

CTFshow刷题日记-MISC-图片篇(上)基础操作和信息附加

原创

OceanSec 于 2021-09-09 22:54:18 发布 609 收藏 9

分类专栏: #CTF 文章标签: python misc ctf

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/q20010619/article/details/120211803>

版权



[CTF 专栏收录该内容](#)

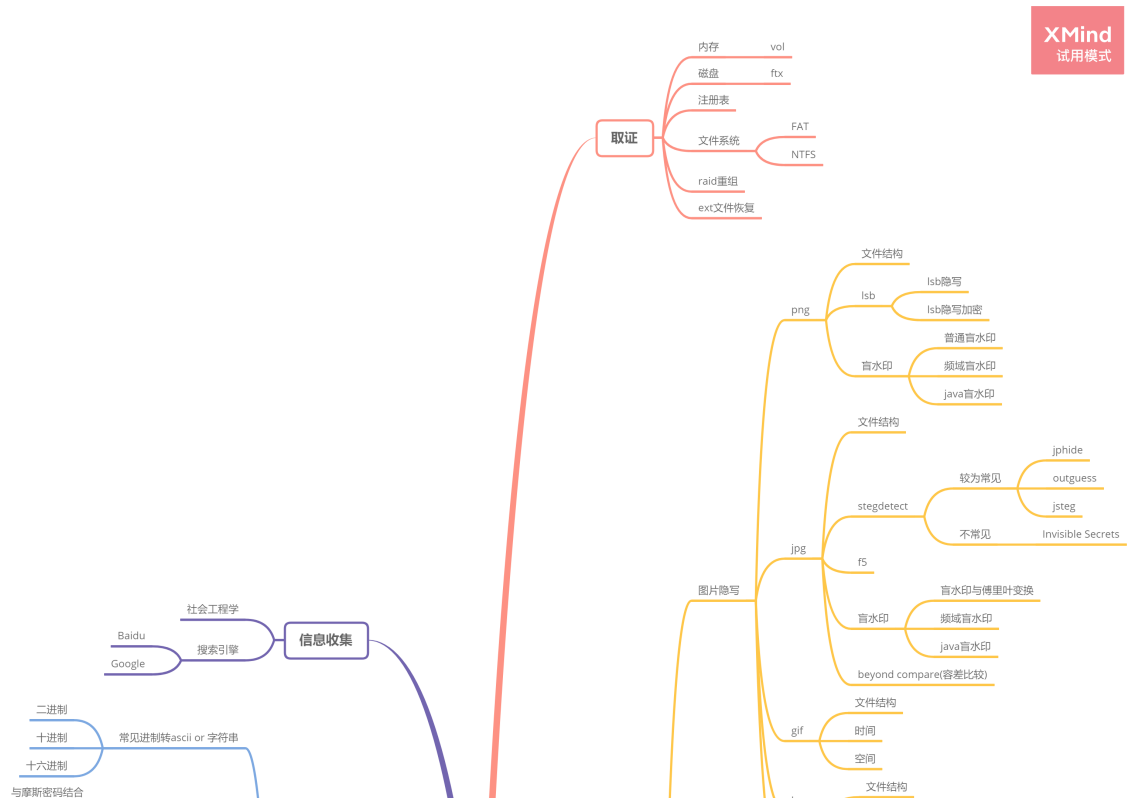
66 篇文章 29 订阅

订阅专栏

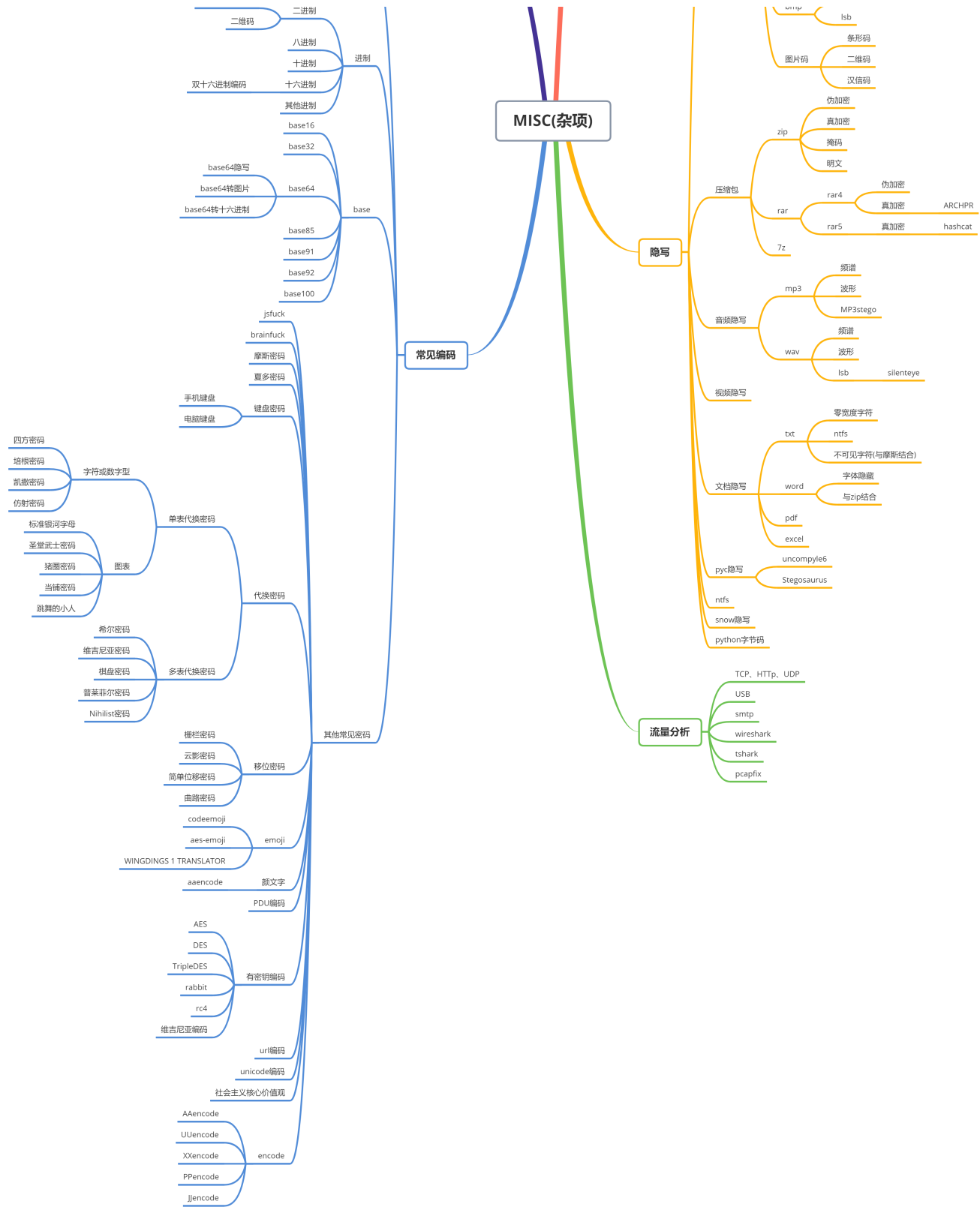
ctfshow 图片篇引语

- 大部分题目仅涉及单一知识点, 但可能有多种解法;
- 找到flag并不困难, 关键是了解每一题背后的原理;
- 藏在哪儿? 为什么可以这样藏? 请多考虑这两个问题;

misc脑图-misc之神丁神给的



XMind 试用模式



基础操作

misc1-签到题

打开图片就是flag

misc2-改后缀

下载之后是一段文本，打开发现png的文件头，将txt后缀改成png，查看图片，获得flag

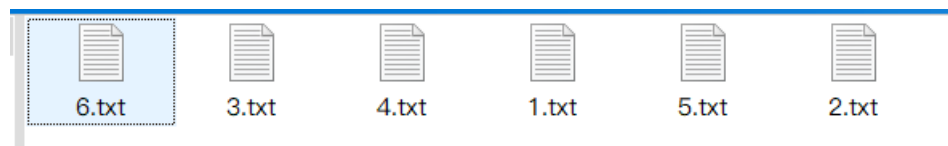
misc3-bpg格式

下载我就直接打开了，我还寻思就这？，原来是我的图片查看器帮了忙，bpg文件，推荐蜂蜜浏览器

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-ChXAbQGw-1631199055893)(F:_笔记\mdpic1.CTFshow刷题日记-MISC-图片篇(上)] - 副本\image-20210909170530982.png)

