

┌──(kali㉿kali)-[~/Desktop]
└─$
```


zsteg (补充)

> zsteg安装方法 (补充)

```
更换RubyGems的源
gem sources --remove https://rubygems.org/
gem sources --add https://gems.ruby-china.com/
gem sources -l
安装zsteg
git clone https://hub.fastgit.org/zed-0xff/zsteg.git
cd zsteg
gem install zsteg
```

> zsteg的使用方法 (常见)

```
查看帮助
zsteg -h

查看LSB信息
zsteg pcat.png

检测zlib
# -b的位数是从1开始的
zsteg zlib.bmp -b 1 -o xy -v

显示细节
zsteg pcat.png -v

尝试所有已知的组合
zsteg pcat.png -a

导出内容
zsteg -E "b1,bgr,lsb,xy" pcat.png > p.exe

更多的使用方法可以查看README.md
```

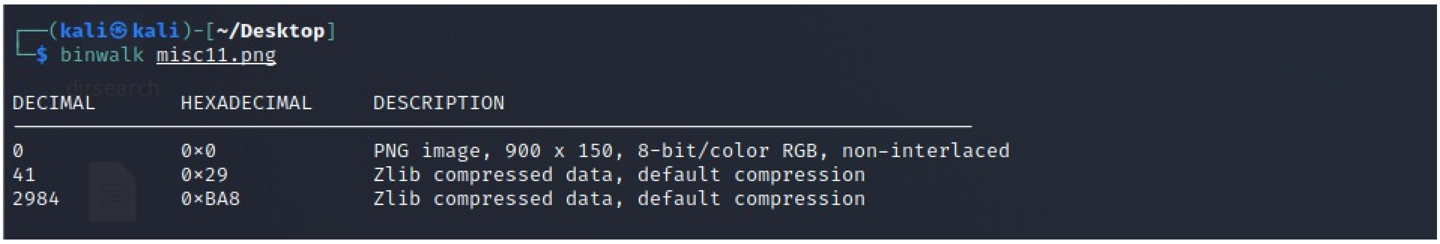
misc10

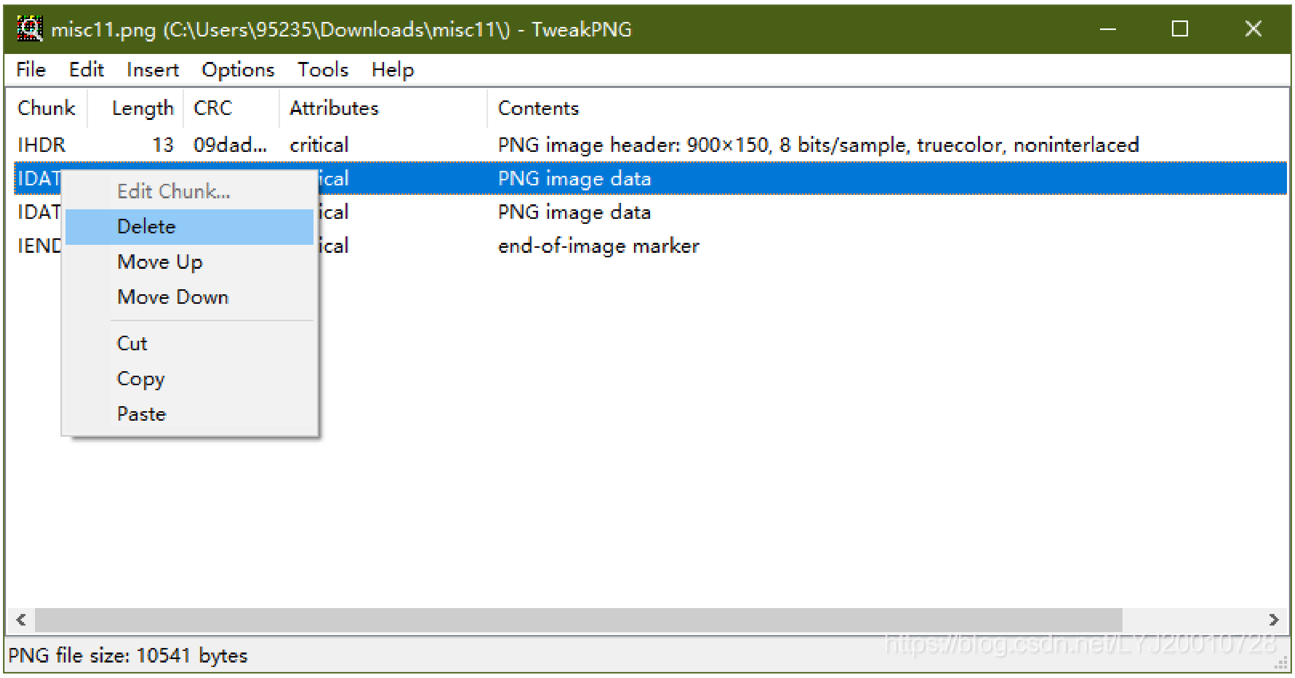
> 用 `binwalk` 查看图片，分离图片，查看数据块即可发现flag，需要注意的是zlib是PNG IDAT的可选压缩格式

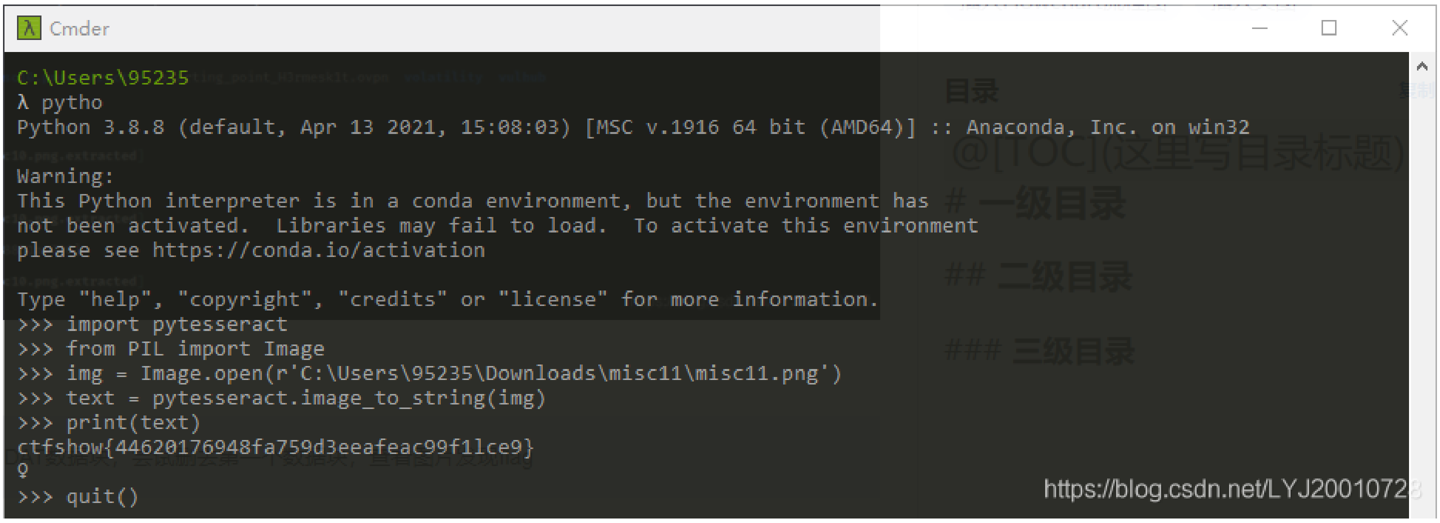
```
┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk misc10.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 900 x 150, 8-bit/color RGB, non-interlaced
1382          0×566           Zlib compressed data, default compression
4325          0×10E5          Zlib compressed data, default compression


┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk -e misc10.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0×0             PNG image, 900 x 150, 8-bit/color RGB, non-interlaced
1382          0×566           Zlib compressed data, default compression
4325          0×10E5          Zlib compressed data, default compression


┌──(kali㉿kali)-[~/Desktop]
└─$ ls
ctf  dirsearch  misc10.png  _misc10.png.extracted  starting_point_H3rmesk1t.ovpn  volatility  vulhub

┌──(kali㉿kali)-[~/Desktop]
└─$ cd _misc10.png.extracted

┌──(kali㉿kali)-[~/Desktop/_misc10.png.extracted]
└─$ ls
10E5  10E5.zlib  566  566.zlib

┌──(kali㉿kali)-[~/Desktop/_misc10.png.extracted]
└─$ cat 10E5
ctfshow{353252424ac69cb64f643768851ac790}

┌──(kali㉿kali)-[~/Desktop/_misc10.png.extracted]
└─$
```


misc11

> binwalk 查看发现两个IDAT数据块，尝试删去第一个数据块，查看图片发现flag

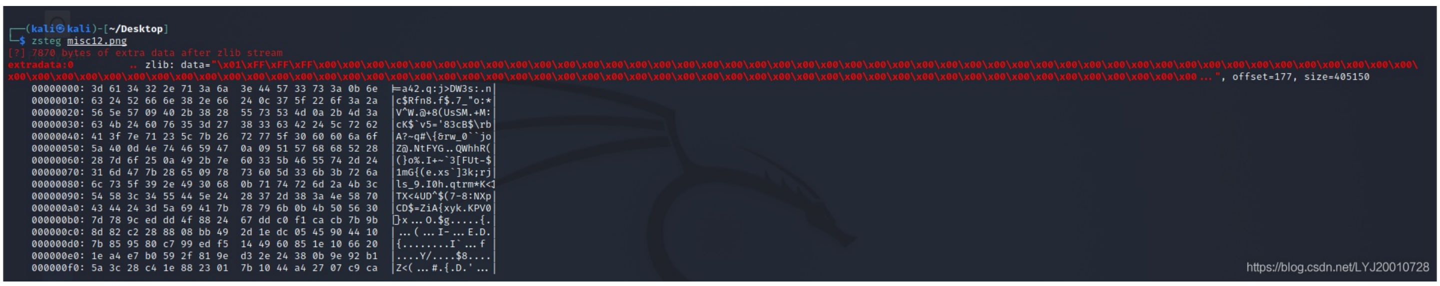




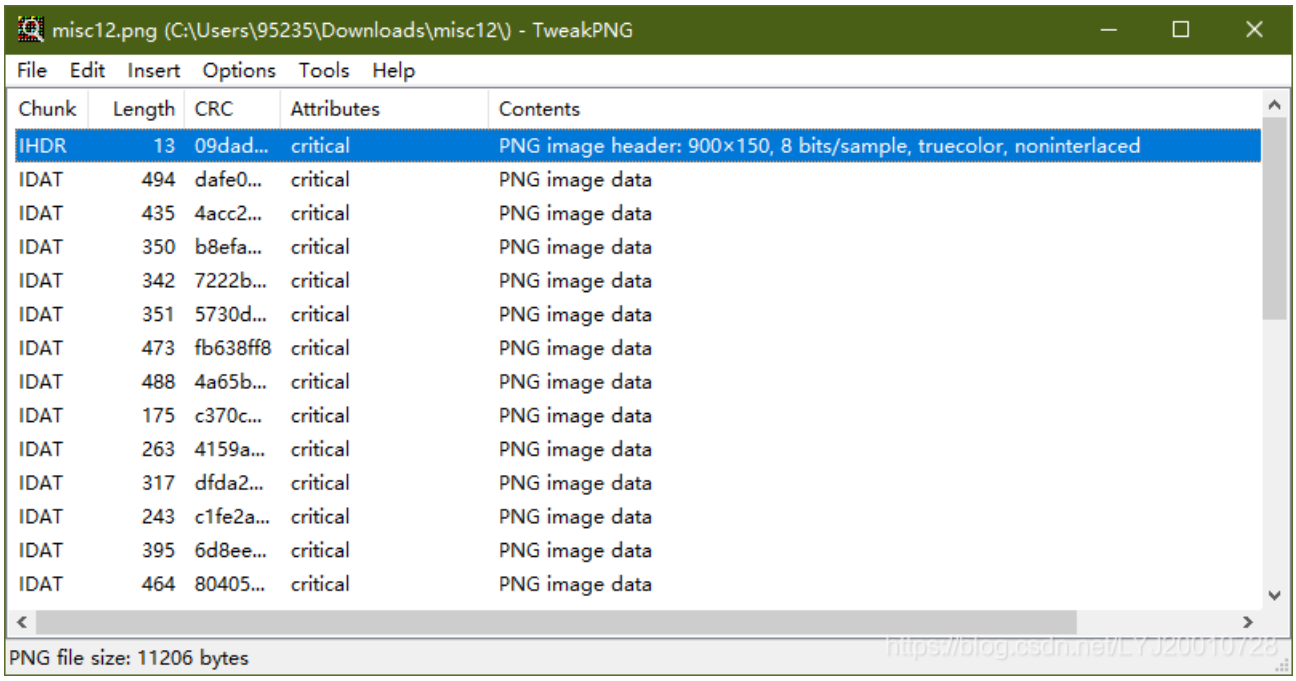


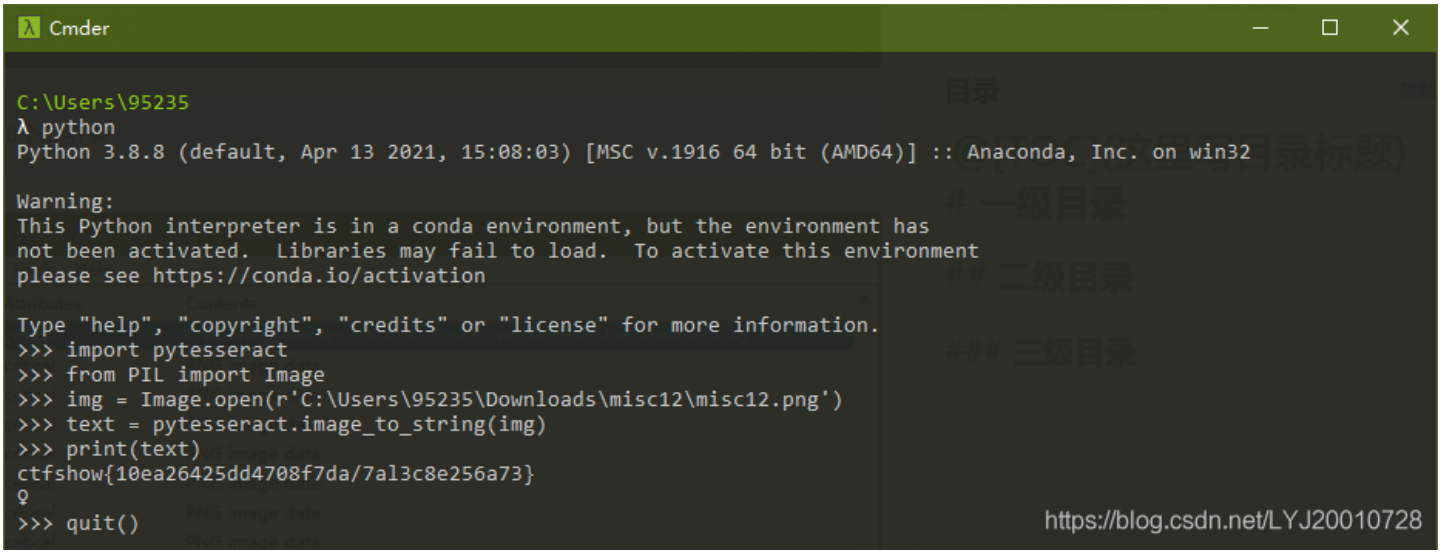
misc12

> zsteg查看图片提示数据块异常



测试后发现需要删掉前8个IDAT块





misc13

`HxD` 查看发现图片尾部存在可疑数据，观察发现 `{` 前面那一串字符从第一位开始每隔一位选取一个字符，连起来就是ctfshow，编写脚本提取flag

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

misc14

`binwalk` 查看图片，发现JPEG图片， `foremost` 和 `binwalk` 无法成功提取，用 `HxD` 打开搜索文件头手动提取

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  对应文本
000007E0   00 00 01 00 06 00 00 01 1A 00 05 00 00 00 01 00  ................
000007F0   00 01 96 01 1B 00 05 00 00 00 01 00 00 01 9E 01  ..–...........ž.
00000800   28 00 03 00 00 00 01 00 02 00 00 02 01 00 04 00  (...............
00000810   00 00 01 00 00 01 A6 02 02 00 04 00 00 00 01 00  ......¦.........
00000820   00 04 D5 00 00 00 00 00 00 00 48 00 00 00 01 00  ..Õ.......H.....
00000830   00 00 48 00 00 00 01 FF D8 FF E0 00 10 4A 46 49  ..H....ÿØÿà..JFI
00000840   46 00 01 01 01 00 78 00 78 00 00 FF DB 00 43 00  F.....x.x..ÿÛ.C.
00000850   02 01 01 02 01 01 02 02 02 02 02 02 02 03 05  ................
00000860   03 03 03 03 03 06 04 04 03 05 07 06 07 07 07 06  ................
00000870   07 07 08 09 0B 09 08 08 0A 08 07 07 0A 0D 0A 0A  ................
00000880   0B 0C 0C 0C 0C 07 09 0E 0F 0D 0C 0E 0B 0C 0C 0C  ................
00000890   FF DB 00 43 01 02 02 02 03 03 03 06 03 03 06 0C  ÿÛ.C............
000008A0   08 07 08 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  ................
000008B0   0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  ................
000008C0   0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C 0C  ................
000008D0   0C 0C 0C 0C 0C FF C0 00 11 08 00 18 01 9C 03 01  .....ÿÀ.....œ..
000008E0   22 00 02 11 01 03 11 01 FF C4 00 1F 00 00 01 05  ".......ÿÄ......
000008F0   01 01 01 01 01 01 00 00 00 00 00 00 00 00 01 02  ................
00000900   03 04 05 06 07 08 09 0A 0B FF C4 00 B5 10 00 02  .........ÿÄ.µ...
00000910   01 03 03 02 04 03 05 05 04 04 00 00 01 7D 01 02  .............}..
00000920   03 00 04 11 05 12 21 31 41 06 13 51 61 07 22 71  ......!1A..Qa."q
00000930   14 32 81 91 A1 08 23 42 B1 C1 15 52 D1 F0 24 33  .2.'¡.#B±Á.RÑð$3
00000940   62 72 82 09 0A 16 17 18 19 1A 25 26 27 28 29 2A  br,.......%&'()*
00000950   34 35 36 37 38 39 3A 43 44 45 46 47 48 49 4A 53  456789:CDEFGHIJS
00000960   54 55 56 57 58 59 5A 63 64 65 66 67 68 69 6A 73  TUVWXYZcdefghijs
00000970   74 75 76 77 78 79 7A 83 84 85 86 87 88 89 8A 92  tuvwxyzƒ„…†‡ˆ‰Š'
00000980   93 94 95 96 97 98 99 9A A2 A3 A4 A5 A6 A7 A8 A9  ""•–—˜™š¢£¤¥¦§¨©
00000990   AA B2 B3 B4 B5 B6 B7 B8 B9 BA C2 C3 C4 C5 C6 C7  ª²³´µ¶·¸¹ºÂÃÄÅÆÇ
000009A0   C8 C9 CA D2 D3 D4 D5 D6 D7 D8 D9 DA E1 E2 E3 E4  ÈÉÊÒÓÔÕÖ×ØÙÚáâãä
000009B0   E5 E6 E7 E8 E9 EA F1 F2 F3 F4 F5 F6 F7 F8 F9 FA  åæçèéêñòóôõö÷øùú
000009C0   FF C4 00 1F 01 00 03 01 01 01 01 01 01 01 01 01  ÿÄ..............
000009D0   00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0A  ................
000009E0   0B FF C4 00 B5 11 00 02 01 02 04 04 03 04 07 05  .ÿÄ.µ...........
000009F0   04 04 00 01 02 77 00 01 02 03 11 04 05 21 31 06  .....w.......!1.
00000A00   12 41 51 07 61 71 13 22 32 81 08 14 42 91 A1 B1  .AQ.aq."2...B'¡±
00000A10   C1 09 23 33 52 F0 15 62 72 D1 0A 16 24 34 E1 25  Á.#3Rð.brÑ..$4á%
00000A20   F1 17 18 19 1A 26 27 28 29 2A 35 36 37 38 39 3A  ñ....&'()*56789:
00000A30   43 44 45 46 47 48 49 4A 53 54 55 56 57 58 59 5A  CDEFGHIJSTUVWXYZ
```

