




# ctfshow-misc入门 1-30

原创

V3geD4g  于 2021-03-31 16:30:41 发布  1177  收藏 4

分类专栏: [wp](#) 文章标签: [其他](#) [python](#) [信息安全](#)

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/xczzhf/article/details/115353783>

版权



[wp 专栏收录该内容](#)

3 篇文章 0 订阅

订阅专栏

ctfshow开始上新misc入门了, 题目知识点涵盖的非常广, 八神师傅tql, 目前暂时出到30题, 简单记录下

## misc1-3

打开图片就是, 其中3需要下载一个bpgviewer, 百度下个就行

## misc4

txt改成png就行

## misc5

记事本或者010打开, 文件末尾即为flag, 或者strings misc5.png | grep ctfshow

## misc6

记事本或者010打开, flag在文件中间, 或者strings misc6.png | grep ctfshow

## misc7

同上, strings misc7.jpg | grep ctfshow

## misc8

同上, strings misc8.png | grep ctfshow, 在photoshop:LayerName中

## misc9

同上, strings misc9.png | grep ctfshow

## misc 10

zlib解压最后一个idat块, 代码如下

```
import zlib
s=bytes.fromhex('789C4B2E492BCEC82FAF363635363235323132494C36B34C4E3233493333313637B3B030354C4C36B734A8050009960BD1')
print(zlib.decompress(s))
```