misc4-改后缀

下载发现是六个文档



然后每个文档打开都是看似乱码的六种不同格式的图片，只要根据其文件头将其改为正确的格式即可

从1-6分别是png, jpg, bmp, gif, tif, webp

每张图都是flag的一部分

信息附加

misc5-7-010直接搜

010打开，搜ctfshow找到flag

misc8-分离文件

提示：flag在图片文件中图片文件中。

给的是png，根据提示猜测可能是改了高度，或者要分离文件，该高度无果，在kali中使用分离命令foremost分离图片，成功分离出flag

misc9-010直接搜

提示：flag在图片块里

010打开，搜ctfshow找到flag

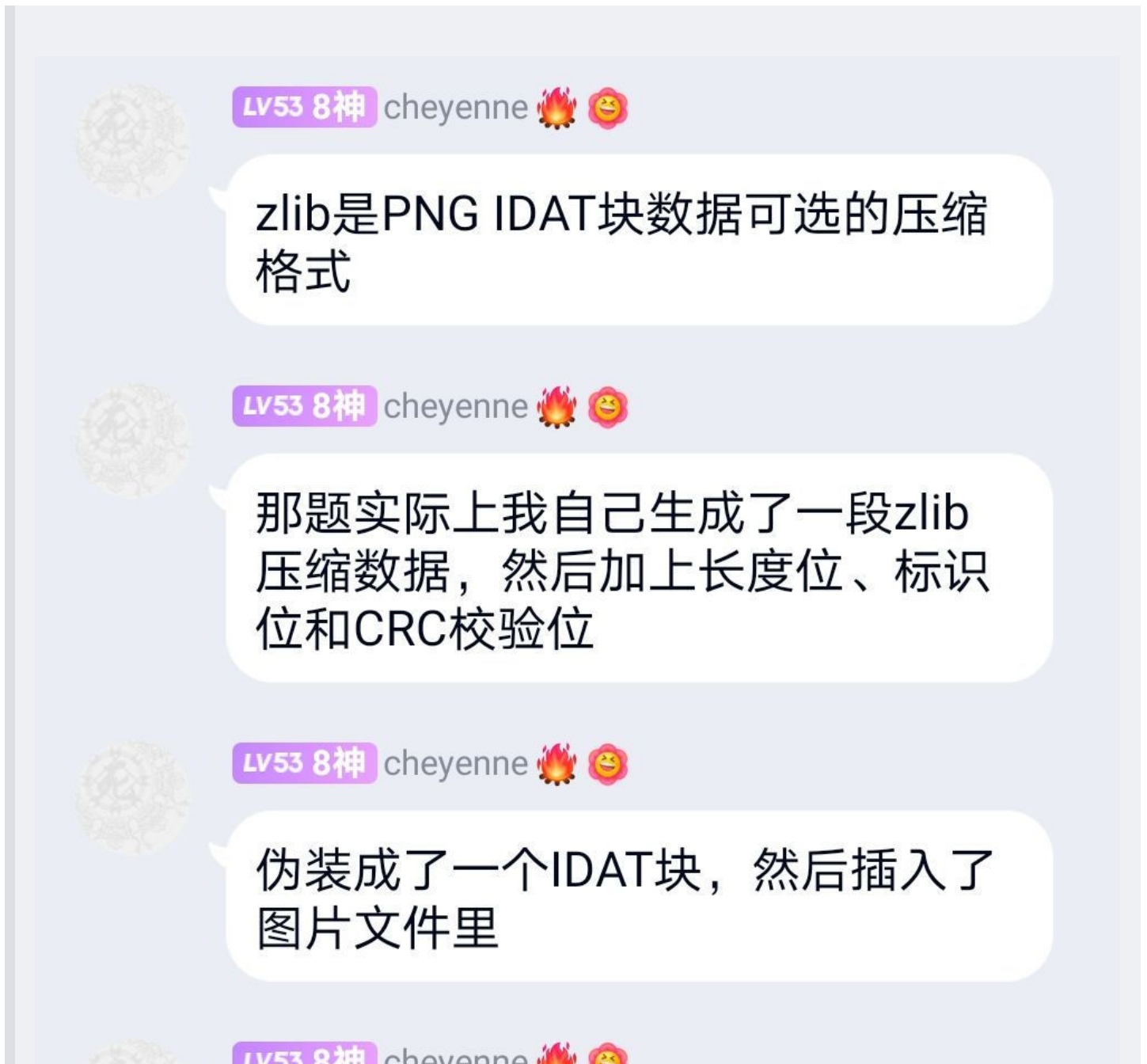
misc10-分离文件

提示：flag在图片数据里

使用binwalk -e命令分离文件



原理-引用z.volcano师傅博客



沐秋的清晨 下午6:24

阿这

来自其他聊天

@沐秋的清晨 binwalk可以一把梭，是因为binwalk会找到zlib块的标记然后提取出来，同时因为这是个压缩数据，binwalk的-e参数会自动把提取到的压缩包尝试进行解压，所以最后的提取结果里就有原始的那段文本，就是flag了

https://pic.af.bsgjrcdn.com/welqj_48890869

misc11-双IDAT块

提示：flag在另一张图里

IDAT

图像数据块IDAT(image data chunk)：它存储实际的数据，在数据流中可包含多个连续顺序的图像数据块。IDAT存放着图像真正的数据信息，因此，如果能够了解IDAT的结构，我们就可以很方便的生成PNG图像。

010打开发现有两个IDAT数据块

| | | | | | |
|-----------------------------|---|-------|-------|-----|-----|
| > struct PNG_SIGNATURE sig | | 0h | 8h | Fg: | Bg: |
| > struct PNG_CHUNK chunk[0] | IHDR (Critical, Public, Unsafe to Copy) | 8h | 19h | Fg: | Bg: |
| > struct PNG_CHUNK chunk[1] | IDAT (Critical, Public, Unsafe to Copy) | 21h | B7Fh | Fg: | Bg: |
| > struct PNG_CHUNK chunk[2] | IDAT (Critical, Public, Unsafe to Copy) | BA0h | 1D81h | Fg: | Bg: |
| > struct PNG_CHUNK chunk[3] | IEND (Critical, Public, Unsafe to Copy) | 2921h | Ch | Fg: | Bg: |