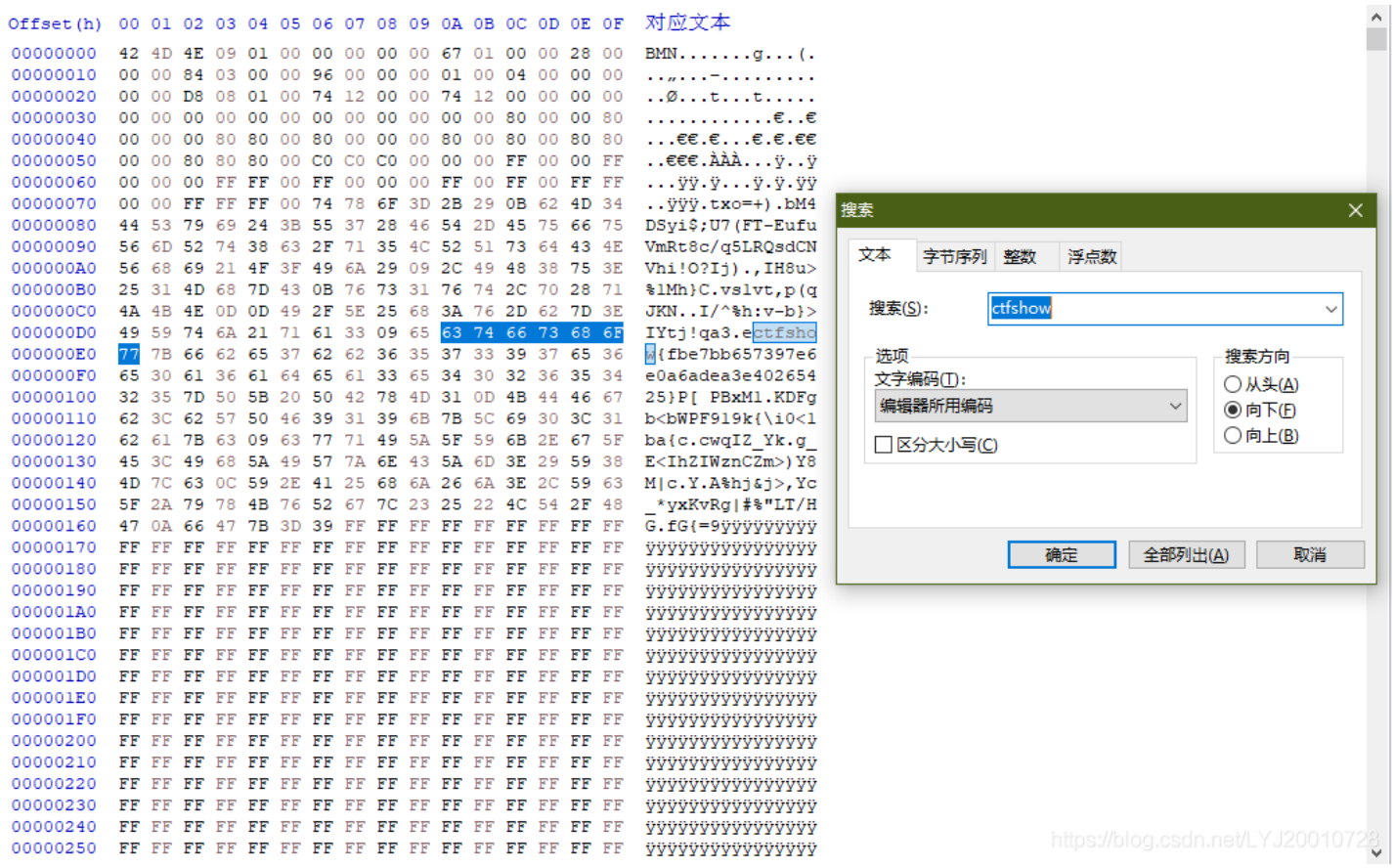


```
C:\Users\95235
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc14\misc.jpg')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
etfshow{ceS28f767fc465b8787cdb936363e6943
ç
>>>
```

python.exe

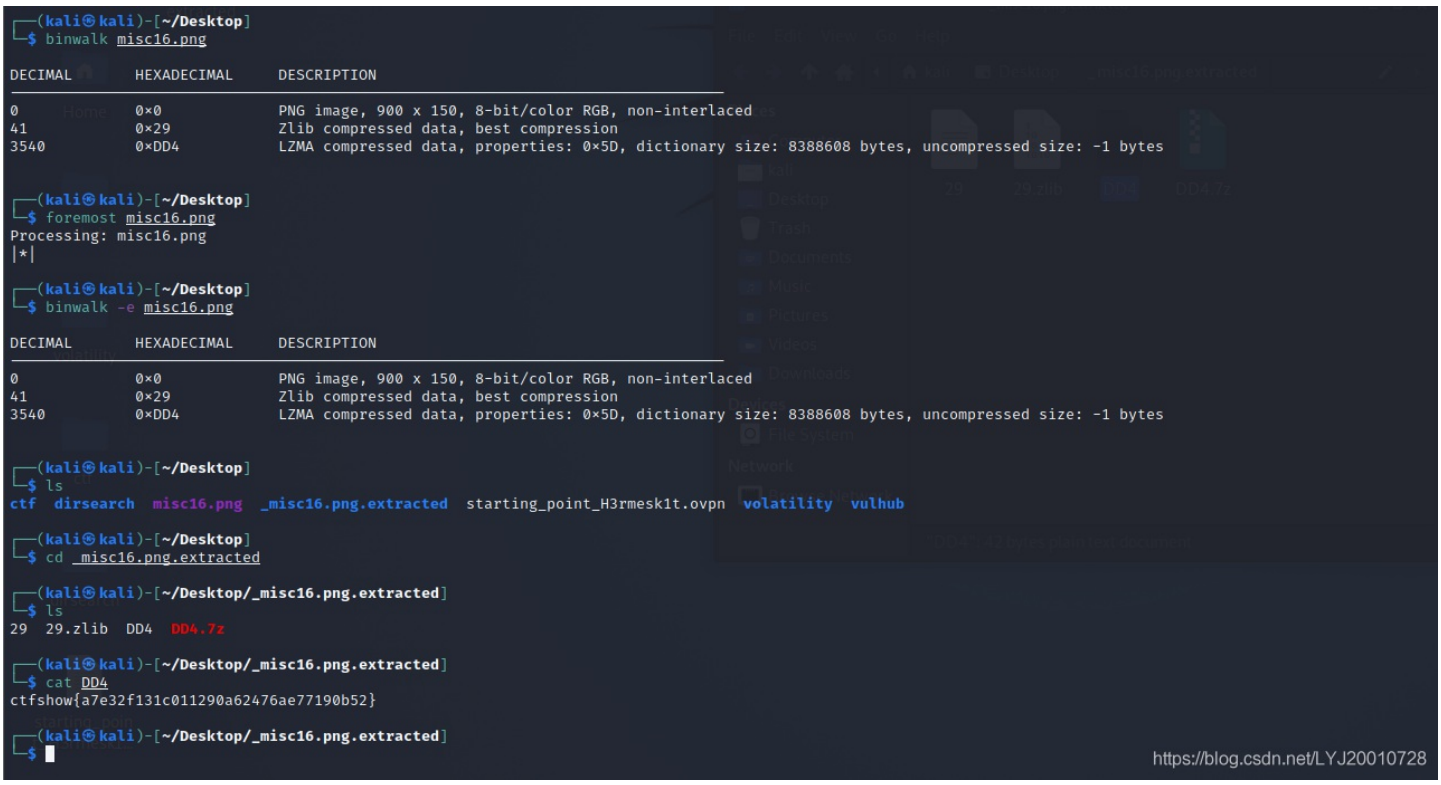
misc15

用 `HxD` 打开搜索关键词 `ctfshow` 即可发现flag



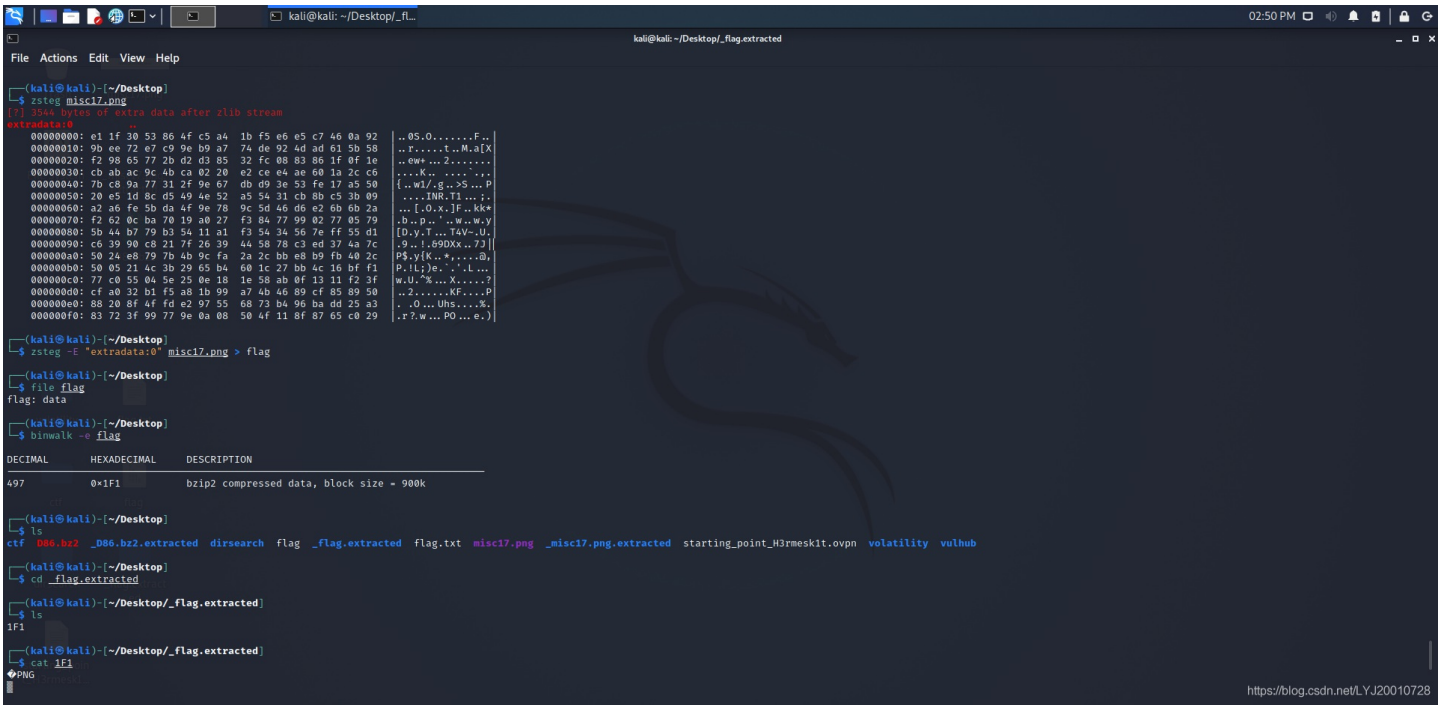
misc16

`binwalk` 查看图片，发现额外数据，用 `binwalk -e` 提取出来，查看提取出来的文件发现flag



misc17

`binwalk` 提取出来的东西解不出，尝试 `zsteg` ，根据提示提取信息得到PNG图片，查看图片发现flag



misc18

> 用 `exiftool` 查看图片，flag在标题、作者、照相机和镜头型号里

```
┌──(kali㉿kali)-[~/Desktop]
└─$ exiftool misc18.jpg
ExifTool Version Number         : 12.16
File Name                       : misc18.jpg
Directory                       : .
File Size                       : 21 KiB
File Modification Date/Time     : 2021:03:13 11:44:41-05:00
File Access Date/Time           : 2021:07:28 14:54:22-04:00
File Inode Change Date/Time     : 2021:07:28 14:54:21-04:00
File Permissions                : rw-------
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 120
Y Resolution                    : 120
Exif Byte Order                 : Big-endian (Motorola, MM)
Camera Model Name               : 28ac17e5f0
Artist                          : 5d60c208f7
XP Title                        : ctfshow{32
XP Author                       : 5d60c208f7
Padding                         : (Binary data 2072 bytes, use -b option to extract)
About                           : uuid:faf5bdd5-ba3d-11da-ad31-d33d75182f1b
Title                           : ctfshow{32
Description                     : ctfshow{32
Creator                         : 5d60c208f7
Warning                         : [minor] Fixed incorrect URI for xmlns:MicrosoftPhoto
Lens Model                      : 2d4cf5a839}
Image Width                     : 900
Image Height                    : 150
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 900x150
Megapixels                      : 0.135
```


misc19

> 用 `exiftool` 查看图片，flag在主机上的文档名里

```
┌──(kali㉿kali)-[~/Desktop]
└─$ exiftool misc19.tif
ExifTool Version Number         : 12.16
File Name                       : misc19.tif
Directory                       : .
File Size                       : 26 KiB
File Modification Date/Time     : 2021:03:24 23:12:27-04:00
File Access Date/Time           : 2021:07:28 14:57:41-04:00
File Inode Change Date/Time     : 2021:07:28 14:57:41-04:00
File Permissions                : rw-------
File Type                       : TIFF
File Type Extension             : tif
MIME Type                       : image/tiff
Exif Byte Order                 : Little-endian (Intel, II)
Subfile Type                    : Full-resolution image
Image Width                     : 900
Image Height                    : 150
Bits Per Sample                 : 8 8 8
Compression                     : LZW
Photometric Interpretation      : RGB
Document Name                   : ctfshow{dfdcf08038cd446a5
Strip Offsets                   : 21688 25422
Orientation                     : Horizontal (normal)
Samples Per Pixel               : 3
Rows Per Strip                  : 97
Strip Byte Counts               : 3733 749
X Resolution                    : 72
Y Resolution                    : 72
Planar Configuration            : Chunky
Resolution Unit                 : inches
Software                        : Adobe Photoshop CC 2019 (Windows)
Modify Date                     : 2021:03:25 10:35:18
Host Computer                   : eb50782f8d3605d}
Predictor                       : Horizontal differencing
XMP Toolkit                     : Adobe XMP Core 5.6-c145 79.163499, 2018/08/13-16:40:22
Creator Tool                    : Adobe Photoshop CC 2019 (Windows)
Create Date                     : 2021:03:13 11:03:03+08:00
Metadata Date                   : 2021:03:25 10:35:18+08:00
Format                          : image/tiff
Color Mode                      : RGB
ICC Profile Name                : sRGB IEC61966-2.1
Instance ID                     : xmp.iid:ae9ae05b-7497-6e4b-8083-763920ef3505
Document ID                     : adobe:docid:photoshop:c214f24b-b22e-c14d-a0f5-91da2f09bb14
Original Document ID            : xmp.did:ff921484-29ad-7544-a030-38f38d997aa5
History Action                  : created, converted, saved
History Instance ID             : xmp.iid:ff921484-29ad-7544-a030-38f38d997aa5, xmp.iid:ae9ae05b-7497-6e4b-8083-763920ef3505
History When                    : 2021:03:13 11:03:03+08:00, 2021:03:25 10:35:18+08:00
History Software Agent          : Adobe Photoshop CC 2019 (Windows), Adobe Photoshop CC 2019 (Windows)
History Parameters              : from image/png to image/tiff
History Changed                 : /
```


misc20

> 用 `exiftool` 查看图片，flag在评论里

```
┌──(kali㉿kali)-[~/Desktop]
└─$ exiftool misc20.jpg
ExifTool Version Number         : 12.16
File Name                       : misc20.jpg
Directory                       : .
File Size                       : 14 KiB
File Modification Date/Time     : 2021:03:24 04:32:48-04:00
File Access Date/Time           : 2021:07:28 14:59:31-04:00
File Inode Change Date/Time     : 2021:07:28 14:59:31-04:00
File Permissions                : rw-------
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
X Resolution                    : 120
Y Resolution                    : 120
Exif Byte Order                 : Big-endian (Motorola, MM)
Comment                         : 这图片也太难看了。来自：西替爱抚秀大括号西九七九六四必一诶易西爱抚零六易一弟七九西二一弟弟诶弟五九三易四二大括号
Image Width                     : 900
Image Height                    : 150
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 900x150
Megapixels                      : 0.135
```

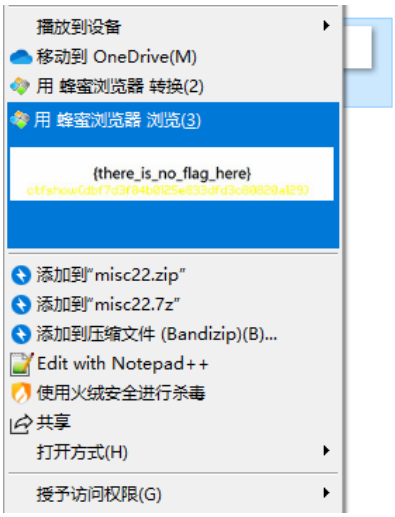

misc21

用 `exiftool` 查看图片，将序列号 `686578285826597329` 转字符得到 `hex(X&Ys)`，分别将 `X/Y Resolution` 和 `X/Y Position` 转成 hex，然后拼接起来，flag为 `ctfshow{e8a221498d5c073b4084eb51b1a1686d}`

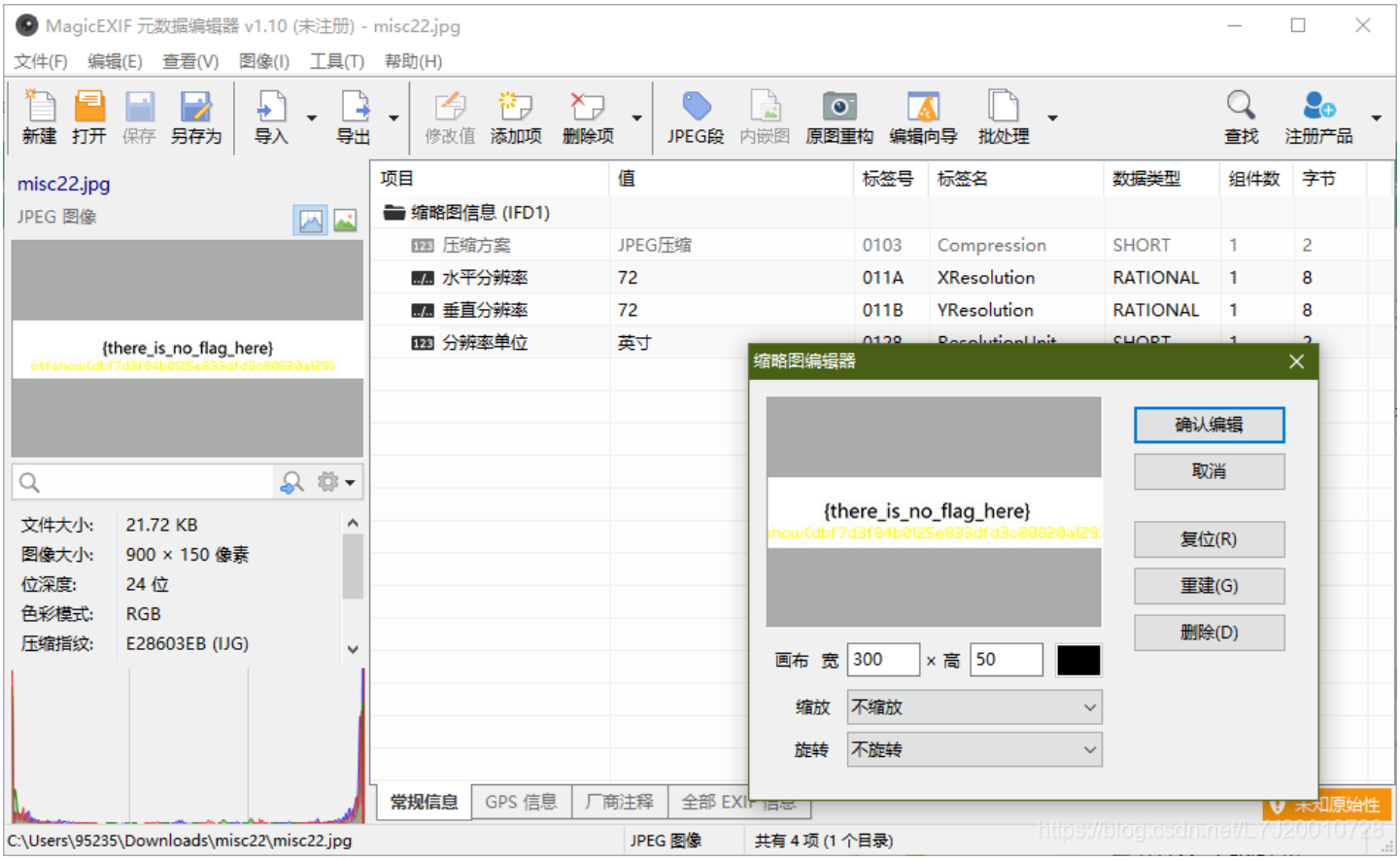
```
┌──(kali㉿kali)-[~/Desktop]
└─$ exiftool misc21.jpg
ExifTool Version Number         : 12.16
File Name                       : misc21.jpg
Directory                       : .
File Size                       : 14 KiB
File Modification Date/Time     : 2021:03:24 12:37:58-04:00
File Access Date/Time           : 2021:07:28 15:00:57-04:00
File Inode Change Date/Time     : 2021:07:28 15:00:57-04:00
File Permissions                : rw-------
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : inches
Exif Byte Order                 : Big-endian (Motorola, MM)
X Resolution                    : 3902939465
Y Resolution                    : 2371618619
Page Name                       : https://ctf.show/
X Position                      : 1082452817
Y Position                      : 2980145261
Target Printer                  : ctfshow{}
Exif Version                    : 0232
Components Configuration        : Y, Cb, Cr, -
Security Classification         : Top Secret
Flashpix Version                : 0100
Color Space                     : Uncalibrated
Serial Number                   : 686578285826597329
Image Width                     : 900
Image Height                    : 150
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 900x150
Megapixels                      : 0.135
```