## misc 11

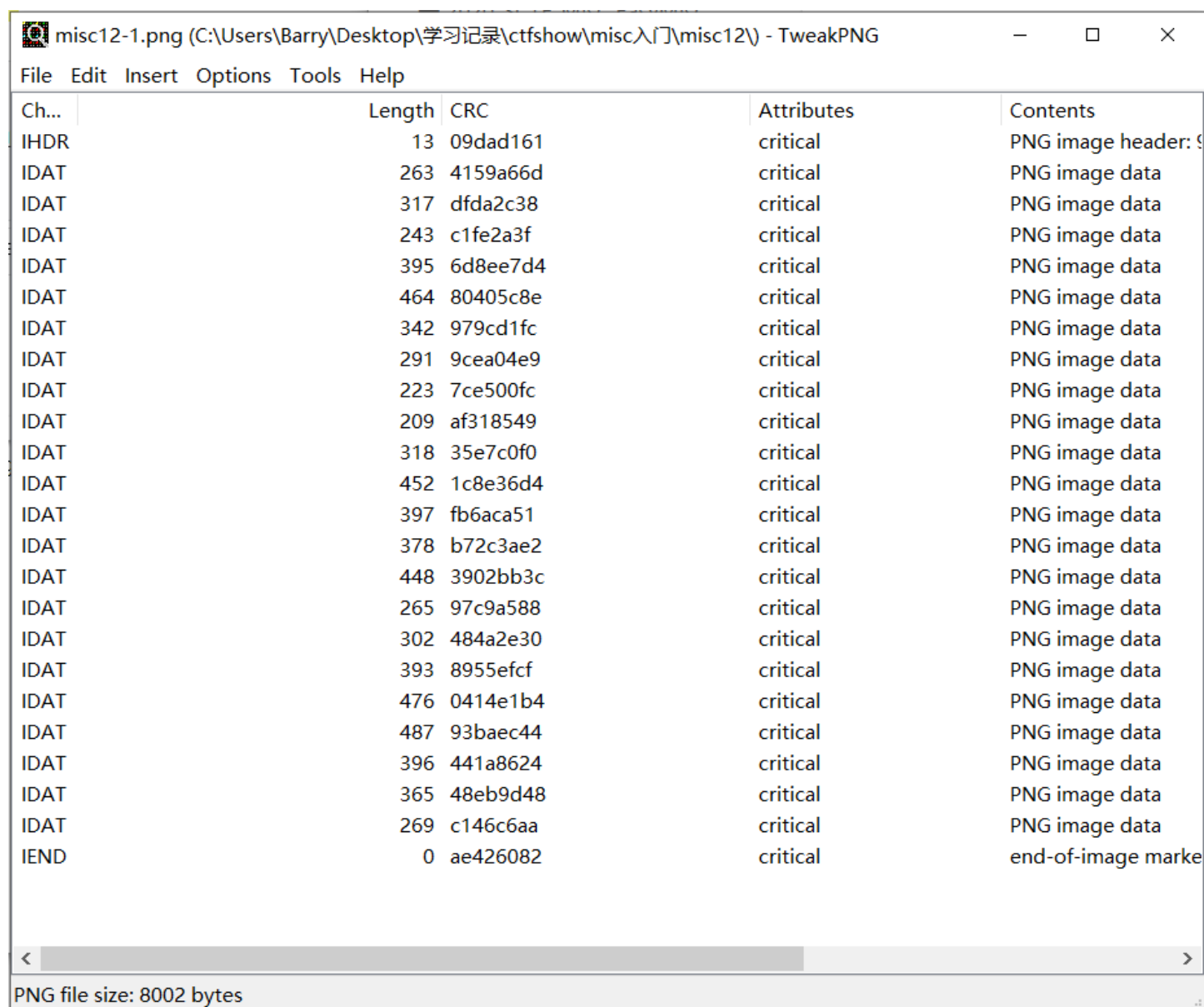
用tweakpng删掉第一个idat块即可

Ch...	Length	CRC	Attributes
IHDR	13	09dad161	critical
IDAT	2931	c464ae32	critical
IDAT	7541	228b674b	critical
IEND	0	ae426082	critical

即大小为2931的这一块

## misc12

与上题一样，也是删除idat块，慢慢尝试删除前八个的时候出了flag，最后剩下这些块



The screenshot shows the TweakPNG application window titled "misc12-1.png (C:\Users\Barry\Desktop\学习记录\ctfshow\misc入门\misc12\)" with a menu bar (File, Edit, Insert, Options, Tools, Help). The main window displays a table of PNG chunks:

Ch...	Length	CRC	Attributes	Contents
IHDR	13	09dad161	critical	PNG image header: 9
IDAT	263	4159a66d	critical	PNG image data
IDAT	317	dfda2c38	critical	PNG image data
IDAT	243	c1fe2a3f	critical	PNG image data
IDAT	395	6d8ee7d4	critical	PNG image data
IDAT	464	80405c8e	critical	PNG image data
IDAT	342	979cd1fc	critical	PNG image data
IDAT	291	9cea04e9	critical	PNG image data
IDAT	223	7ce500fc	critical	PNG image data
IDAT	209	af318549	critical	PNG image data
IDAT	318	35e7c0f0	critical	PNG image data
IDAT	452	1c8e36d4	critical	PNG image data
IDAT	397	fb6aca51	critical	PNG image data
IDAT	378	b72c3ae2	critical	PNG image data
IDAT	448	3902bb3c	critical	PNG image data
IDAT	265	97c9a588	critical	PNG image data
IDAT	302	484a2e30	critical	PNG image data
IDAT	393	8955efcf	critical	PNG image data
IDAT	476	0414e1b4	critical	PNG image data
IDAT	487	93baec44	critical	PNG image data
IDAT	396	441a8624	critical	PNG image data
IDAT	365	48eb9d48	critical	PNG image data
IDAT	269	c146c6aa	critical	PNG image data
IEND	0	ae426082	critical	end-of-image marke

At the bottom of the window, it states "PNG file size: 8002 bytes".

## misc13

用tweakpng发现IEND块长度异常，正常来讲它的长度应该是0，但是这里为2，一开始不知道多的这个块有啥意思，尝试用zsteg得到了错误的答案，后来问了八神师傅，提示多的那两位是flag的位置，才恍然大悟，这里也加了混淆，即需要隔位取，所以不能直接strings出来

```
00001020 | 7A 00 00 00 02 49 45 4E 44 0D E1 67 7D 8B 8F | z....IEND. 醜?
```

这里多了两位0D E1

```
00000DE0 | D4 63 1A 74 B9 66 85 73 86 68 AA 6F 4B 77 B0 7B | Üc.t?f?s?h?oKw?{"
00000DF0 | 21 61 14 65 53 36 A5 65 54 33 34 65 78 61 25 34 | !a.eS6 T34exa%4"
00000E00 | DD 38 EF 66 AB 35 10 31 95 38 1F 62 82 37 BA 65 | ?醜?.1?.b?篡??篡"
00000E10 | 45 34 7C 32 54 64 7E 37 3A 64 E4 65 F1 36 FA 66 | E4|2Td~7:d缺?鷗f"
00000E20 | F5 34 1E 31 07 32 1D 66 54 38 F1 33 32 39 E9 61 | ?.1.2.ft8?29閑閑"
00000E30 | 6C 7D 94 28 62 E7 A1 CA A7 24 8E 7E B8 2A AC 1F | 1)?b纒失$葵??*?"
00000E40 | A1 93 E3 FF 9F 13 00 AF 30 88 2A 73 79 F6 9F 49 | ???.??sy鯨Iy?裂"
00000E50 | 20 D1 85 84 93 13 F7 35 D1 85 25 55 17 06 9E EA | 襲創.?襲%U..焜ê"
00000E60 | B9 59 9C C7 15 3F 79 B2 A6 4D C3 17 AA 7C 12 31 | 筭濂.?y拔M?猓.112
00000E70 | 25 03 FE FE AB C8 63 7C BE CE 1C DB 4E D4 7D 35 | %. c|疚.時許52
00000E80 | D6 42 BD B2 FF 7C FC 1A 78 1B 7E 02 6C 7D E2 22 | 總匠 \ \ . . 1..52
```