把第一段IDAT删除，另存为图片内容就是flag

misc12-多IDAT块

提示：flag在另一张图里

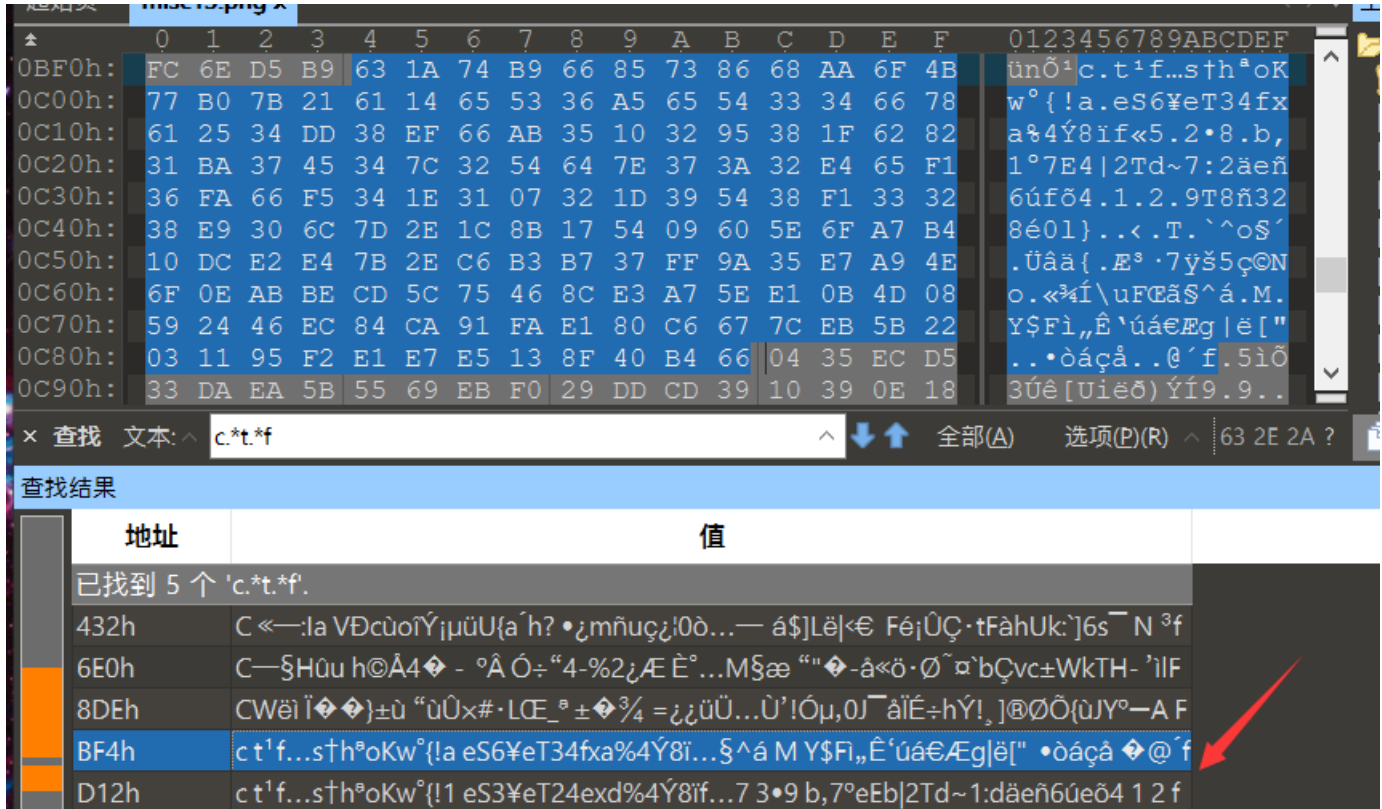
打开发现有十多个IDAT，可以使用tweakpng这个工具边删边保存，发现删除前八个flag就出来了

misc13

试了多种方法没解出来

在010使用正则表达式搜索 `c.*t.*f`

发现两个有点像flag的字符串



flag格式是ctfshow很明显这是每隔一个字符有一个垃圾字符，python简单处理下

```
str = "ct¹f...s†h³oKw°{!aeS6¥eT34fxa%4Y8if«5.2•8b,1°7E4|2Td~7:2äeñ6úfõ4129T8ñ328é01}"
flag = ""
for i in range(len(str)):
    if (i%2==0):
        flag = flag + str[i]
print(flag)
```

运行得到flag

```
PS C:\... python .\test.py
ctfshow{ae6e3fa48f528b1742d72e6f41298380}
```

misc14-手动分离图片

jpg格式的图片

在kali中使用命令却分不出来，使用010

在0x837位置发现另一张jpg图片，从837选择到末尾，右键选择->保存选择

```
07F0h: 00 01 96 01 1B 00 05 00 00 00 01 00 00 01 9E 01  ..-.....ž.
0800h: 28 00 03 00 00 00 01 00 02 00 00 02 01 00 04 00  (.....
0810h: 00 00 01 00 00 01 A6 02 02 00 04 00 00 00 01 00  ..|.....
0820h: 00 04 D5 00 00 00 00 00 00 00 48 00 00 00 01 00  ..Ö.....H....
0830h: 00 00 48 00 00 00 01 FF D8 FF E0 00 10 4A 46 49  ..H....ÿøÿà..JFI
0840h: 46 00 01 01 01 00 78 00 8 00 00 FF DB 00 43 00  F.....x.x..ÿÛ.C.
0850h: 02 01 01 02 01 01 02 02 02 02 02 02 02 03 05  (.....
0860h: 03 03 03 03 03 06 04 04 03 05 07 06 07 07 07 06  ..
0870h: 07 07 08 09 0B 09 08 08 0A 08 07 07 0A 0D 0A 0A  ..
0880h: 0B 0C 0C 0C 0C 07 09 0E 0F 0D 0C 0E 0B 0C 0C 0C  ..
0890h: FF DB 00 43 01 02 02 02 03 03 03 06 03 03 06 0C  ÿÛ.C.....
08A0h: 08 07 08 06 06 06 06 06 06 06 06 06 06 06 06 06  ..
```