misc22

直接查看图片没有发现什么，但是用 `Honeyview` 浏览缩略图时发现数据

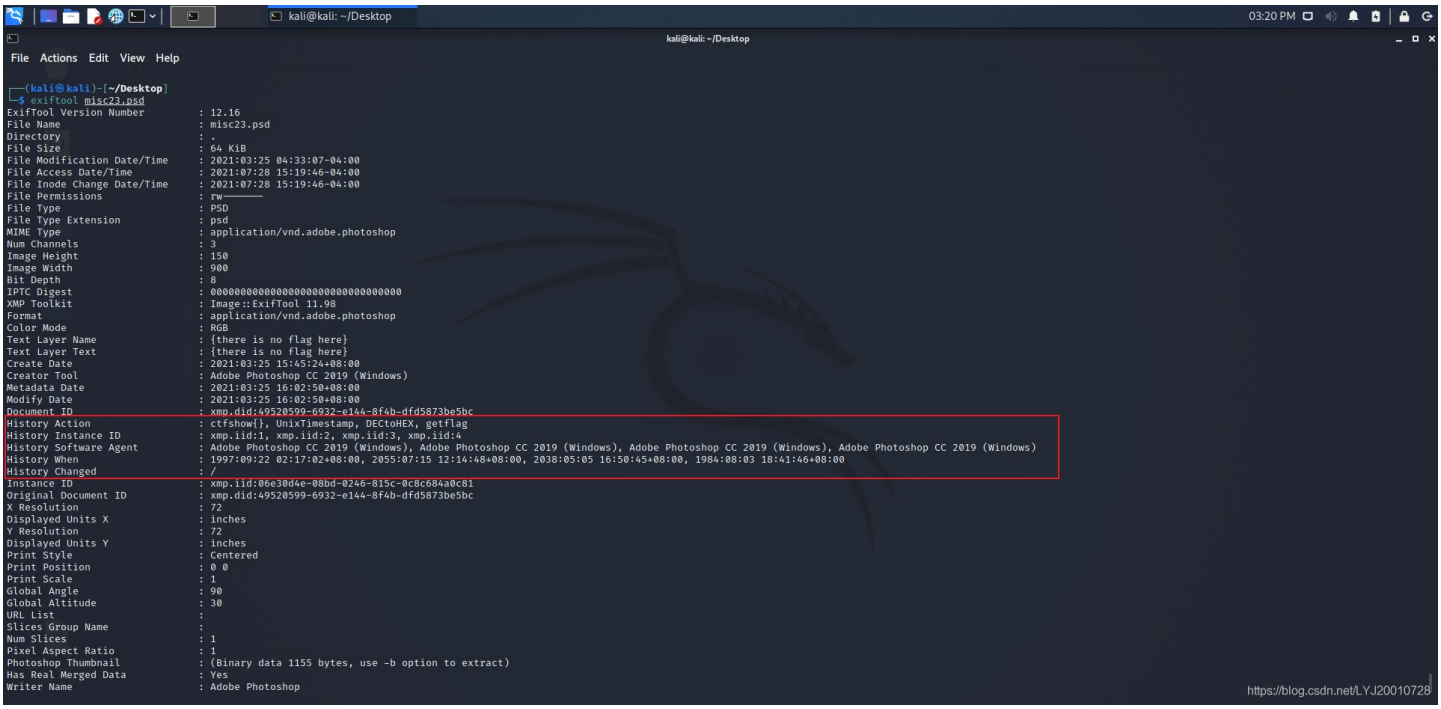


利用 `MagicEXIF` 查看图片，flag为 `ctfshow{dbf7d3f84b0125e833dfd3c80820a129}`

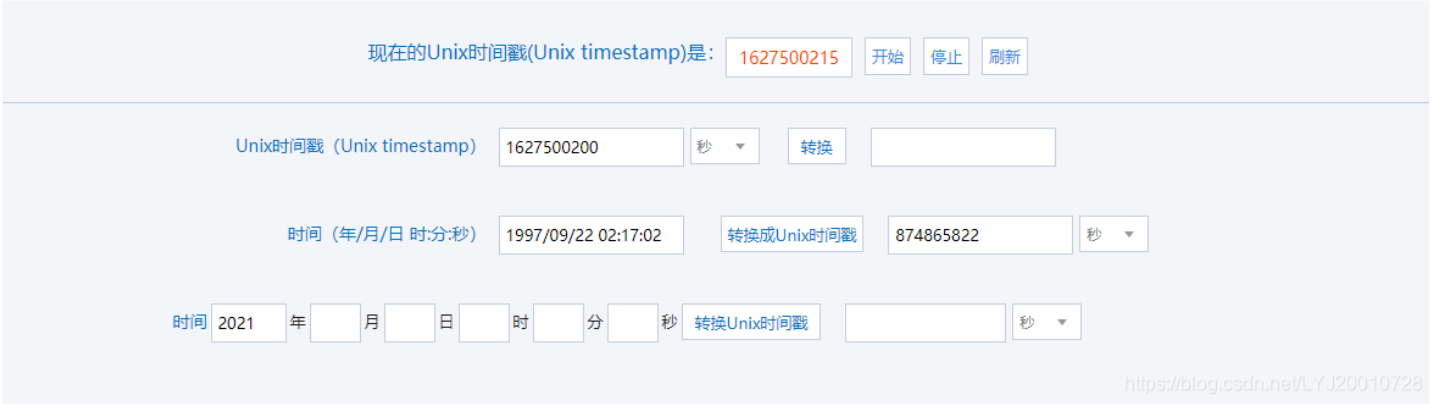


misc23

用 `exiftool` 看一下发现有好几个历史时间， `History Action` 中有提示

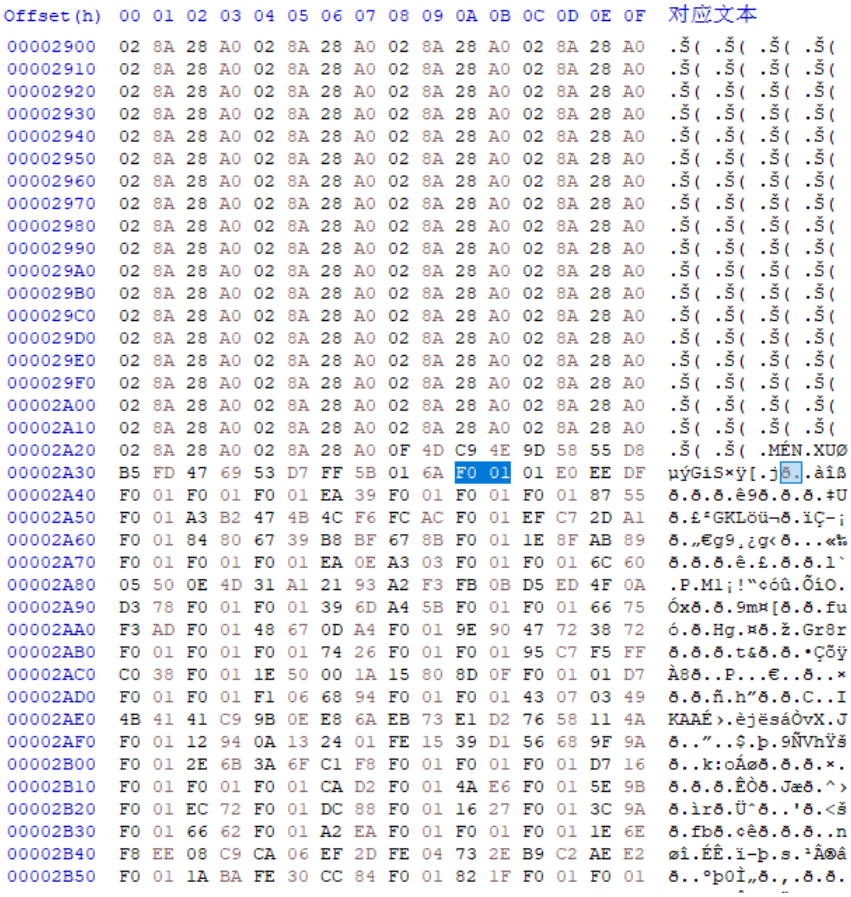


将给出的四个时间的时间戳转换出来，分别hex后拼在一起，转换地址



misc41

提示中的 `F001` 是突破点， `HxD` 查看图片发现有有大量 `F001` 组成了某种形状



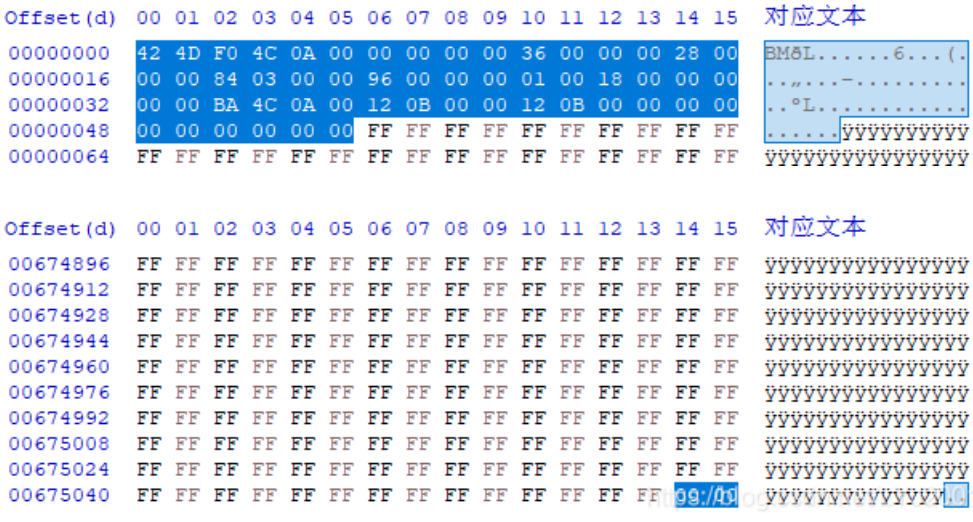
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  对应文本
00002900  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002910  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002920  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002930  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002940  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002950  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002960  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002970  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002980  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002990  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029A0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029B0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029C0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029D0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029E0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
000029F0  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002A00  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002A10  02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0  .Š( .Š( .Š( .Š(
00002A20  02 8A 28 A0 02 8A 28 A0 0F 4D C9 4E 9D 58 55 D8  .Š( .Š( .MÉN.XUØ
00002A30  B5 FD 47 69 53 D7 FF 5B 01 6A F0 01 01 E0 EE DF  µýGiS×ÿ[.jð..àîß
00002A40  F0 01 F0 01 F0 01 EA 39 F0 01 F0 01 F0 01 87 55  ð.ð.ð.ê9ð.ð.ð.‡U
00002A50  F0 01 A3 B2 47 4B 4C F6 FC AC F0 01 EF C7 2D A1  ð.£²GKLöü¬ð.ïÇ-¡
00002A60  F0 01 84 80 67 39 B8 BF 67 8B F0 01 1E 8F AB 89  ð.„€g9¸¿g‹ð...«‰
00002A70  F0 01 F0 01 F0 01 EA 0E A3 03 F0 01 F0 01 6C 60  ð.ð.ð.ê.£.ð.ð.l`
00002A80  05 50 0E 4D 31 A1 21 93 A2 F3 FB 0B D5 ED 4F 0A  .P.M1¡!"¢óû.ÕíO.
00002A90  D3 78 F0 01 F0 01 39 6D A4 5B F0 01 F0 01 66 75  Óxð.ð.9m¤[ð.ð.fu
00002AA0  F3 AD F0 01 48 67 0D A4 F0 01 9E 90 47 72 38 72  ó.ð.Hg.¤ð.ž.Gr8r
00002AB0  F0 01 F0 01 F0 01 74 26 F0 01 F0 01 95 C7 F5 FF  ð.ð.ð.t&ð.ð.•Çõÿ
00002AC0  C0 38 F0 01 1E 50 00 1A 15 80 8D 0F F0 01 01 D7  À8ð..P...€..ð..×
00002AD0  F0 01 F0 01 F1 06 68 94 F0 01 F0 01 43 07 03 49  ð.ð.ñ.h"ð.ð.C..I
00002AE0  4B 41 41 C9 9B 0E E8 6A EB 73 E1 D2 76 58 11 4A  KAAÉ›.èjësáÒvX.J
00002AF0  F0 01 12 94 0A 13 24 01 FE 15 39 D1 56 68 9F 9A  ð..".."$.þ.9ÑVhŸš
00002B00  F0 01 2E 6B 3A 6F C1 F8 F0 01 F0 01 F0 01 D7 16  ð..k:oÁøð.ð.ð.×.
00002B10  F0 01 F0 01 F0 01 CA D2 F0 01 4A E6 F0 01 5E 9B  ð.ð.ð.ÊÒð.Jæð.^›
00002B20  F0 01 EC 72 F0 01 DC 88 F0 01 16 27 F0 01 3C 9A  ð.ìrð.Üˆð..'ð.<š
00002B30  F0 01 66 62 F0 01 A2 EA F0 01 F0 01 F0 01 1E 6E  ð.fbð.¢êð.ð.ð..n
00002B40  F8 EE 08 C9 CA 06 EF 2D FE 04 73 2E B9 C2 AE E2  øî.ÉÊ.ï-þ.s.¹Â®â
00002B50  F0 01 1A BA FE 30 CC 84 F0 01 82 1F F0 01 F0 01  ð..ºþ0Ì„ð.‚.ð.ð.
```


把 `F001` 出现过的位置中所有十六进制的值单独截取出来，每四位分隔开，把F001替换成0，其他值替换成空格，得到一张含有flag的图片，这里也可以采用 `CyberChef` 来解决问题，flag为 `ctfshow{fcbd427caf4a52f1147ab44346cd1cdd}`

图片篇(文件结构)

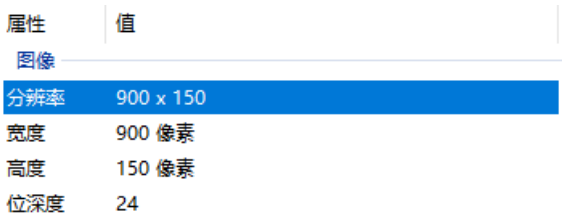
misc24

用 `HxD` 查看一下图片，文件头占了53个字节，文件尾的位置在675053字节处 (后面两个字节是windows的"补0")，因为每个像素点由3个字节 (十六进制码6位) 表示，每个字节负责控制一种颜色，分别为蓝（Blue）、绿（Green）、红（Red），所以文件真实的像素大小为： `(675053-53)/3=225000`

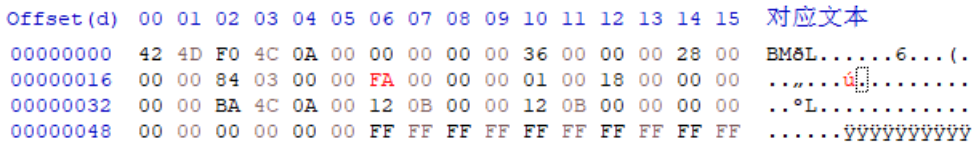


```
Offset(d)  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  对应文本
00000000  42 4D F0 4C 0A 00 00 00 00 00 36 00 00 00 28 00  BMðL......6...(.
00000016  00 00 84 03 00 00 96 00 00 00 01 00 18 00 00 00  ..„...–.........
00000032  00 00 BA 4C 0A 00 12 0B 00 00 12 0B 00 00 00 00  ..ºL............
00000048  00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF  ......ÿÿÿÿÿÿÿÿÿÿ
00000064  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ

Offset(d)  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15  对应文本
00674896  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674912  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674928  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674944  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674960  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674976  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00674992  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00675008  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00675024  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00675040  FF FF FF FF FF FF FF FF FF FF FF FF FF FF 99 A9  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ™©
```


题目给的图片是 `900*150=135000` 个像素大小



尝试后发现这题的宽度是对的，所以正确的高度是 `225000/900=250`，将高度改成

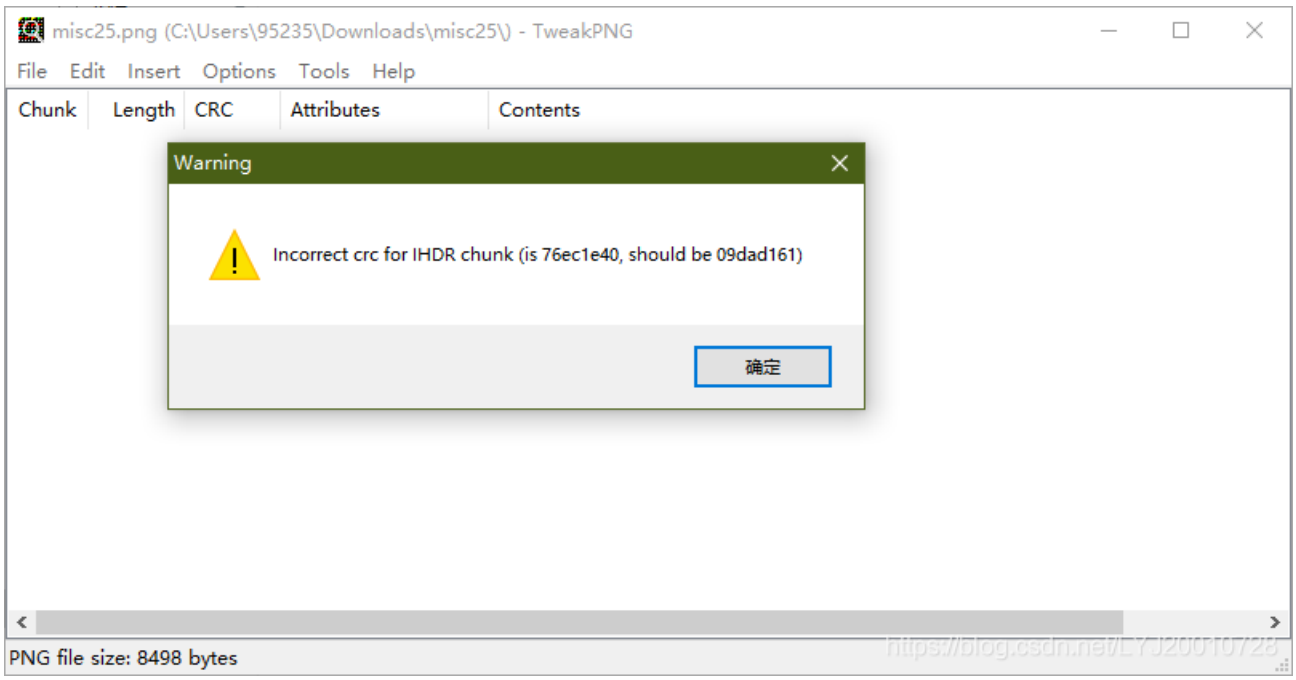


ctfshow{dd7d8bc9e5e873eb7da3fa51d92ca4b7}

{there_is_no_flag_here}

misc25

用 `TweakPNG` 查看图片发现图片的CRC值不对，猜测应该是修改了宽高，用脚本跑一下看看



```
C:\Users\95235\Desktop
λ python exp.py
宽为：bytearray(b'\x00\x00\x03\x84')
高为：bytearray(b'\x00\x00\x00\xfa')

C:\Users\95235\Desktop
λ
```

cmd.exe

根据脚本计算出来的值修改宽高，保存后即可看到flag

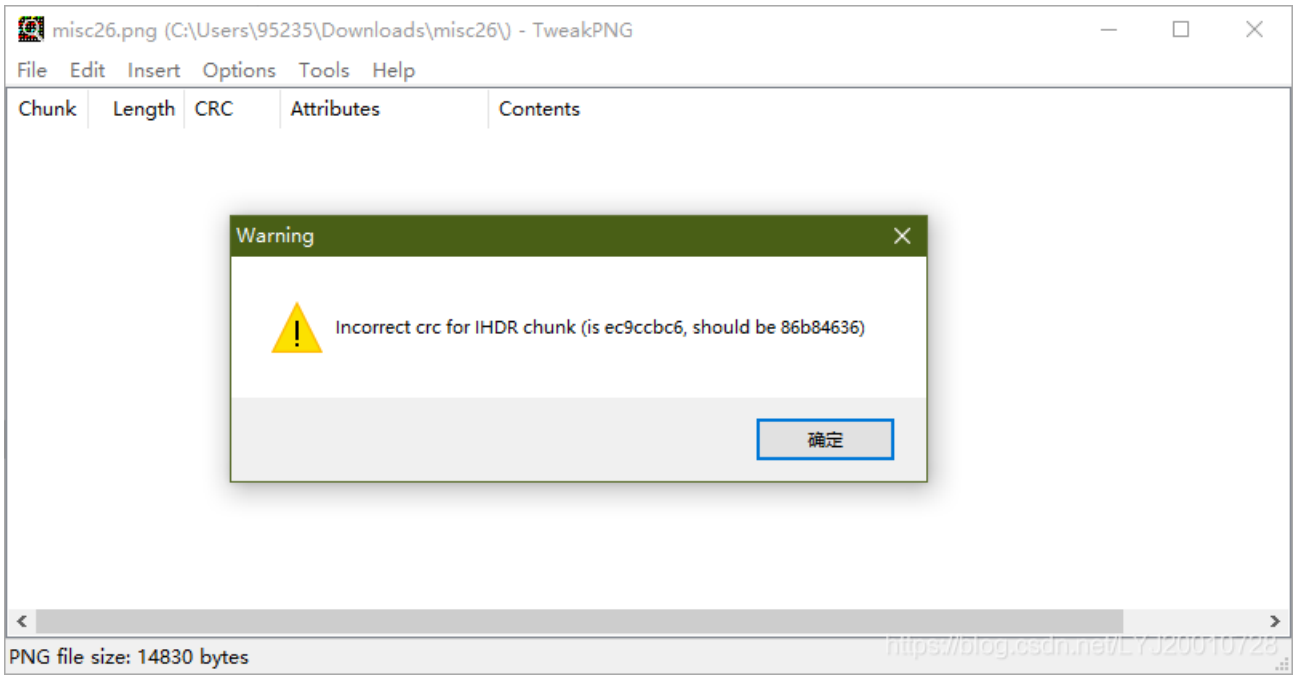
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	对应文本
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG........IHDR
00000010	00	00	03	84	00	00	00	FA	08	02	00	00	00	76	EC	1E	...„...ú....vì.
00000020	40	00	00	20	F9	49	44	41	54	78	DA	ED	DD	D9	95	E4	@.. ùIDATxÚíÝÙ•ä
00000030	36	B6	05	50	59	23	43	CA	0A	F9	20	1B	64	82	3C	90	6¶.PY#CÊ.ù .d‚<.
00000040	05	65	41	39	50	06	C8	00	19	A0	FF	FA	CF	87	D5	5C	.eA9P.È.. ÿúÏ‡Õ\

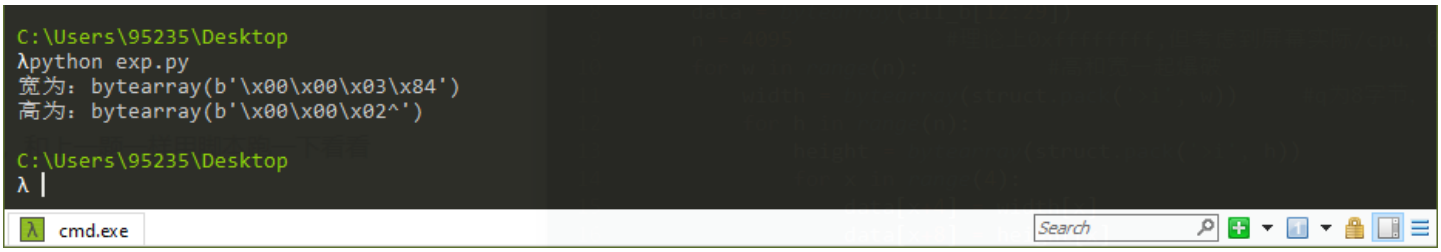
```
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc25\misc25.png')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
{there_is_no_flag_here}