到这个位置发现flag

随便写个脚本提取下

```
r = ''
s=bytes.fromhex('631A74B96685738668AA6F4B77B07B216114655336A5655433346578612534DD38EF66AB35103195381F628237BA6545347C3254647E373A64E465F136FA66F5341E3107321D665438F1333239E9616C7D942862E7A1CAA7248E7EB82AAC1FA193E3FF9F1300AF30882A7379F69F4920D185849313F735D185255517069EEAB9599CC7153F79B2A64DC317AA7C12312503FEFEABC8637CBECE1CDB4ED47D35D643BDB3FF7C5C1A781B7F026C7953327A7CC43E972E74B2471754C1A6E56FED38C5C80F4989933904D5A7DF2714589C964C1F5BDF9C929239ABA43BD3CA3109C059EAF30F5A23DCDC34C8DE3A9C35A0A7ABD55645BC5D3F5450D240DDB6147DFCDCFE33D27235C072BB9792BE5C8923')
for i in range(0,len(s),2):
    try:
        r+=chr(s[i])
    except:
        pass
print(r)
```

## misc14

图片里有两个jpg，手动提取下，定位到第三个FFD8为位置直接复制到末尾保存为新图片，即为flag

## misc15

记事本010或者string就有，利用的是bmp文件头部12-15位的这个偏移量

## misc16

binwalk分离，flag在DD4这个文件里，具体原理还不太清楚，好像是LZMA压缩的数据？

## misc17

卡了最久的一题，太难想到了，原理也不清楚，都是后来问师傅才做出来的

首先binwalk一下可以发现一个bzip2压缩包，但是无论binwalk分解还是手动分离都是损坏的，打不开，后来解没有任何思路了

```
root@kali:~/文档# binwalk misc17.png
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 900 x 150, 8-bit/color RGB, non-interlaced
3462	0xD86	bzip2 compressed data, block size = 900k

问了师傅才知道，需要先用zsteg提取数据，然后再用binwalk分离，最后得到一张png即为flag

### zsteg

```
root@kali:~/文档# zsteg misc17.png
[?] 3544 bytes of extra data after zlib stream
extradata:0
00000000: e1 1f 30 53 86 4f c5 a4 1b f5 e6 e5 c7 46 0a 92 |..0S.0.....F.
00000010: 9b ee 72 e7 c9 9e b9 a7 74 de 92 4d ad 61 5b 58 |..r.....t..M.a[
00000020: f2 98 65 77 2b d2 d3 85 32 fc 08 83 86 1f 0f 1e |..ew+...2.....
00000030: cb ab ac 9c 4b ca 02 20 e2 ce e4 ae 60 1a 2c c6 |...K.. ..`.,
00000040: 7b c8 9a 77 31 2f 9e 67 db d9 3e 53 fe 17 a5 50 |{..w1/.g..>S...
00000050: 20 e5 1d 8c d5 49 4e 52 a5 54 31 cb 8b c5 3b 09 | ....INR.T1...;
00000060: a2 a6 fe 5b da 4f 9e 78 9c 5d 46 d6 e2 6b 6b 2a |... [.0.x.]F..kk
00000070: f2 62 0c ba 70 19 a0 27 f3 84 77 99 02 77 05 79 |.b..p..'..w..w.
00000080: 5b 44 b7 79 b3 54 11 a1 f3 54 34 56 7e ff 55 d1 |[D.y.T...T4V~.U
```

至于zsteg如何分离数据而不是lsb数据，也是问了师傅才知道的

```
zsteg -E 'extradata:0' misc17.png > 17.tmp
```

### binwalk

```
root@kali:~/文档# binwalk 17.tmp
```

DECIMAL	HEXADECIMAL	DESCRIPTION
197	0x1F1	bzip2 compressed data, block size = 900k

binwalk -e出来的1F1直接就是一张png图片，就是flag了

## misc18

exiftool一下就出来了

```

root@kali:~/文档# exiftool misc18.jpg
ExifTool Version Number      : 12.16
File Name                    : misc18.jpg
Directory                    : .
File Size                    : 21 KiB
File Modification Date/Time  : 2021:03:14 00:44:41+08:00
File Access Date/Time       : 2021:03:28 18:56:42+08:00
File Inode Change Date/Time  : 2021:03:28 18:56:42+08:00
File Permissions             : rwxrw-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