保存生成一个新文件，即为flag

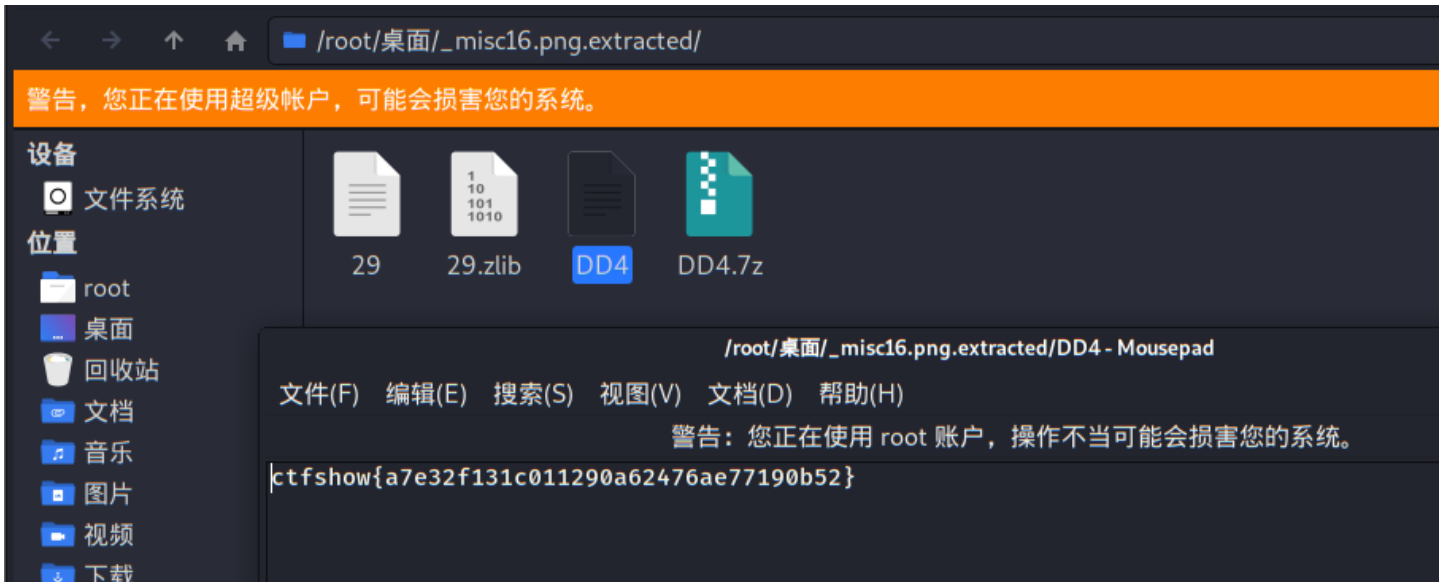
misc15-010一把梭

提示：flag被跳过去了

不知道啥意思但是010一把撸



misc16-binwalk一把梭



misc17-steg

提示：flag在图片数据里

使用zsteg命令发现隐藏的数据

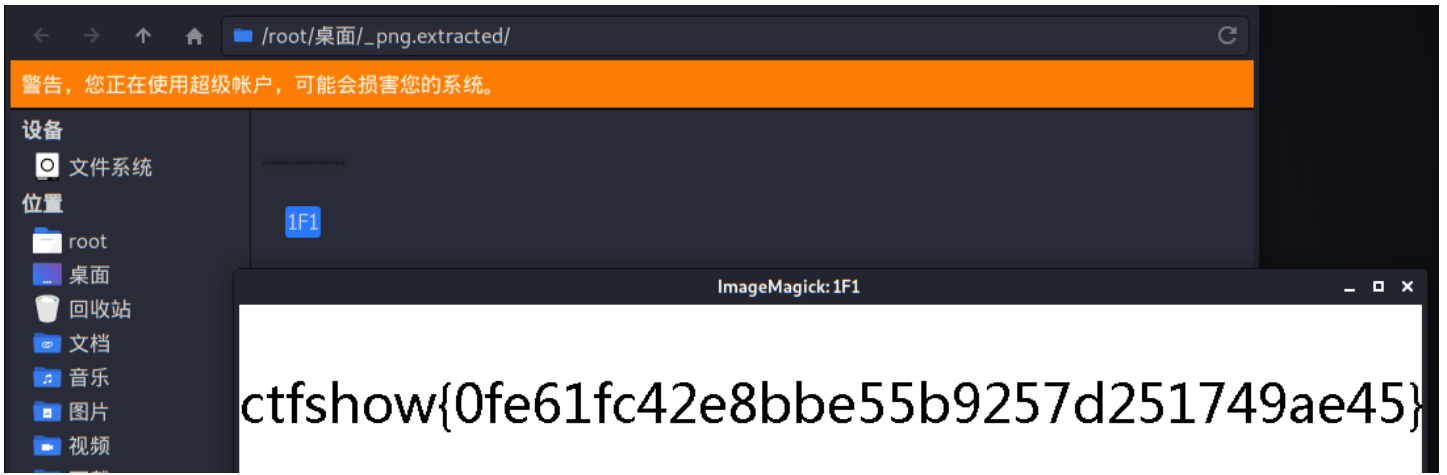
```
root@kali:~/桌面# zsteg misc17.png
[?] 3544 bytes of extra data after zlib stream
extradata:0
00000000: e1 1f 30 53 86 4f c5 a4 1b f5 e6 e5 c7 46 0a 92 |..0S.O.....F..|
00000010: 9b ee 72 e7 c9 9e b9 a7 74 de 92 4d ad 61 5b 58 |..r.....t..M.a[X|
00000020: f2 98 65 77 2b d2 d3 85 32 fc 08 83 86 1f 0f 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...P|
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 |... [.O.x.]F..kk*|