ctfshow{494f611¢cc5842dd597f460874ce38f57}
ø
>>>
```

python.exe

misc26

用 TweakPNG 查看图片发现图片的CRC值不对，和上一题一样用脚本跑一下看看





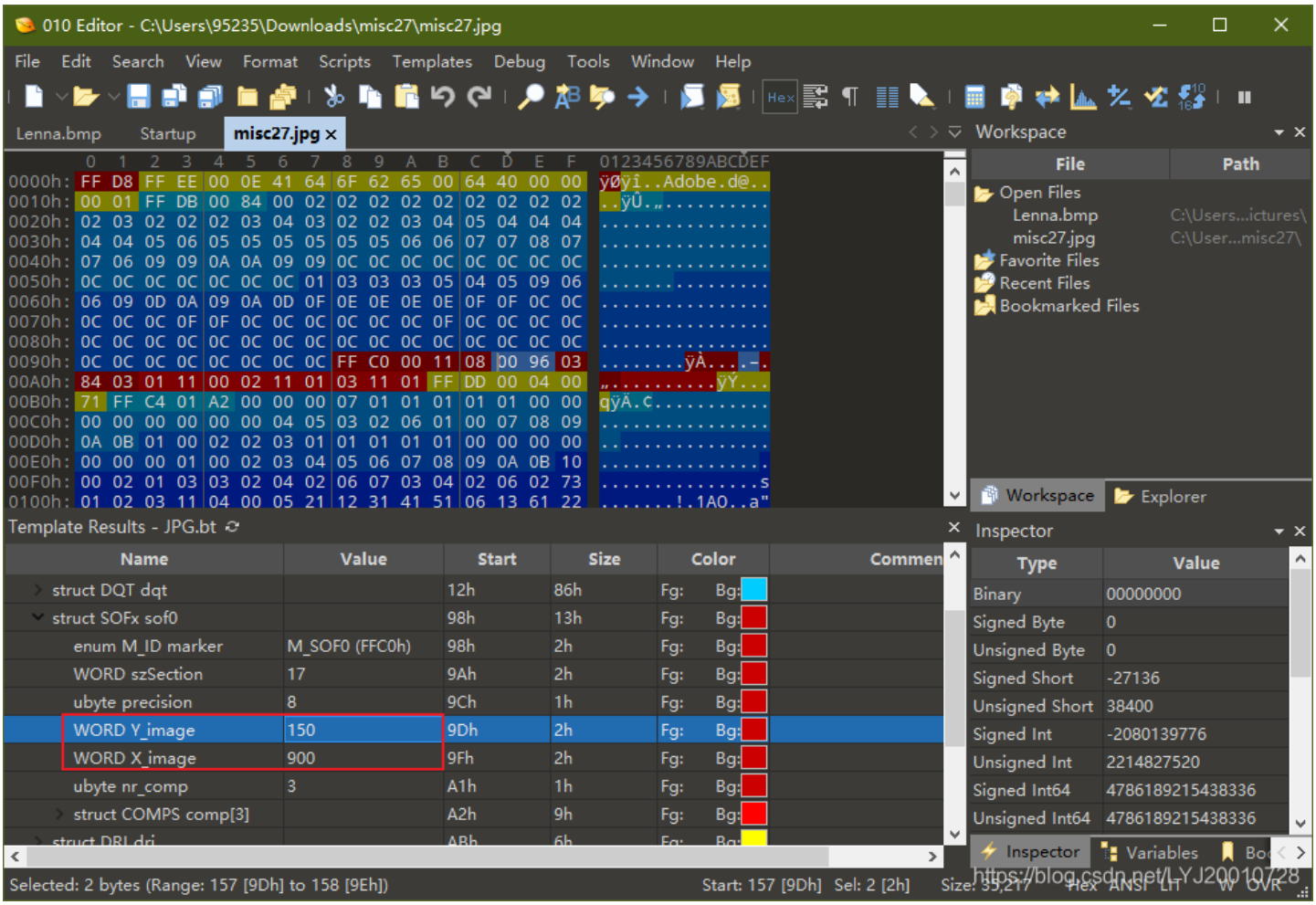
根据脚本计算出来的值修改宽高，保存后即可看到flag

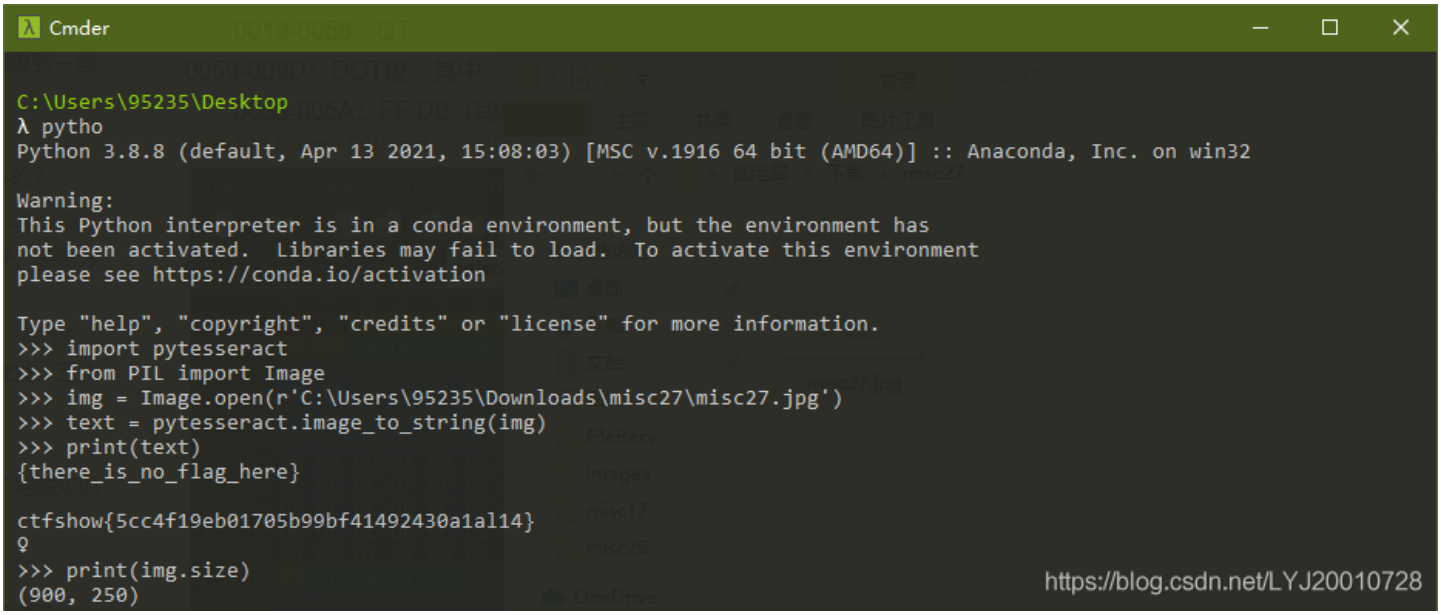
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F   对应文本
00000000   89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52   %PNG........IHDR
00000010   00 00 03 84 00 00 02 5E 08 06 00 00 00 EC 9C CB   ...„...^.....ìœË
00000020   C6 00 00 39 B5 49 44 41 54 78 DA ED DD C1 B1 EA   Æ..9µIDATxÚíÝÁ±ê
00000030   4C 62 36 60 A5 40 0A 2C 9C 00 19 B8 48 81 2A 47   Lb6`¥@.,œ..¸H.*G
00000040   C0 C6 01 B0 F2 9E 8D 03 38 55 13 01 11 B8 8A B5   ÀÆ.°òž..8U...¸Šµ
00000050   77 6C 1C 00 01 78 C3 CA FB CF A3 99 CB 8C BE BE   wl...xÃÊûÏ£™ËŒ¾¾
00000060   2D E8 6E 75 0B 01 CF 5B F5 D4 FF 7B BE 7B 38 3A   -ènu..Ï[õÔÿ{¾{8:
```

```
Cmder
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc26\misc26.png')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
{there_is_no_flag_here}

ctfshow{94aef1
+True height(hex) of this picture+
087a7ccf2e28e742efd704c}
Q
>>> print(img.size)
(900, 606)
>>> print(hex(606))
0x25e
>>>
python.exe
Search
```

misc27

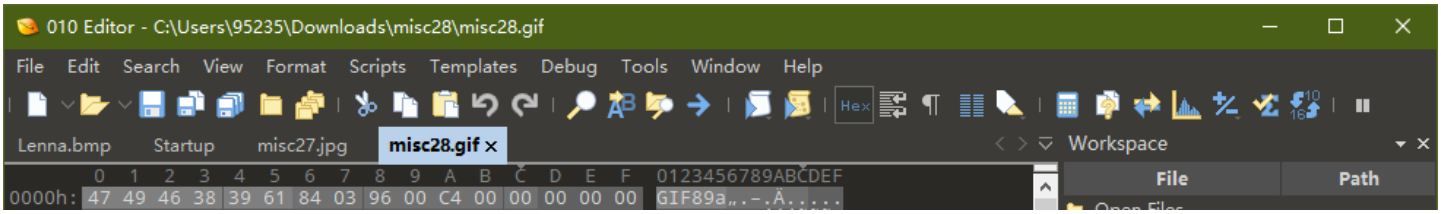
> 根据提示，猜测依旧是修改图片高度，将高度改高后即可发现flag

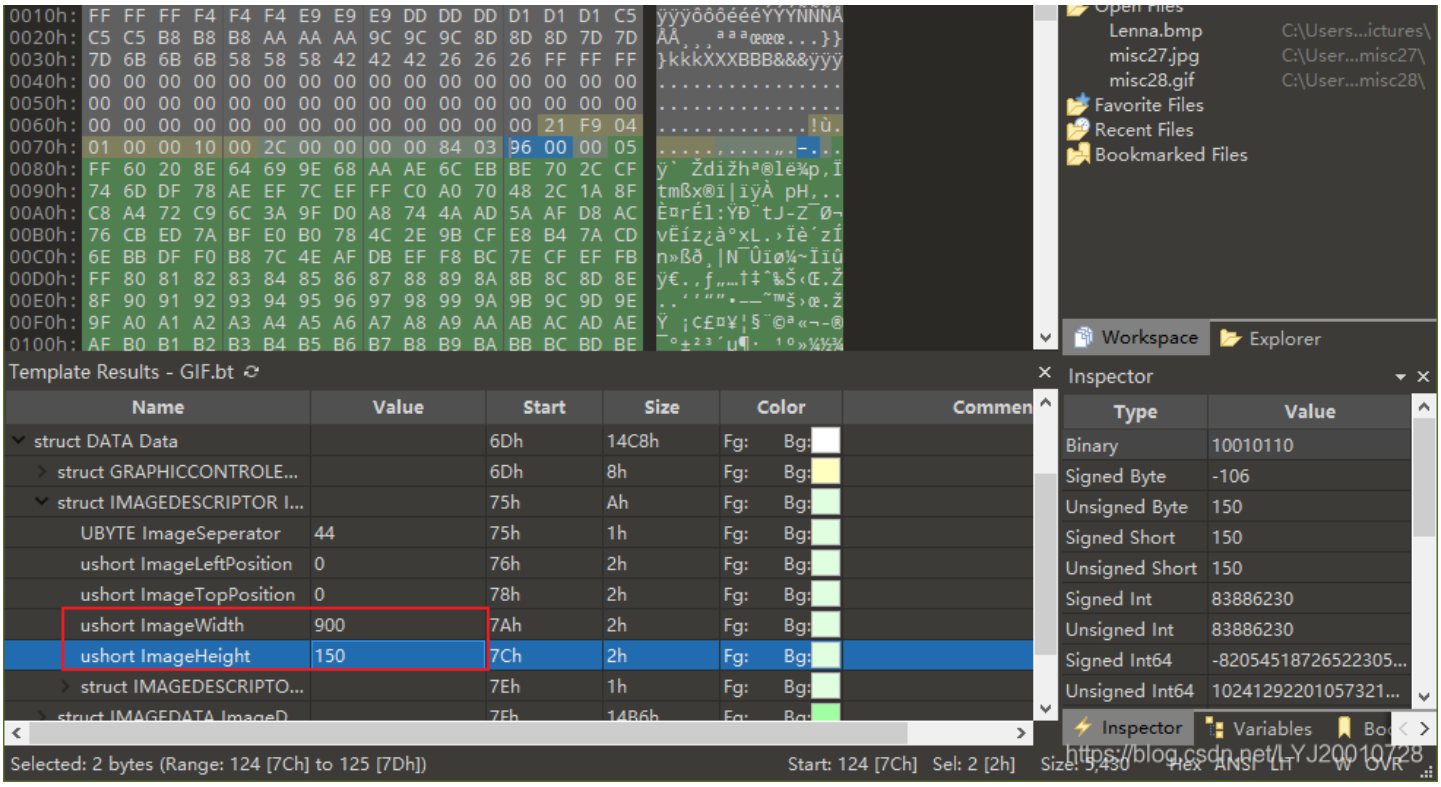


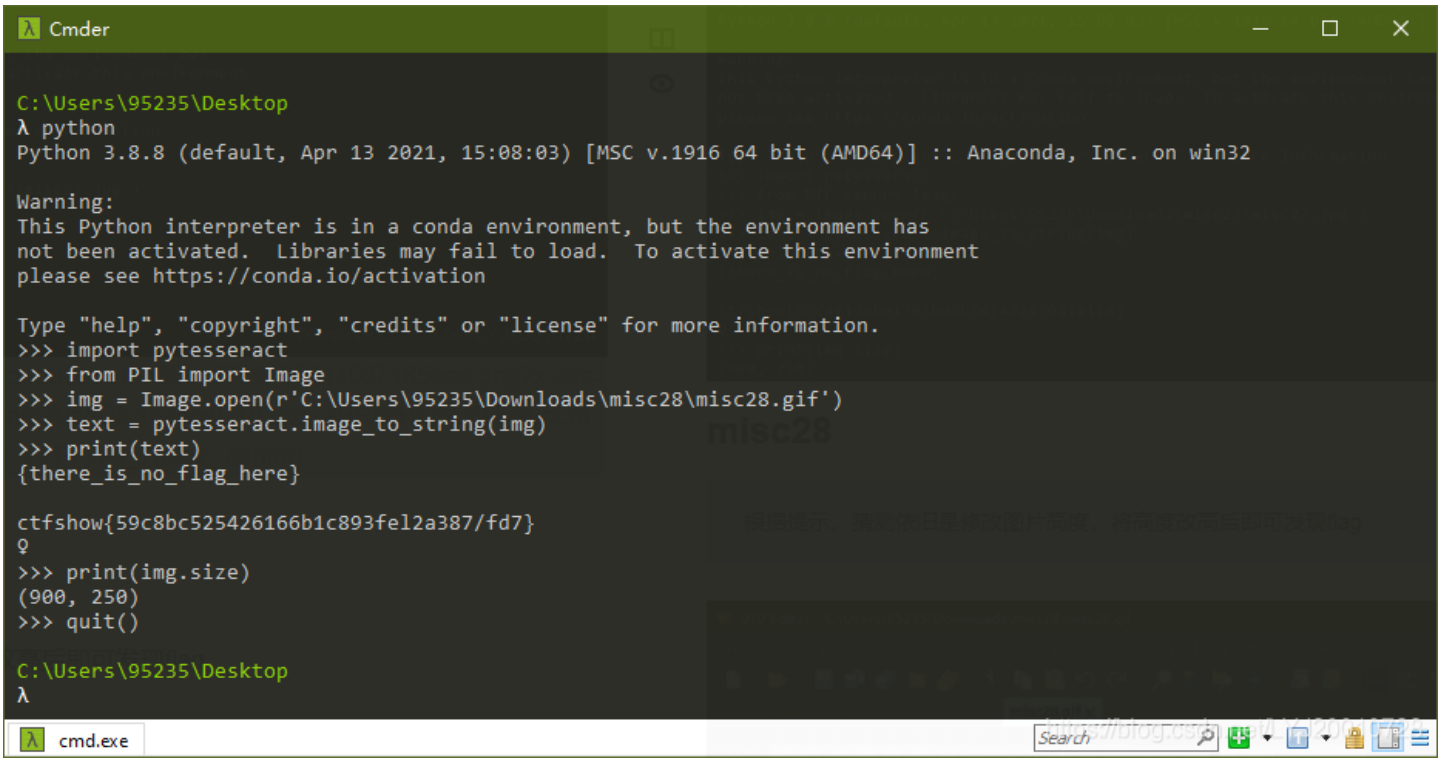


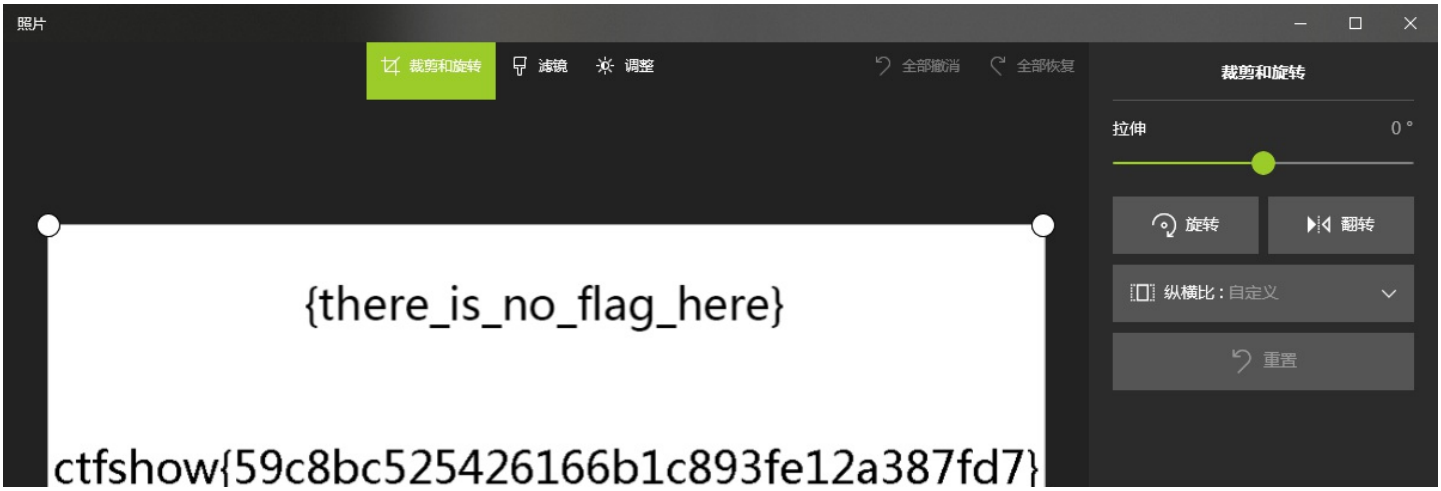
misc28

根据提示，猜测依旧是修改图片高度，将高度改高后即可发现flag，但是需要注意从预览图中能看到flag，但是直接打开看不到，可以使用图片编辑器或者Stegsolve打开



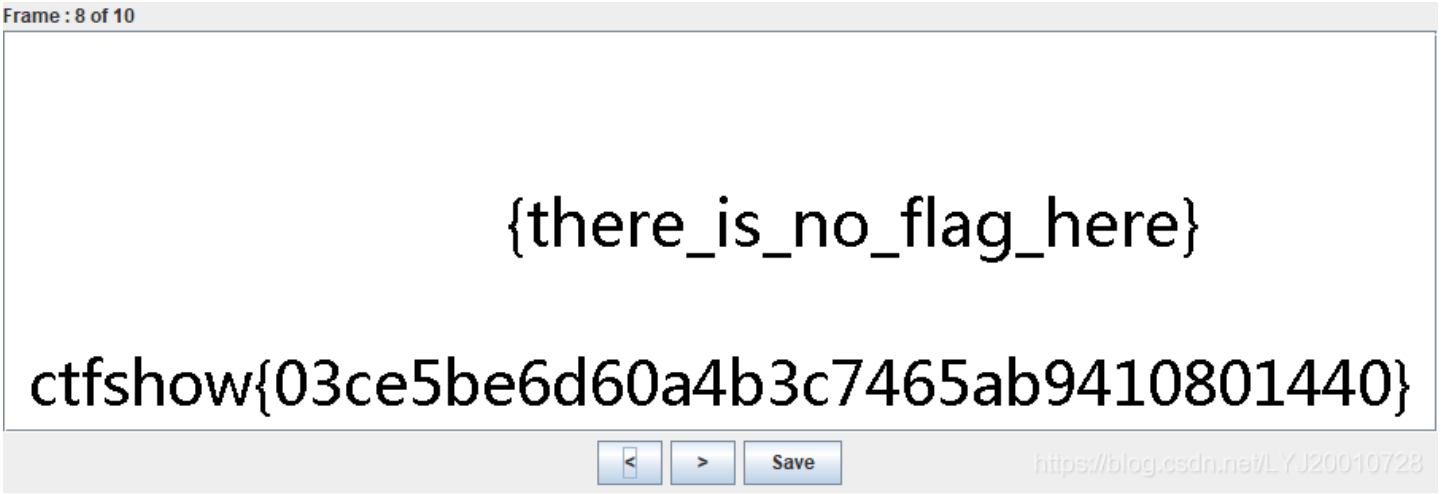

Open Files
Lenna.bmp C:\Users...ictures\
misc27.jpg C:\User...misc27\
misc28.gif C:\User...misc28\
Favorite Files
Recent Files
Bookmarked Files
Workspace
Explorer
Template Results - GIF.bt
Name
Value
Start
Size
Color
Comment
struct DATA Data 6Dh 14C8h
struct GRAPHICCONTROLE... 6Dh 8h
struct IMAGEDESCRIPTOR I... 75h Ah
UBYTE ImageSeperator 44 75h 1h
ushort ImageLeftPosition 0 76h 2h
ushort ImageTopPosition 0 78h 2h
ushort ImageWidth 900 7Ah 2h
ushort ImageHeight 150 7Ch 2h
struct IMAGEDESCRIPTO... 7Eh 1h
Inspector
Type
Value
Binary 10010110
Signed Byte -106
Unsigned Byte 150
Signed Short 150
Unsigned Short 150
Signed Int 83886230
Unsigned Int 83886230
Signed Int64 -82054518726522305...
Unsigned Int64 10241292201057321...
Inspector
Variables
Selected: 2 bytes (Range: 124 [7Ch] to 125 [7Dh])
Start: 124 [7Ch] Sel: 2 [2h]
https://blog.csdn.net/LYJ20010728


Cmder
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation
Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc28\misc28.gif')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
{there_is_no_flag_here}
ctfshow{59c8bc525426166b1c893fe12a387fd7}
>>> print(img.size)
(900, 250)
>>> quit()
C:\Users\95235\Desktop
λ
cmd.exe


照片
裁剪和旋转
滤镜
调整
全部撤消
全部恢复
裁剪和旋转
拉伸
0°
旋转
翻转
纵横比：自定义
重置
{there_is_no_flag_here}
ctfshow{59c8bc525426166b1c893fe12a387fd7}