同上，flag在exif信息中

## misc20

同上，flag在exif信息中，还是个中文的，注意诶是a不是i

```

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第七九西二一第第诶第五九三易四二大括号
Image Width                  : 900

```

## misc21

提示flag在序列号里，所以用继续用exiftool看一下，发现序列号是串hex，提示hex(X&Ys)

```

Color Space                  : Uncalibrated
Serial Number                : misc21.jpg : 686578285826597329
Image Width                  : 900
Image Height                 : 150

```

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	00	68	65	78	28	58	26	59	73	29					

刚好exif信息里还有两组XY，分开hex再合起来就是flag（一开始合起来hex的，怎么也不对，后来问了八神师傅）

```

EXIF Byte Order          : Big-endian (Motorola, M
X Resolution              : 3902939465
Y Resolution              : 2371618619
Page Name                 : https://ctf.show/
X Position                : 1082452817
Y Position                : 2980145261
  
```

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

## misc22

又是直接在strings就出来了，在photoshop:LayerName中，据八神师傅说原意不是这个，不知道原本是咋做的

## misc23

给了个psd文件，提示flag在时间中，再次用exiftool看一下

```

Modify Date              : 2021:03:23 16:02:30+08:00
Document ID              : xmp.did:49520599-6932-e144-8f4b-dfd5873be5bc
History Action           : ctfshow{, UnixTimestamp, DECtoHEX, getflag
History Instance ID     : xmp.iid:1, xmp.iid:2, xmp.iid:3, xmp.iid:4
History Software Agent   : Adobe Photoshop CC 2019 (Windows), Adobe Photo
shop CC 2019 (Windows), Adobe Photoshop CC 2019 (Windows), Adobe Photoshop CC 20
19 (Windows)
History When             : 1997:09:22 02:17:02+08:00, 2055:07:15 12:14:48
+08:00, 2038:05:05 16:50:45+08:00, 1984:08:03 18:41:46+08:00
History Changed          : /
  
```

发现history action一栏有提示，时间戳转hex就是flag，遂将history when一栏中的4个时间转为数字再转为hex

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

## misc24

提示flag在图片上面。一般的图片藏flag都会将flag藏在图片下面，如果图片是png，只需要将高度修改即可，然而此题给的是bmp文件，像素信息直接就是在图片的hex中，所以存在将flag藏在图片上面的可能。

打开图片属性，发现图片大小是900\*150，即135000个像素，而图片本身去掉文件头后应该是675000/3=225000个像素，明显多了很多，于是猜测图片高度，发现250\*900刚好等于225000个像素，将图片的高度改为250后即可得到flag

即6-9位修改位FA 00 00 00即可

	0	1	2	3	4	5	6	7	8	9	A
	42	4D	F0	4C	0A	00	00	00	00	00	36
	00	00	84	03	00	00	FA	00	00	00	01

## misc25

随便拉长png高度即可，flag在图片下面

## misc26

首先拉长图片高度，发现flag确实在图片下面，但是其中两位是图片的正确高度，脚本爆破下高度即可

```
import os
import binascii
import struct

misc = open("misc26.png", "rb").read()

for i in range(1000000):
    data = misc[12:20] + struct.pack('>i', i) + misc[24:29]
    crc32 = binascii.crc32(data) & 0xffffffff
    if crc32 == 0xEC9CCBC6:
        print(i)
        print("hex:" + hex(i)) # 转为16进制
```

## misc27

flag在图片下面，修改jpg高度即可，也是直接搜索150的hex值的位置就行了，无论啥图片

，应该都可以在winhex里搜宽高所代表的的hex值

## misc28

flag在图片下面，修改gif的高度即可，需要注意的是，gif的每一帧都有宽高所以修改的地方不止一处

## misc29

同上一题，将每一帧的高度都改掉，flag在某一帧里

---

{there\_is\_no\_flag\_here}

ctfshow{03ce5be6d60a4b3c7465ab9410801440}

## misc30

50的hex值的位置就行了，无论啥图片

，应该都可以在winhex里搜宽高所代表的的hex值

## misc28

flag在图片下面，修改gif的高度即可，需要注意的是，gif的每一帧都有宽高所以修改的地方不止一处

## misc29

同上一题，将每一帧的高度都改掉，flag在某一帧里

[外链图片转存中...(img-VD7OPM1f-1617179348206)]

## misc30

将宽度改为950即可