00000070: f2 62 0c ba 70 19 a0 27 f3 84 77 99 02 77 05 79 |.b..p.. ' ..w..w.y|
00000080: 5b 44 b7 79 b3 54 11 a1 f3 54 34 56 7e ff 55 d1 |[D.y.T...T4V~.U.|
00000090: c6 39 90 c8 21 7f 26 39 44 58 78 c3 ed 37 4a 7c |.9..!.89DXx..7J||
000000a0: 50 24 e8 79 7b 4b 9c fa 2a 2c bb e8 b9 fb 40 2c |P$.y{K..*,...@,|
000000b0: 50 05 21 4c 3b 29 65 b4 60 1c 27 bb 4c 16 bf f1 |P.!L;)e.`.' .L...|
000000c0: 77 c0 55 04 5e 25 0e 18 1e 58 ab 0f 13 11 f2 3f |w.U.^%...X.....?|
000000d0: cf a0 32 b1 f5 a8 1b 99 a7 4b 46 89 cf 85 89 50 |..2.....KF....P|
000000e0: 88 20 8f 4f fd e2 97 55 68 73 b4 96 ba dd 25 a3 |. .O...Uhs....%.|
000000f0: 83 72 3f 99 77 9e 0a 08 50 4f 11 8f 87 65 c0 29 |.r?.w...PO...e.)|
```