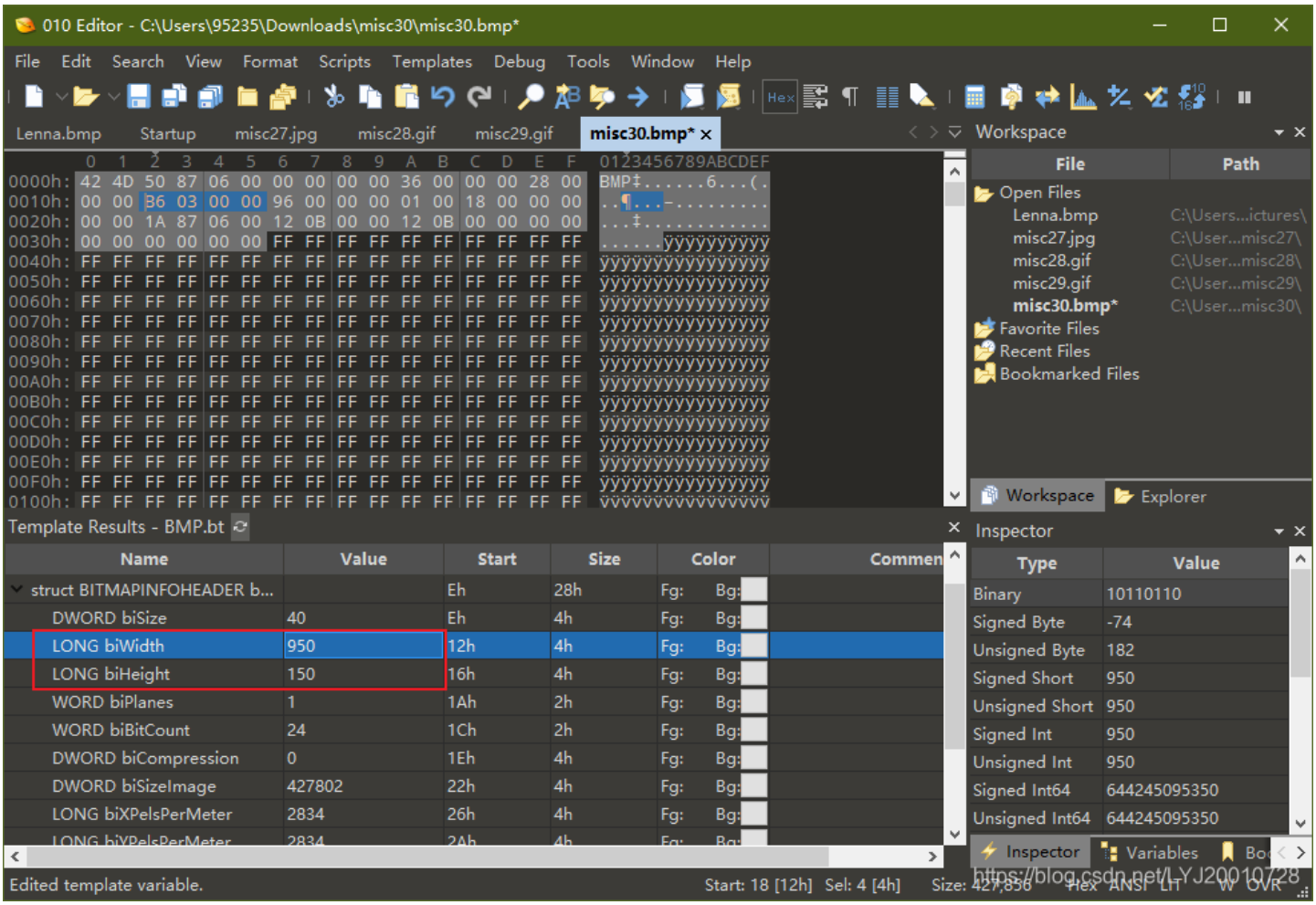
misc29

> GIF有很多帧，将每一帧的高度都改高后，用 `Stegsolve` 查看，在第八帧即可发现flag



misc30

> 根据提示修改BMP图片宽度即可发现flag



misc31

> 根据题给描述，计算正确宽度

```
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> s = (487253-53)/3
>>> s /= 150
>>> print(s)
1082.6666666666667
>>>
```

python.exe

010 Editor - C:\Users\95235\Downloads\misc31\misc31.bmp

File Edit Search View Format Scripts Templates Debug Tools Window Help

misc31.bmp ×

```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0000h: 42 4D 58 6F 07 00 00 00 00 00 36 00 00 00 28 00  BMXo......6...(.
0010h: 00 00 34 03 00 00 96 00 00 00 01 00 18 00 00 00  ..4.....-.......
0020h: 00 00 22 6F 07 00 12 0B 00 00 12 0B 00 00 00 00  .."o............
0030h: 00 00 00 00 00 00 FF FF FF FF FF FF FF FF FF FF  ......ÿÿÿÿÿÿÿÿÿÿ
0040h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0050h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0060h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0070h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0080h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0090h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00A0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00B0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00C0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00D0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00E0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00F0h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
0100h: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Workspace

File	Path
Open Files	
misc31.bmp	C:\User...misc31\
Favorite Files	
Recent Files	
misc30.bmp	C:\User...misc30\
misc29.gif	C:\User...misc29\
misc28.gif	C:\User...misc28\
misc27.jpg	C:\User...misc27\
Lenna.bmp	C:\Users...ictures\
Bookmarked Files	

Workspace Explorer

Template Results - BMP.bt

Name	Value	Start	Size	Color	Commen
struct BITMAPINFOHEADER b...		Eh	28h	Fg: Bg:	
DWORD biSize	40	Eh	4h	Fg: Bg:	
LONG biWidth	1082	12h	4h	Fg: Bg:	
LONG biHeight	150	16h	4h	Fg: Bg:	
WORD biPlanes	1	1Ah	2h	Fg: Bg:	
WORD biBitCount	24	1Ch	2h	Fg: Bg:	
DWORD biCompression	0	1Eh	4h	Fg: Bg:	
DWORD biSizeImage	487202	22h	4h	Fg: Bg:	
LONG biXPelsPerMeter	2834	26h	4h	Fg: Bg:	
LONG biYPelsPerMeter	2834	2Ah	4h	Fg: Bg:	

Inspector

Type	Value
Binary	10000100
Signed Byte	-124
Unsigned Byte	132
Signed Short	900
Unsigned Short	900
Signed Int	900
Unsigned Int	900
Signed Int64	644245095300
Unsigned Int64	644245095300

Inspector Variables Bo

Selected: 4 bytes (Range: 18 [12h] to 21 [15h]) Start: 18 [12h] Sel: 4 [4h] Size: 487,256 Hex ANSI LIT W OVR

Cmder

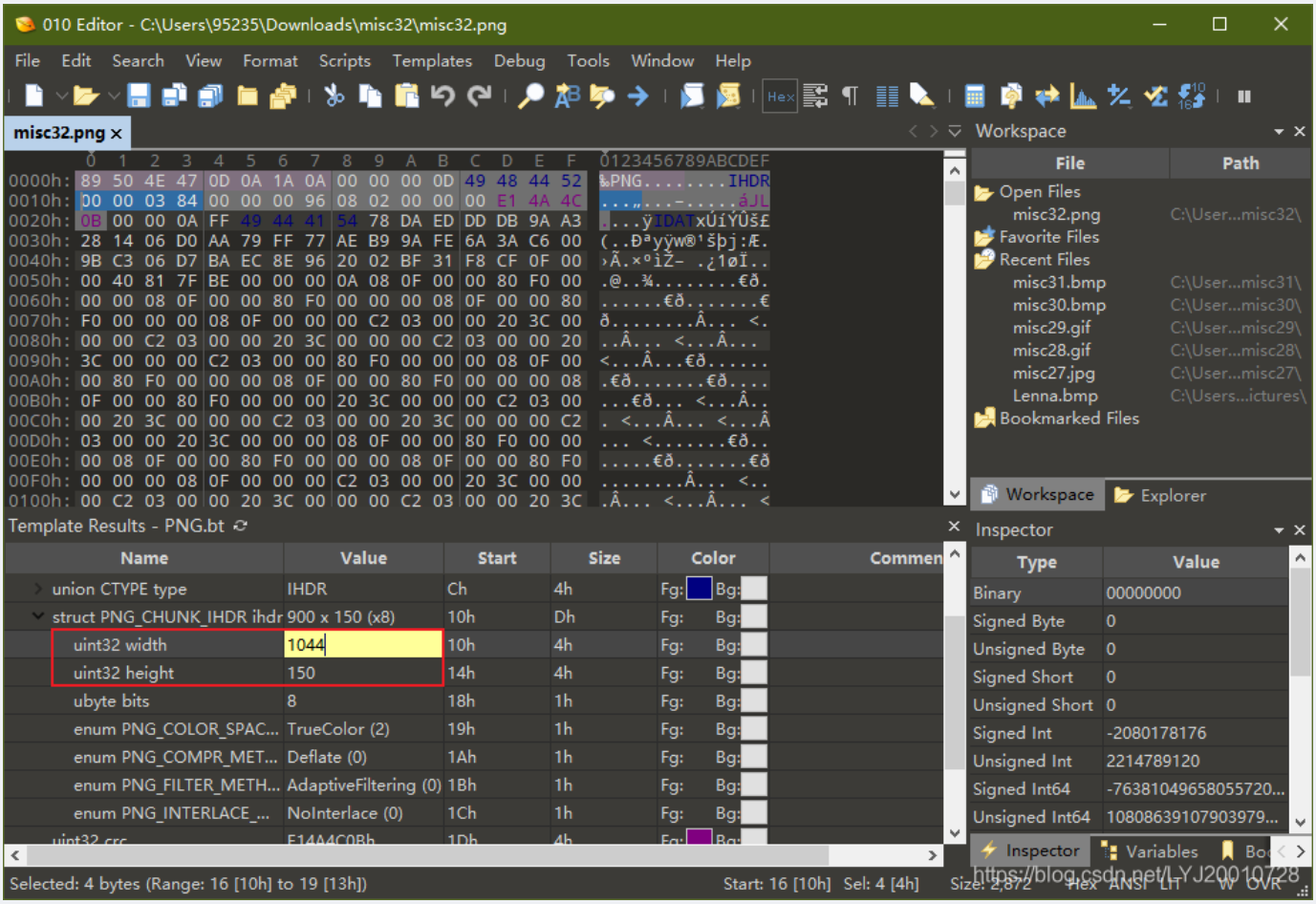
```
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc31\misc31.bmp')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
ctfshow{fb09dcc9005fe3feeefb73646b55efd5}
♀
>>> print(img.size)
(1082, 150)
>>> quit()
```

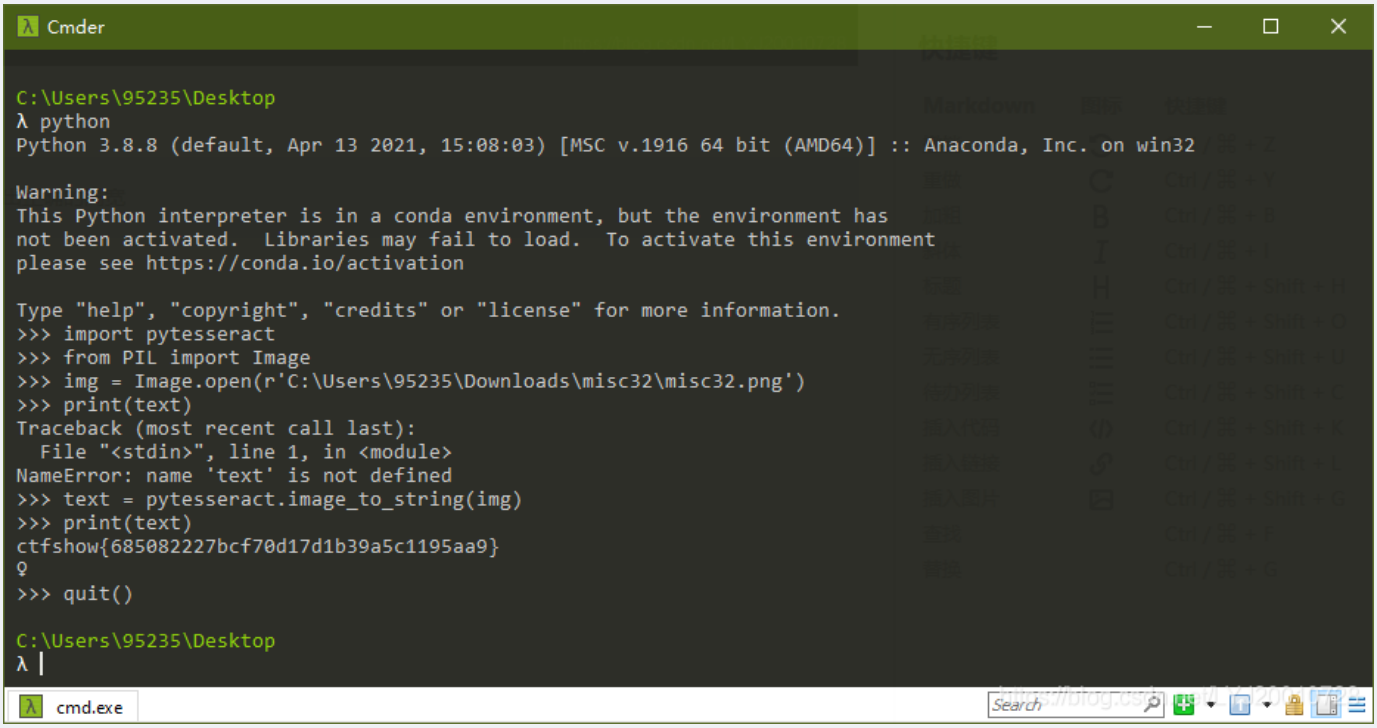
misc32

> 根据题给描述，计算出正确的高宽

```
import zlib
import struct

# 同时爆破宽度和高度
filename = "misc32.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    data = bytearray(all_b[12:29])
    n = 4095
    for w in range(n):
        width = bytearray(struct.pack('>i', w))
        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]
            crc32result = zlib.crc32(data)
            #替换成图片的crc
            if crc32result == 0xE14A4C0B:
                print("宽为：", end = '')
                print(width, end = ' ')
                print(int.from_bytes(width, byteorder='big'))
                print("高为：", end = '')
                print(height, end = ' ')
                print(int.from_bytes(height, byteorder='big'))
```

修改宽高保存后即可看到flag





misc33

根据题给描述，计算出正确的高宽

```
C:\Users\95235\Desktop
λ python exp.py
宽为：bytearray(b'\x00\x00\x03\xd2')
高为：bytearray(b'\x00\x00\x00\x8e')

C:\Users\95235\Desktop
λ
```

cmd.exe

修改宽高保存后即可看到flag

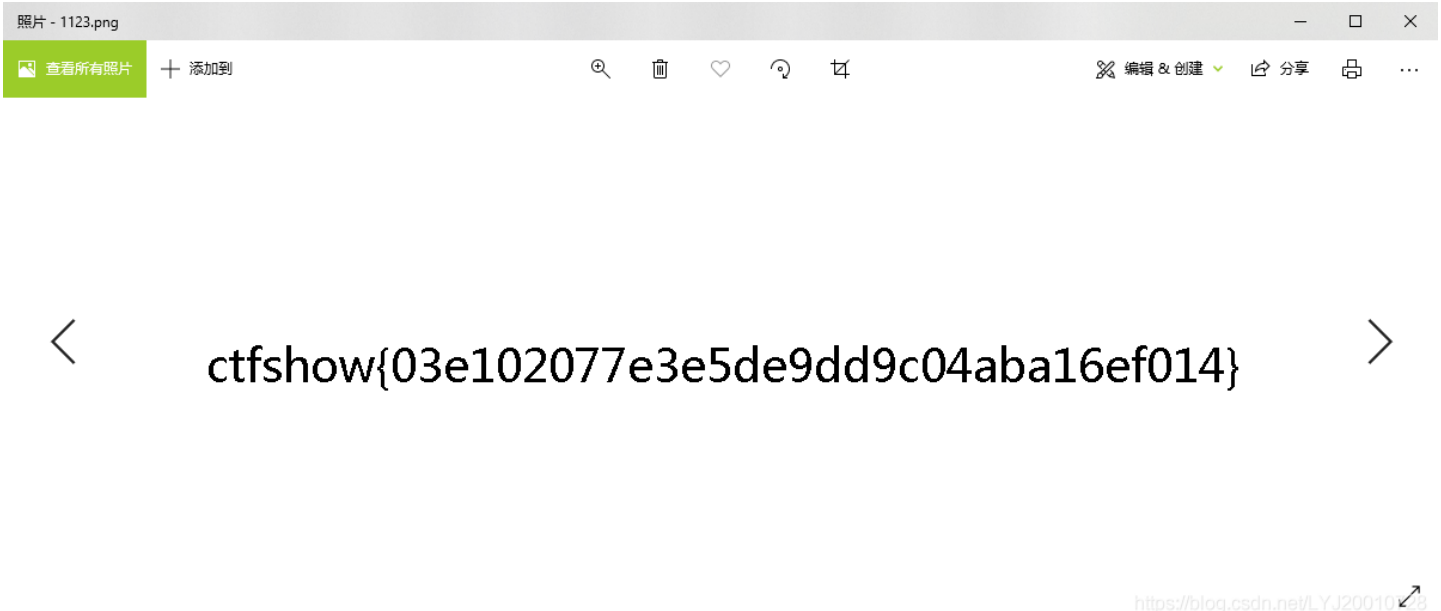
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	对应文本
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG........IHDR
00000010	00	00	03	D2	00	00	00	8E	08	02	00	00	00	52	55	A7	...Ò...Ž....RU§
00000020	98	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	˜....pHYs.......
00000030	13	01	00	9A	9C	18	00	00	0A	B3	49	44	41	54	78	DA	...šœ....³IDATxÚ
00000040	ED	DD	DB	9A	A2	48	16	80	51	DF	FF	A5	99	DB	9E	FE	íÝÛš¢H.€Qßÿ¥™Ûžþ
00000050	66	52	89	7D	44	D7	BA	EC	CE	CA	42	08	22	7E	11	A9	fR‰}D×ºìÎÊB."~.©
00000060	D7	05	00	00	14	7B	D9	05	00	00	20	BB	01	00	40	76	×....{Ù... »..@v

```
Cmder

C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc33\misc33.png')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
ctfshow{03070a10ec3a3282bale352f4e07b0a9}
Q
>>> quit()
```


misc34

利用脚本把生成的所有图片都保存下来了，观察哪个是正常的

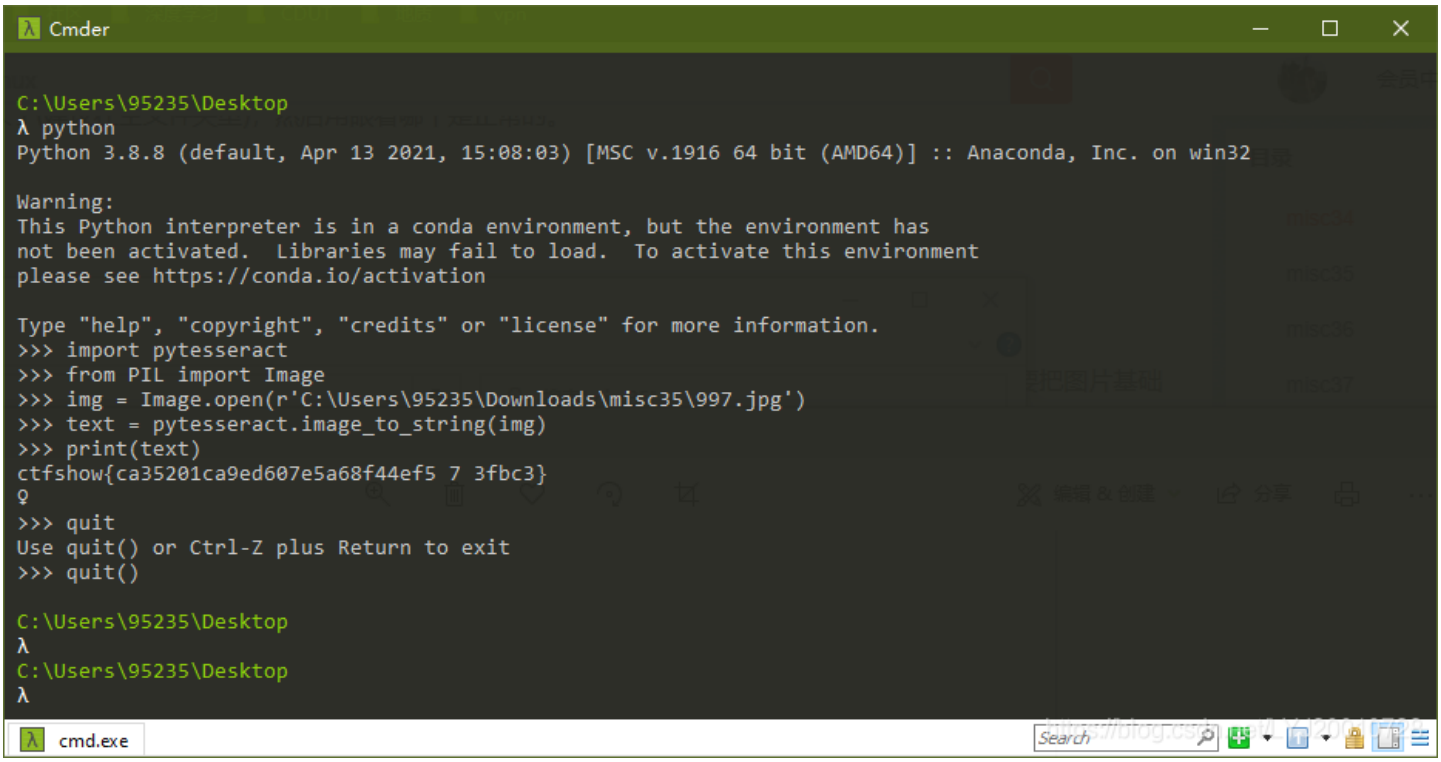
```python
import zlib
import struct
filename = r"C:\Users\95235\Downloads\misc34\misc34.png"
with open(filename, 'rb') as f:
    all_b = f.read()
    #w = all_b[16:20]
    #h = all_b[20:24]
    for i in range(901,1200):
        name = str(i) + ".png"
        f1 = open(r"C:\Users\95235\Downloads\misc34\\" + name,"wb")
        im = all_b[:16]+struct.pack('>i',i)+all_b[20:]
        f1.write(im)
        f1.close()
```



misc35

> 先把图片基础的高度调高一点（高度在600，宽度在993-1000这个范围内都可以得到flag），才能看到flag

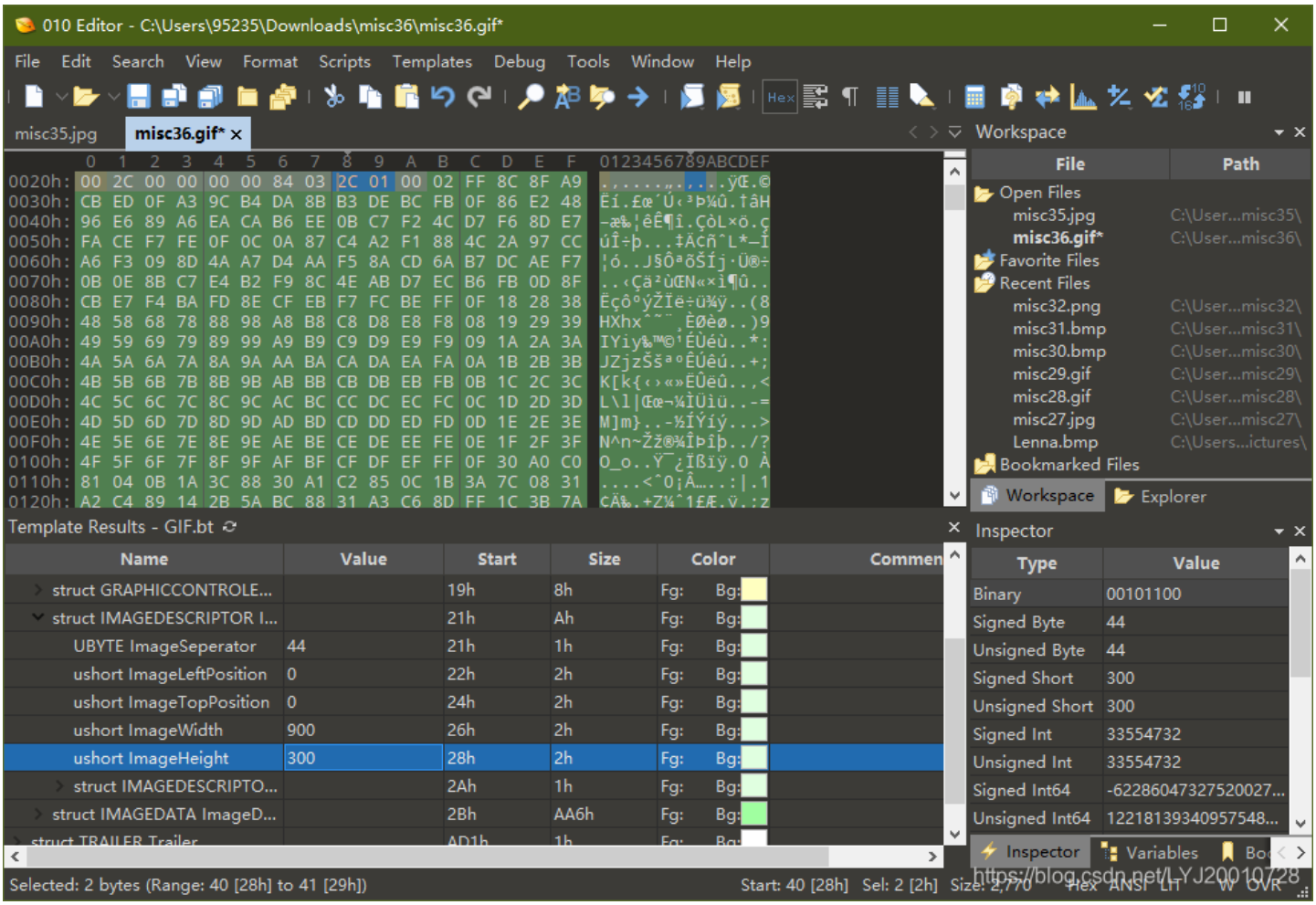
```
import zlib
import struct
filename = r"C:\Users\95235\Downloads\misc35\misc35.jpg"
with open(filename, 'rb') as f:
    all_b = f.read()
    #w = all_b[159:161]
    #h = all_b[157:159]
    for i in range(901,1200):
        name = str(i) + ".jpg"
        f1 = open(r"C:\Users\95235\Downloads\misc35\\" + name,"wb")
        im = all_b[:159]+struct.pack('>h',i)+all_b[161:]
        f1.write(im)
        f1.close()
```



misc36

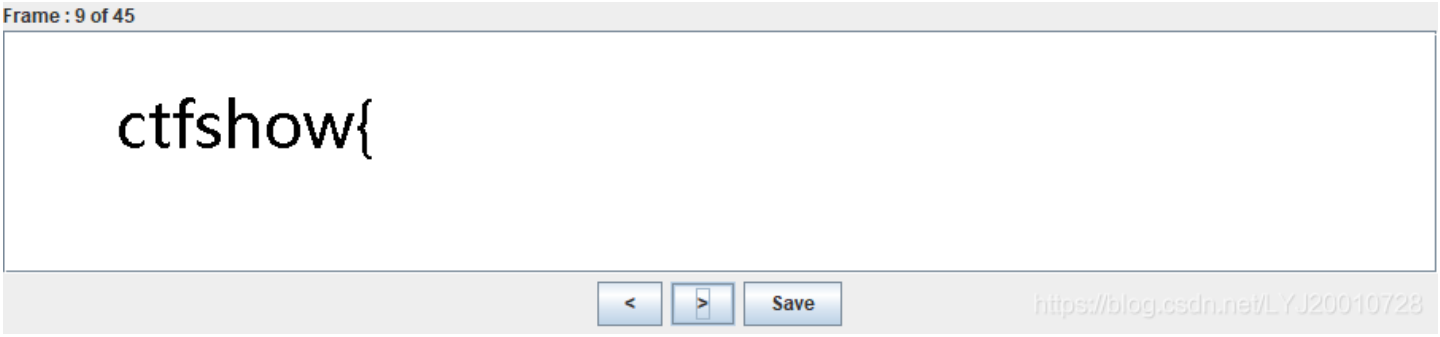
> 和上一题一样先把图片基础的高度调高一点，脚本爆破即可，用照片编辑器查看gif文件

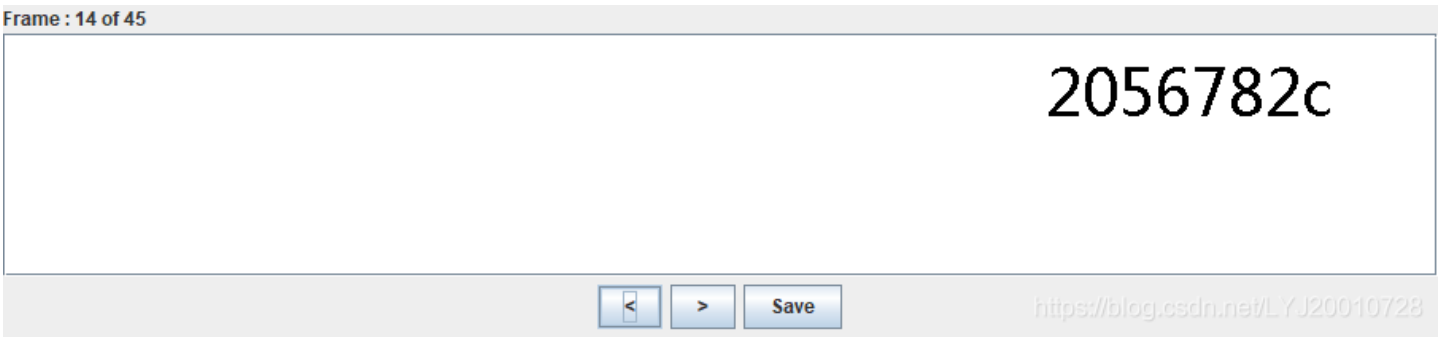
```
import zlib
import struct
filename = r"C:\Users\95235\Downloads\misc36\misc36.gif"
with open(filename, 'rb') as f:
    all_b = f.read()
    for i in range(920,951):
        name = str(i) + ".gif"
        f1 = open(r"C:\Users\95235\Downloads\misc36\\" + name,"wb")
        im = all_b[:38]+struct.pack('>h',i)[::-1]+all_b[40:]
        f1.write(im)
        f1.close()
```

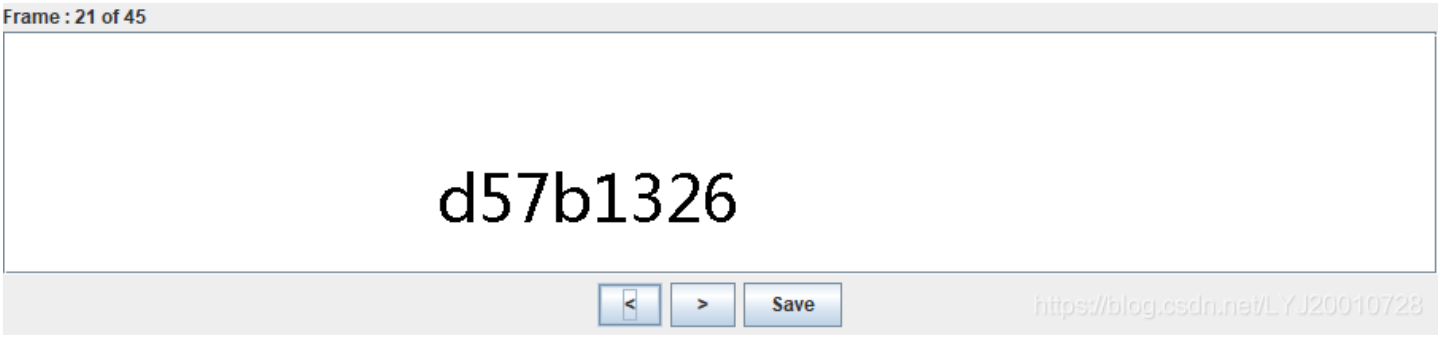


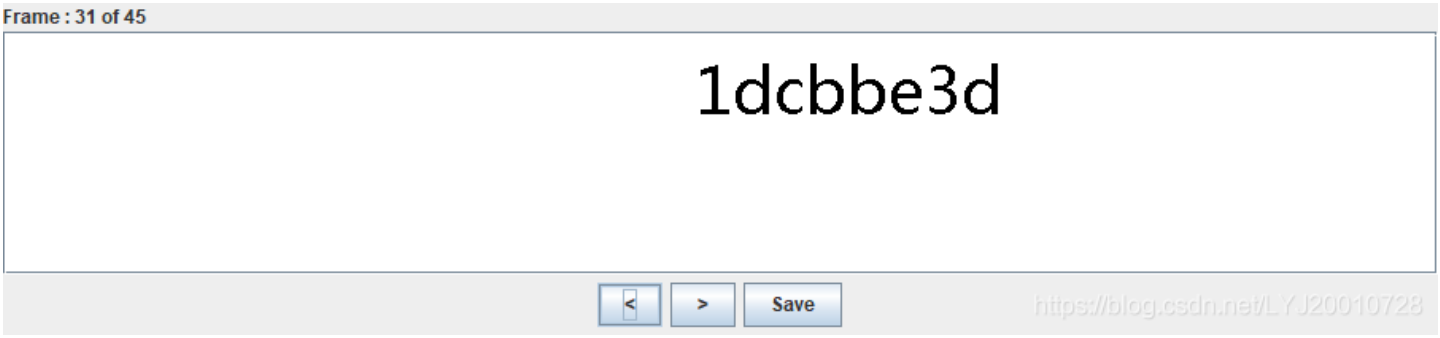
misc37

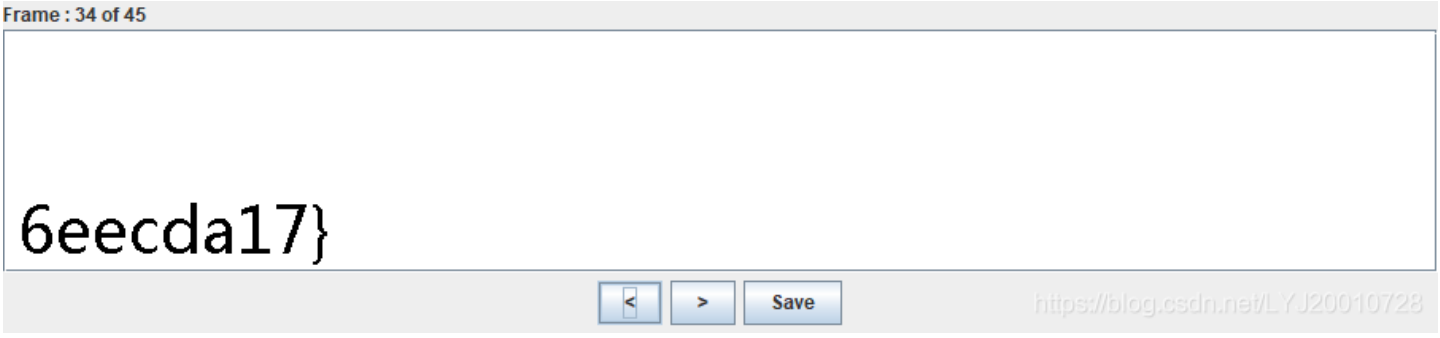
> 用 Stegsolve 查看，flag在 8、14、21、31、34 帧中，拼接起来即可





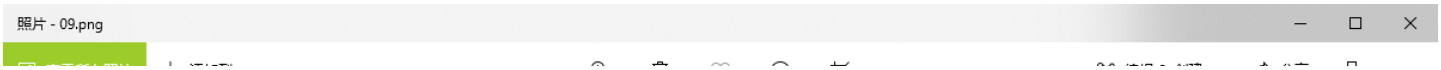




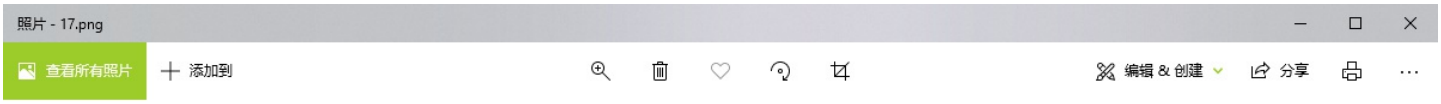


misc38

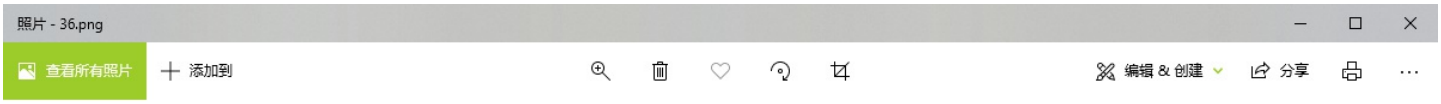
> 题目所给的是apng图片，可以使用APNG Disassembler来把每一帧分离出来， `9、17、36、40` 帧中藏有flag



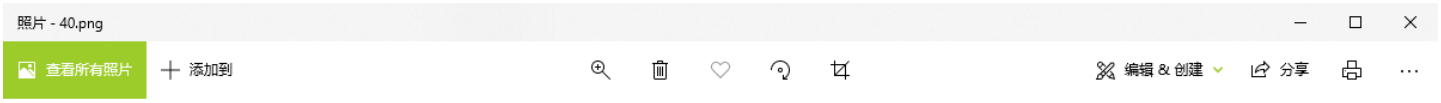
ctfshow{48



b722b570c6



03ef58cc0b



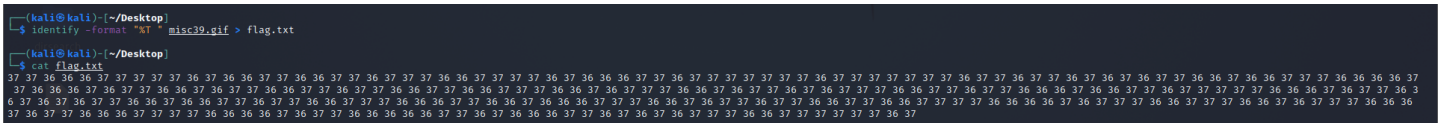
83bbf7680d}

misc39

这里是利用不同帧之间的间隔时间来隐写的，利用 `identify` 来处理该GIF图片

```
安装命令：
sudo apt-get install imagemagick
```

提取命令：`identify -format "%T " misc39.gif > 1.txt`，得到的一串36和37

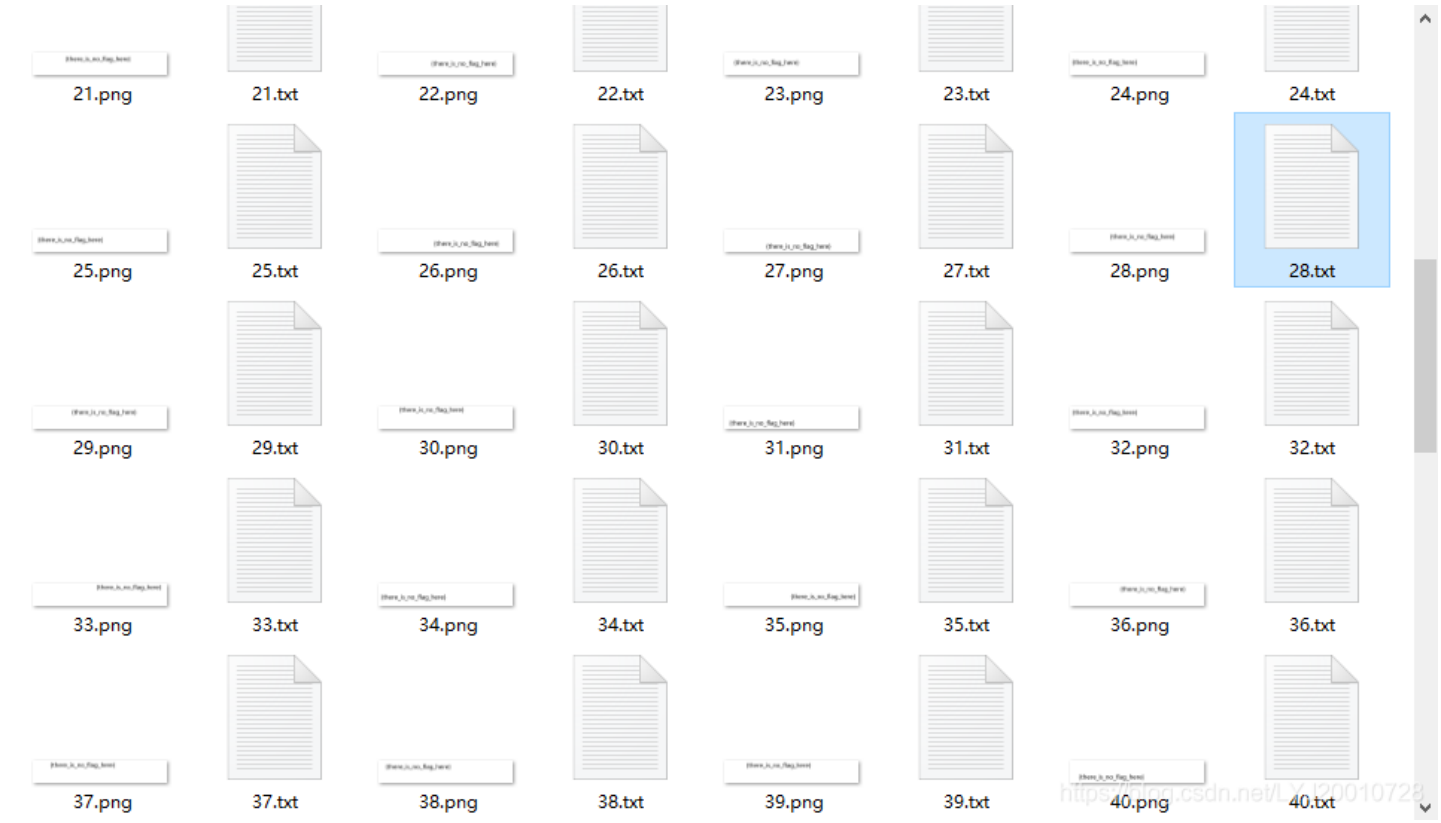


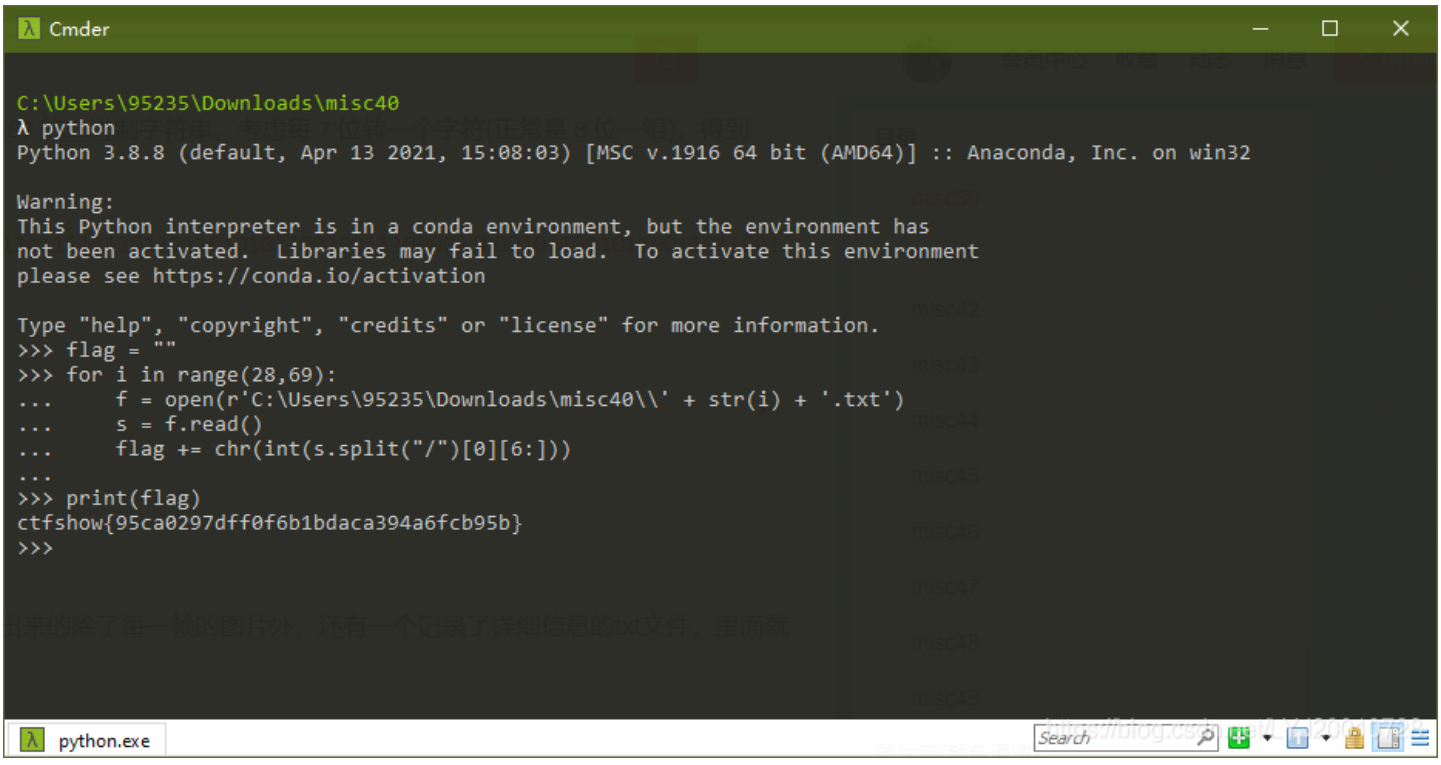
把37换成1、36换成0，得到长度为287的二进制字符串，由于无法整除8，考虑7位一组，转换成字符得到flag

```
C:\Users\95235
λpython
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> s = "11000111110100110011011100111101000110111111101111111101101101010110010011100001100010110010110011011001100
111001011100101101011100110110001001101111100010110010101100100110110011100011001000111001011011001100111100000101110
0011100101110001011000111100001011000001101000110101011100111111101"
>>> flag = ""
>>> for i in range(41):
...     flag += chr(int(s[7*i:7*(i+1)],2))
...
>>> print(flag)
ctfshow{52812ff995fb7be268d963a9ebca0459}
>>>
```


misc40

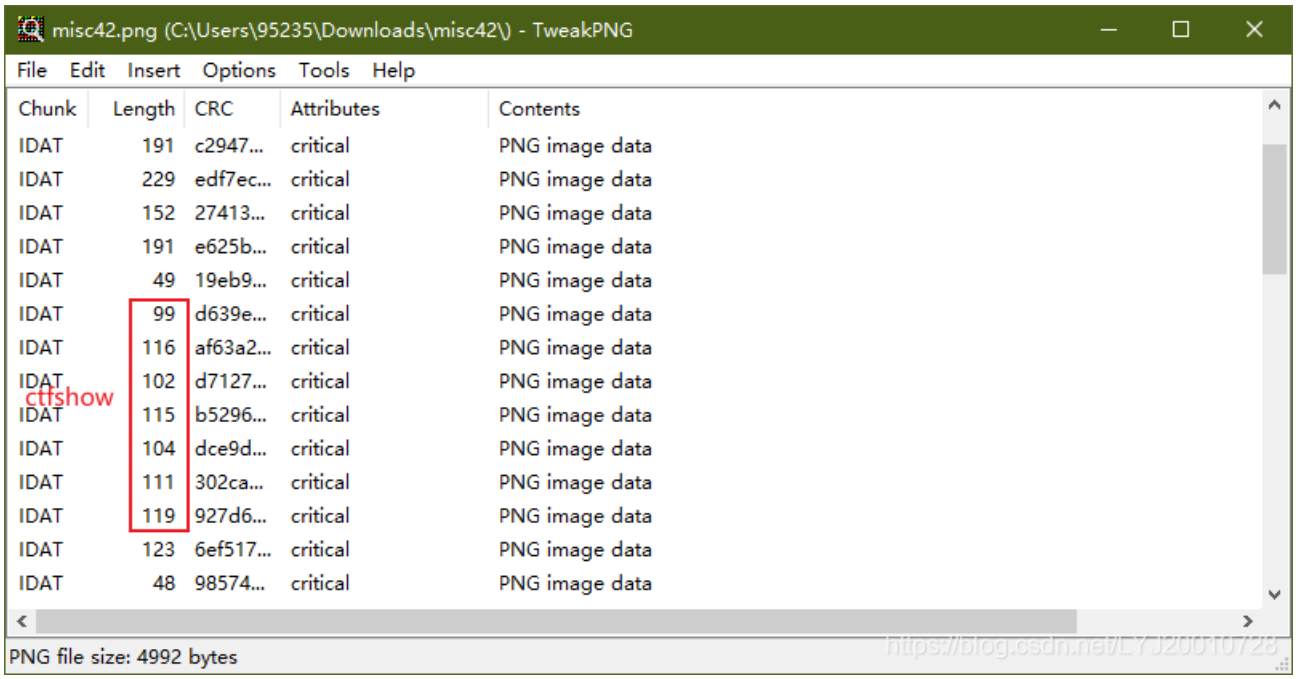
文件识别为 `apng` 文件，使用工具 `APNG Disassembler`，flag在记录详细信息的txt文件中，用脚本把flag提取出来

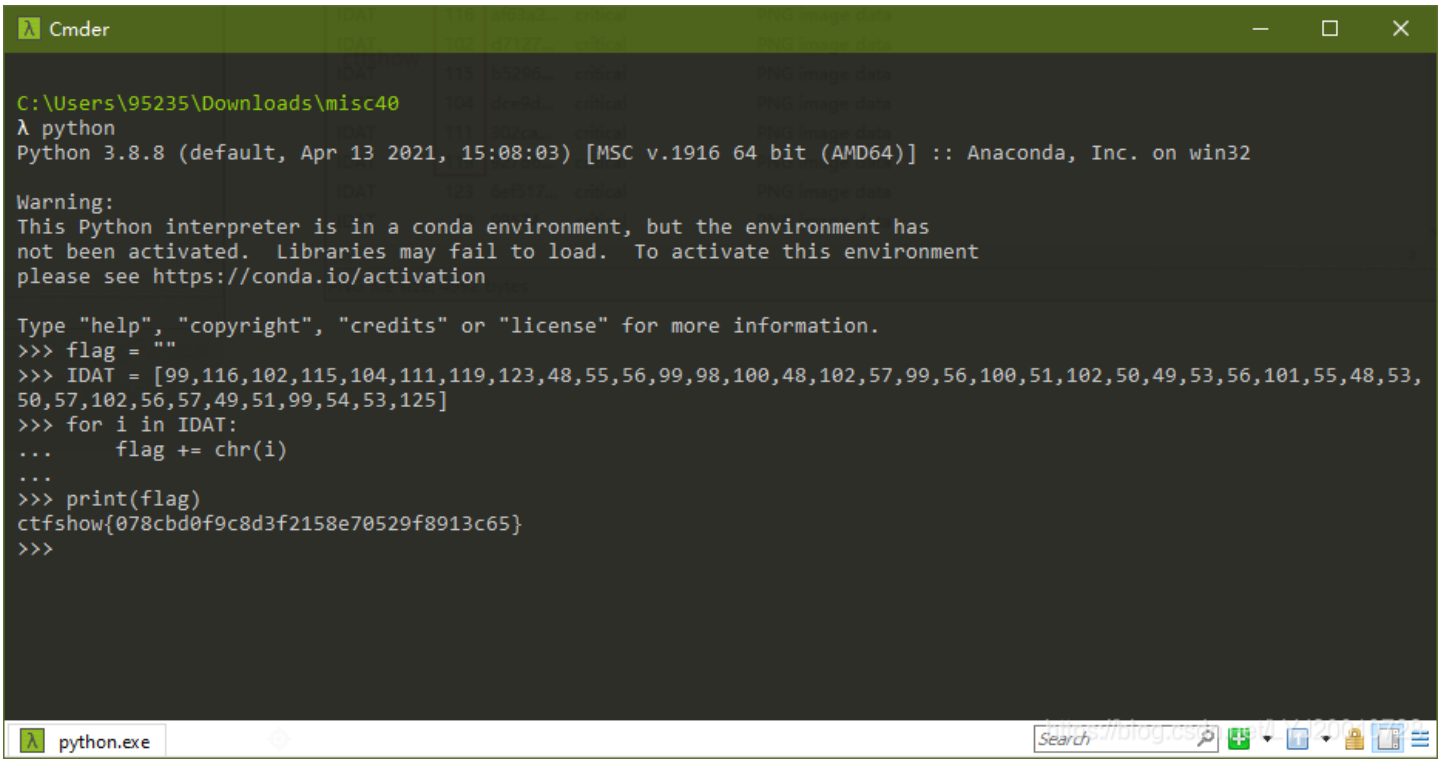




misc42

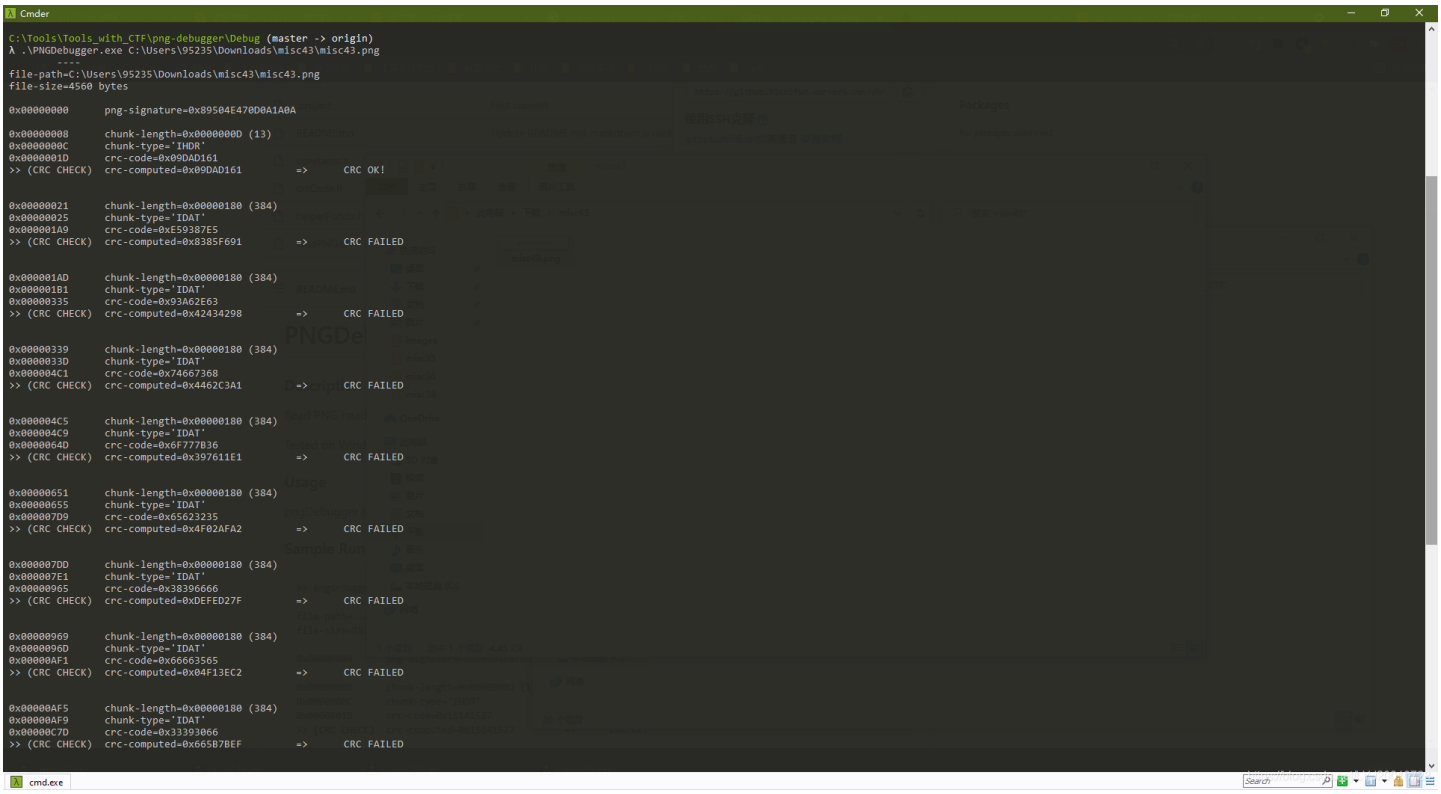
> 根据提示，用 `tweakpng` 打开图片，发现IDAT块的长度很可疑，有一部分IDAT块的长度转换为字符是 `ctfshow`，将后面的接着转换成字符即可得到flag



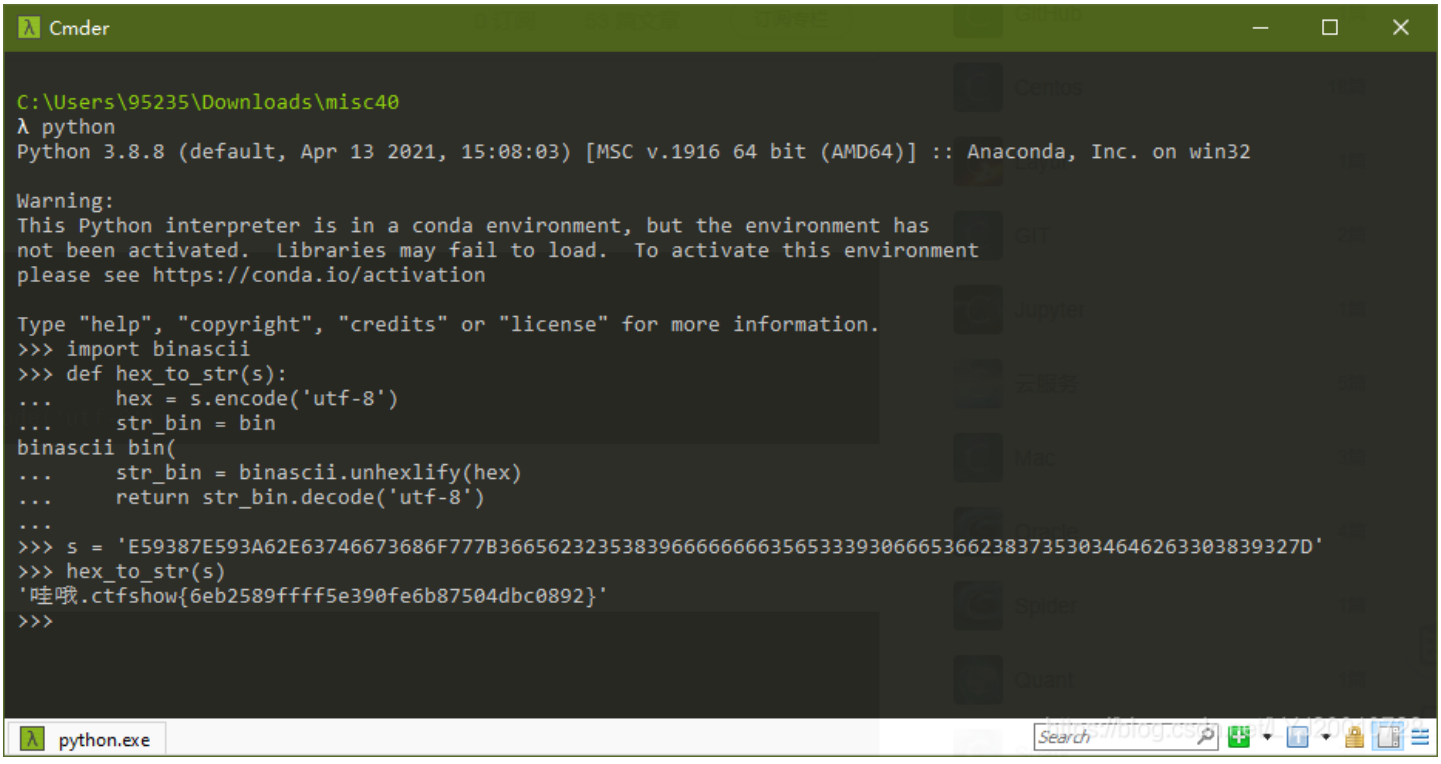


misc43

> 根据题给描述，先用tweakpng打开分析一下图片，发现报了一堆错，使用 `pngdebugger` 分析，发现所有IDAT块的crc32值都是错误的



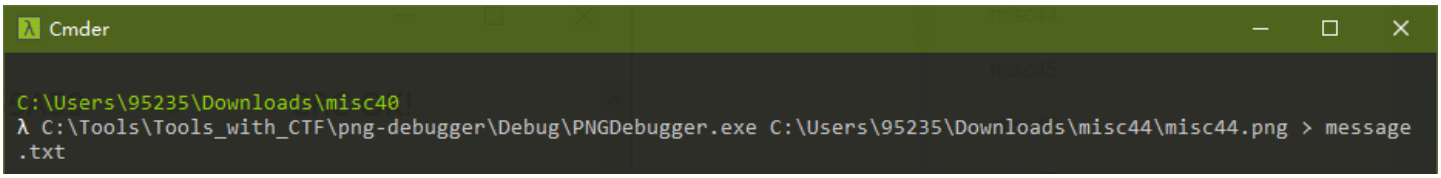
将错误的IDAT块的 `crc-code` 提取出来，拼接起来转字符串即可得到flag



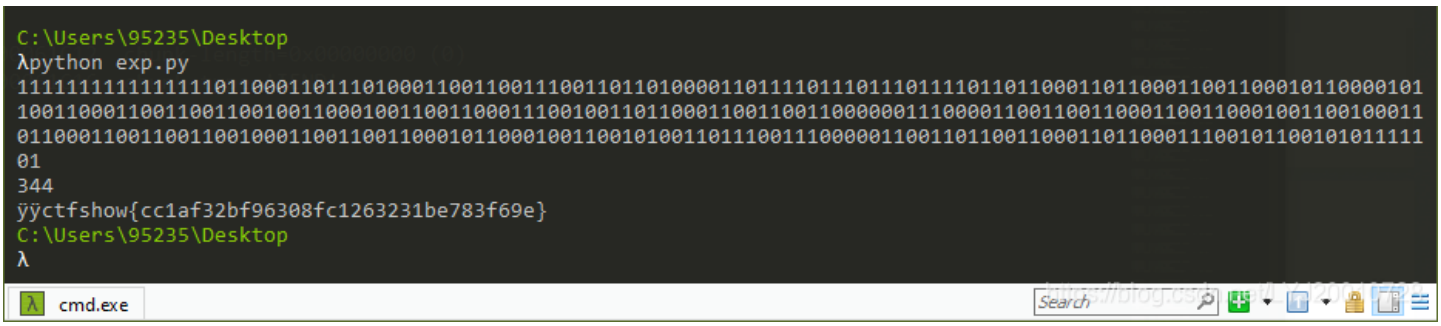

```python
import binascii
def hex_to_str(s):
    hex = s.encode('utf-8')
    str_bin = bin
    str_bin = binascii.unhexlify(hex)
    return str_bin.decode('utf-8')

s = 'E59387E593A62E63746673686F777B3665623235383966666666635653339306665366238373530346462633038393270'
hex_to_str(s)
```

misc44

> 根据提示，用 `PNGDebugger` 打开，把信息导入到txt文件中



> 利用脚本把 `CRC OK` 的替换成1， `CRC FAILED` 替换成0，注意先把前十行的内容删去，再把最后四行删去



misc45

> 根据题给描述，猜测是文件转换，测试后发现转成 `.bmp` 格式后，用 `binwalk` 提取即可，看大师傅的blog发现考察点是 `png和bmp像素点的读取方式`

```
┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk misc45.bmp

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PC bitmap, Windows 3.x format,, 900 x 150 x 24
65536         0x10000         gzip compressed data, has original file name: "flag.png", from Unix, last modified: 2021-03-29 15:44:52

┌──(kali㉿kali)-[~/Desktop]
└─$ binwalk -e misc45.bmp

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PC bitmap, Windows 3.x format,, 900 x 150 x 24
65536         0x10000         gzip compressed data, has original file name: "flag.png", from Unix, last modified: 2021-03-29 15:44:52

┌──(kali㉿kali)-[~/Desktop]
└─$ cd _misc45.bmp.extracted

┌──(kali㉿kali)-[~/Desktop/_misc45.bmp.extracted]
└─$ ls
flag.png
```



```
C:\Users\95235\Desktop
λ python
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated.  Libraries may fail to load.  To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import pytesseract
>>> from PIL import Image
>>> img = Image.open(r'C:\Users\95235\Downloads\misc45\flag.png')
>>> text = pytesseract.image_to_string(img)
>>> print(text)
ctfshow{057a722a5587979c34966c2436283e70}

>>>
```

python.exe

Search

misc46

> 根据题给描述，搜索后猜测应该是画图之类的，先提取出GIF的详细信息

```
identify misc46.gif > message.txt
```

```
┌──(kali㉿kali)-[~/Desktop]
└─$ identify misc46.gif > message.txt

┌──(kali㉿kali)-[~/Desktop]
└─$ gedit message.txt
```

Open | message.txt ~/Desktop | Save

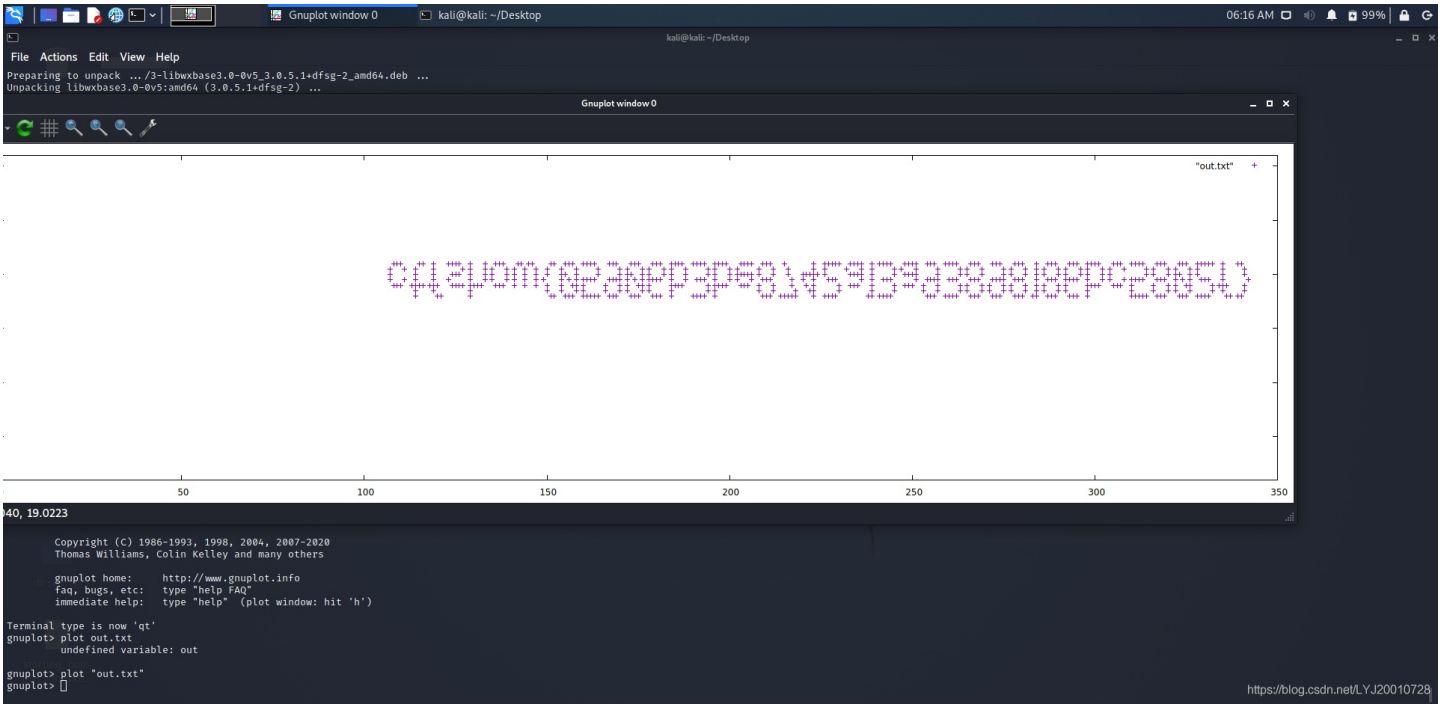
```
1 misc46.gif[0] GIF 900×150 900×150+0+0 8-bit sRGB 2c 0.010u 0:00.015
2 misc46.gif[1] GIF 450×50 900×150+174+49 8-bit sRGB 16c 0.010u 0:00.019
3 misc46.gif[2] GIF 450×50 900×150+196+47 8-bit sRGB 16c 0.010u 0:00.019
4 misc46.gif[3] GIF 450×50 900×150+256+49 8-bit sRGB 16c 0.010u 0:00.019
5 misc46.gif[4] GIF 450×50 900×150+293+52 8-bit sRGB 16c 0.010u 0:00.019
6 misc46.gif[5] GIF 450×50 900×150+220+49 8-bit sRGB 16c 0.010u 0:00.019
7 misc46.gif[6] GIF 450×50 900×150+245+47 8-bit sRGB 16c 0.010u 0:00.019
```

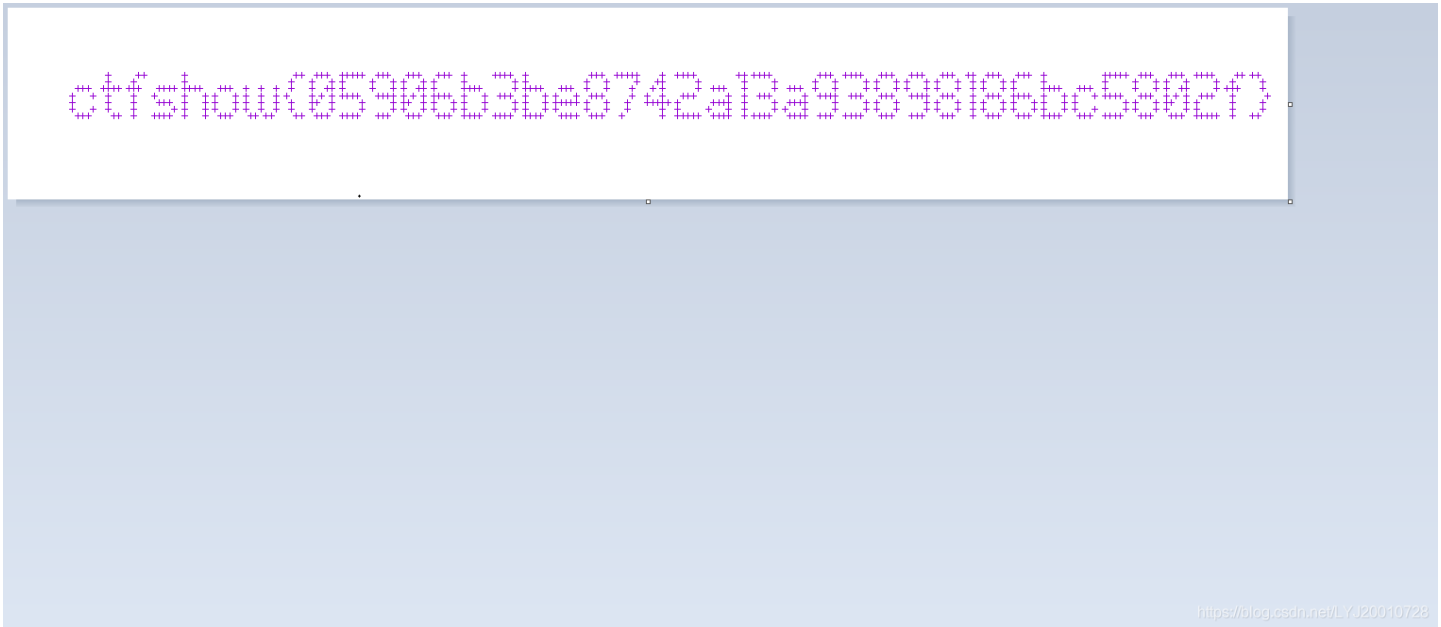
> 观察得到的信息，其中 `0+0、174+49、196+47` 这些是偏移量，用其来进行画图

坐标提取：

```
f = open(r"C:\Users\95235\Downloads\misc46\message.txt","r")
x = f.readlines()
f.close()

f = open(r"C:\Users\95235\Downloads\misc46\out.txt","w")
for i in x:
    f.write(i.split("+")[1])
    f.write(" ")
    f.write(i.split("+")[2][:2])
    f.write("\n")
f.close()
```

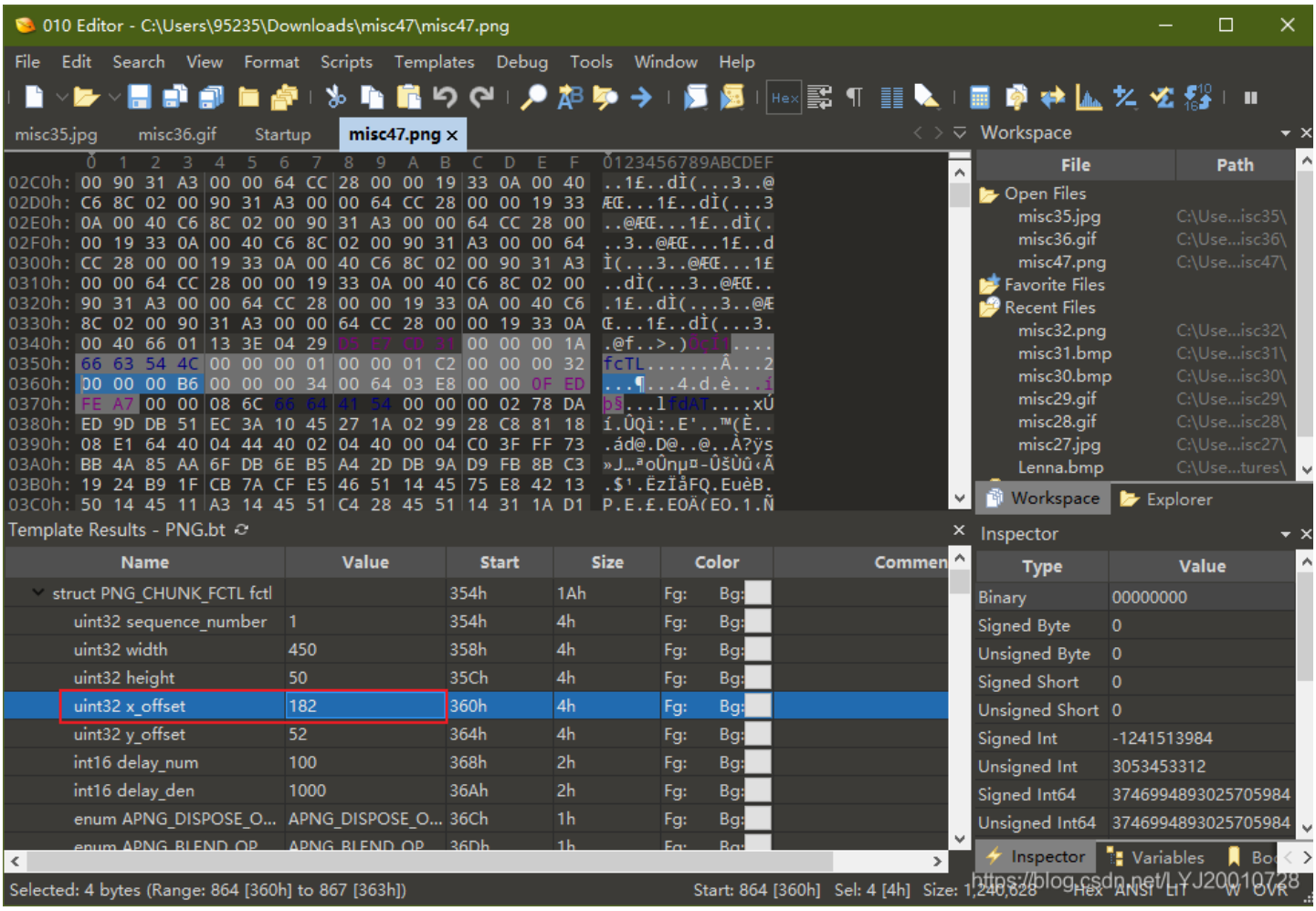
> 根据得到的点坐标进行绘图



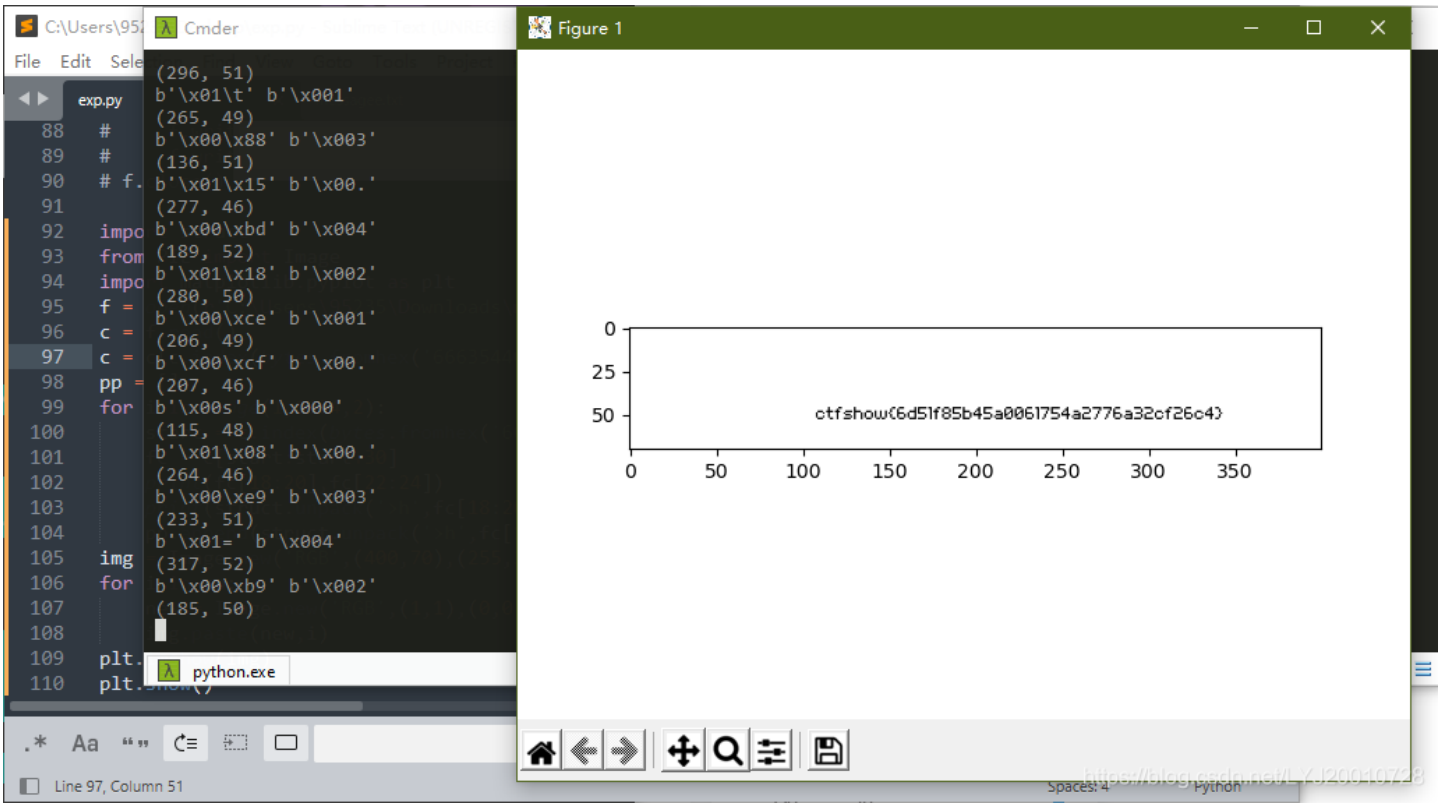


misc47

测试后发现是apng格式，解题的思路是根据每一个IDAT块前面的一个fcTL块中包含的水平垂直偏移量

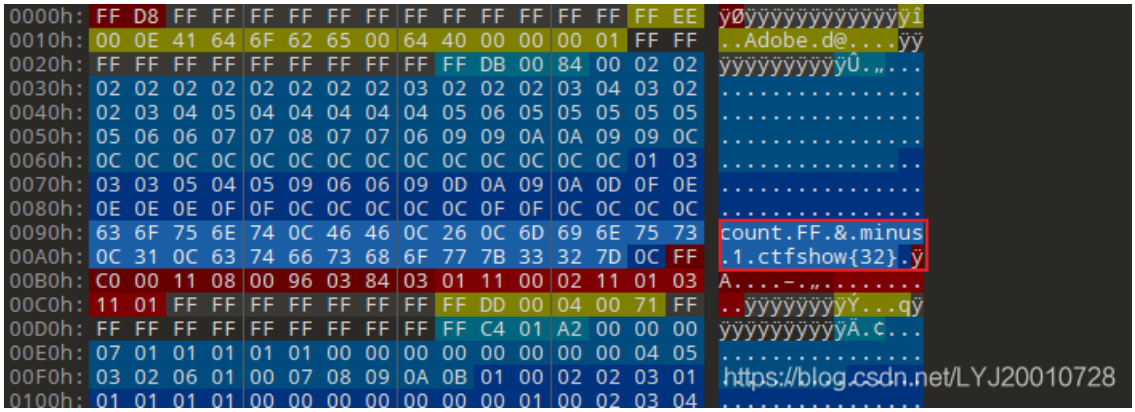



```
import struct
from PIL import Image
import matplotlib.pyplot as plt
f = open(r'C:\Users\95235\Downloads\misc47\misc47.png','rb')
c = f.read()
c = c[c.index(bytes.fromhex('6663544C00000001')):]
pp = []
for i in range(1,1124,2):
    start = c.index(bytes.fromhex('6663544C0000')+struct.pack('>h',i))
    fc = c[start:start+30]
    print(fc[18:20],fc[22:24])
    print(struct.unpack('>h',fc[18:20])+struct.unpack('>h',fc[22:24]))
    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)
plt.imshow(img)
plt.show()
```



misc48

用 010 editor 打开，发现提示 统计FF的数量再减去1、ctfshow{}中包含32个字符



因为flag长度是32位，所以只需要统计前32个段就行

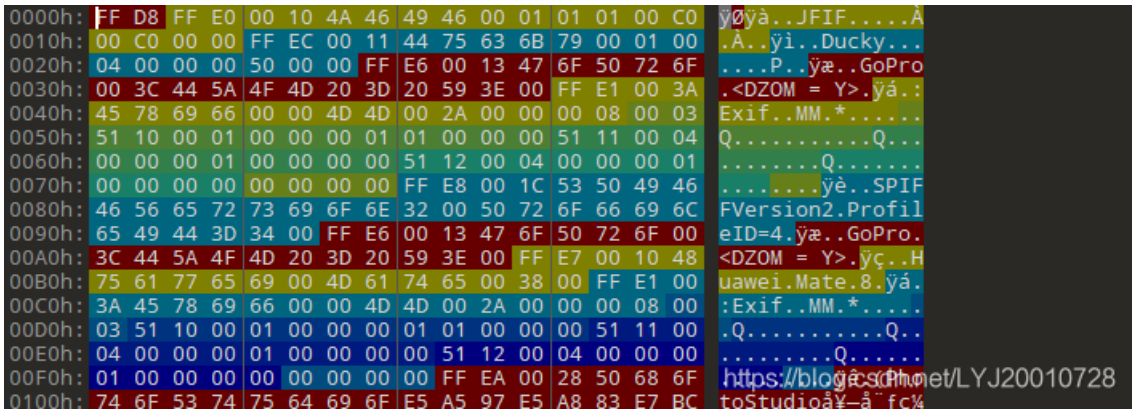
```
0 12 11 0 7 10 13 13 9 0 9 13 0 13 6 0 10 9 2 1 0 1 10 8 11 5 12 7 2 2 3 10
```

分别转换成hex即可

```
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)
```

misc49

> 用 `010 editor` 打开， `FFE` 后面的就是flag的值



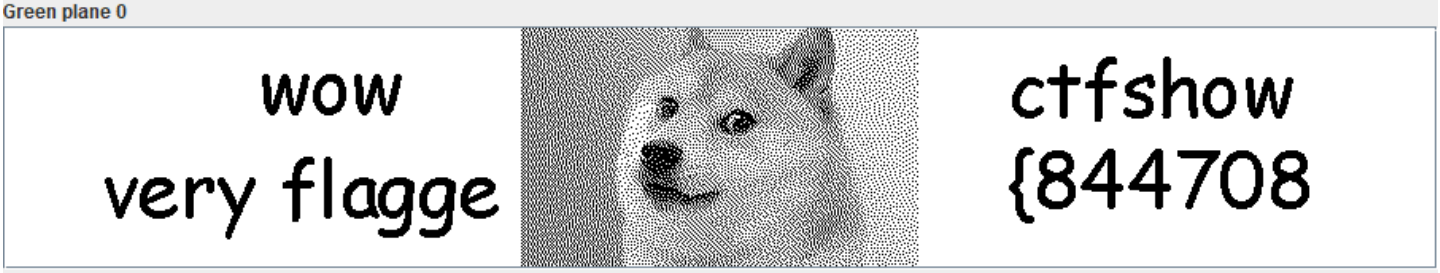
图片篇(颜色通道)

misc50

> 由于是颜色通道篇的，很自然想到用 `Stegsolve` 查看一下，

File Analyse Help

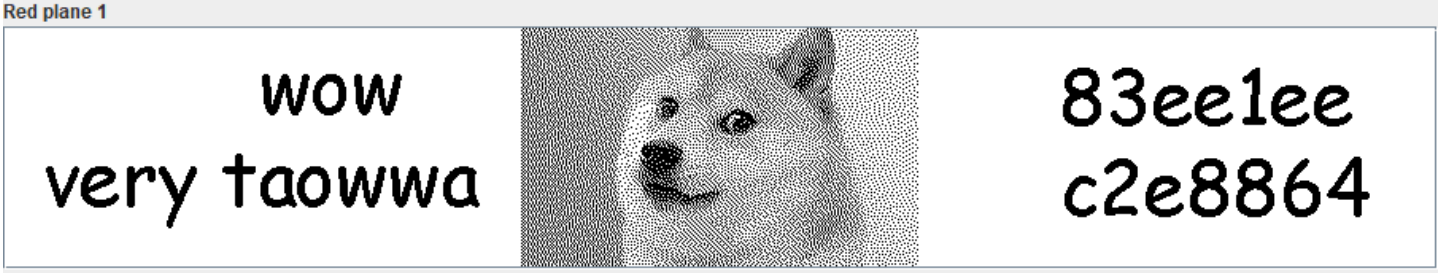
Green plane 0



< >

File Analyse Help

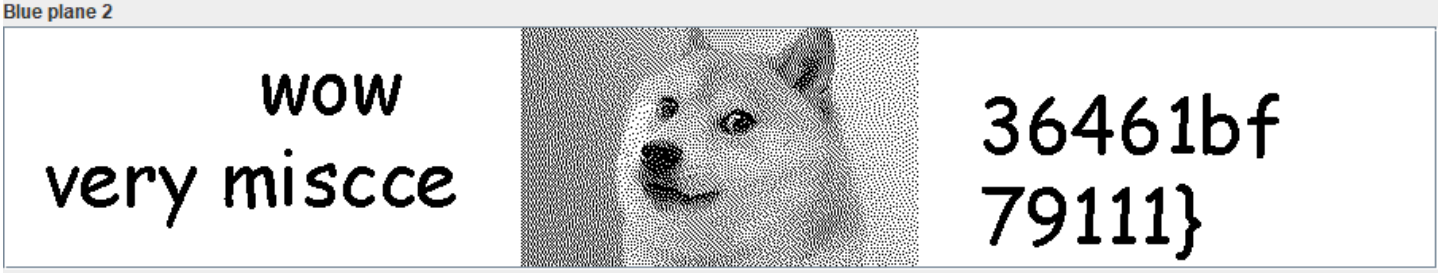
Red plane 1



< >

File Analyse Help

Blue plane 2



< >