使用命令提取数据

```
zsteg -E "extradata:0" ./misc17.png > ./png
// 名字随便定义
```

使用命令去分离

```
binwalk -e png
```

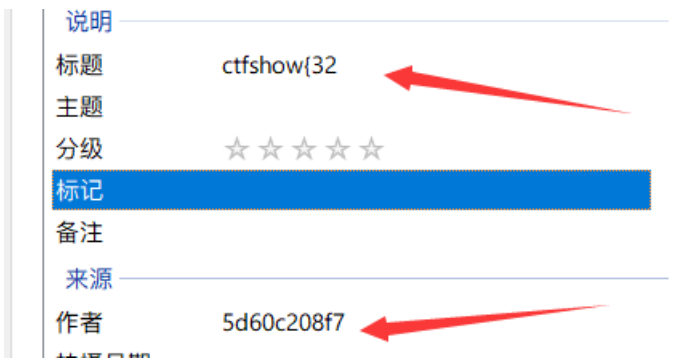



misc18-图片属性

提取: flag在标题、作者、照相机和镜头型号里

简单肯定在属性里边

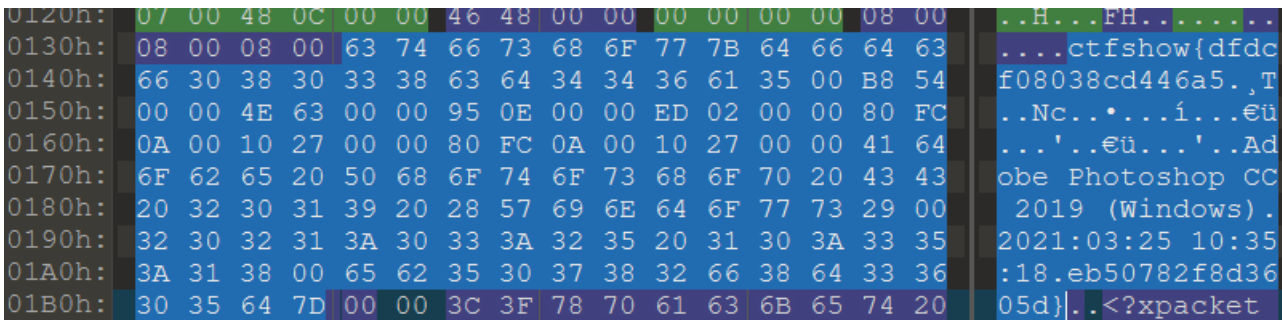
flag分成了三段



misc19-010

提示: flag在主机上的文档名里

010一把梭



两段拼一下就有了

misc20-exiftool工具

提示: flag在评论里

可以使用工具exiftool

```
PS > exiftool misc20.jpg
ExifTool Version Number      : 10.25
File Name                    : misc20.jpg
Directory                   : .
File Size                    : 14 kB
File Modification Date/Time  : 2021:03:24 16:32:48+08:00
File Access Date/Time       : 2021:09:09 21:31:04+08:00
File Creation Date/Time     : 2021:09:09 21:29:41+08:00
File Permissions             : rw-rw-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                      : 杩樫混閽困筧澶o耗鑛嬧筧鉈俗澈鑛o細瓊垮冷鑛辨缺纒€澶o婢葑瘋夕浚灑韞浚潰段鍥涔維涓€€璇舵
橋瓊跨塌錫承洪鑛o橋涓€€寮熲韞浚澹澹夕浜岙嶼察燭紛瓊踪紛浜飾節涓�橋鍥澹涓€€澶o婢葑?
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
-- press RETURN --
```

但是环境问题是乱码, 在linux下查看

```
root@kali:~/桌面# exiftool misc20.jpg
ExifTool Version Number      : 12.06
File Name                    : misc20.jpg
Directory                   : .
File Size                    : 14 kB
File Modification Date/Time  : 2021:03:24 04:32:48-04:00
File Access Date/Time       : 2021:09:09 09:33:02-04:00
File Inode Change Date/Time : 2021:09:09 09:33:02-04:00
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)
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谩易西爱抚零六易一第七九西二一第弟谩第五九三易
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
```

如果没有exiftool, 可以安装, 教程

这flag需要译一下

misc21-转换

提示: flag在序号里

同样使用exiftool命令查看

```
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
```

发现序列号提交不正确，经过十六进制转码发现问题

Ascii Encoding

Text

hex(X&Ys)

Bin

1101000 1100101 1111000 101000 1011000 100110 1011001 1110011 101001

Oct

150 145 170 50 130 46 131 163 51

Dec

104 101 120 40 88 38 89 115 41

Hex

68 65 78 28 58 26 59 73|29

也就是说吧x和y都要分别进行转换而不是合一起转换，python的hex()

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-Sao2gnwl-1631199055913)(F:_笔记\mdpic\1.CTFshow刷题日记-MISC-图片篇(上)]\image-20210909214809200.png)

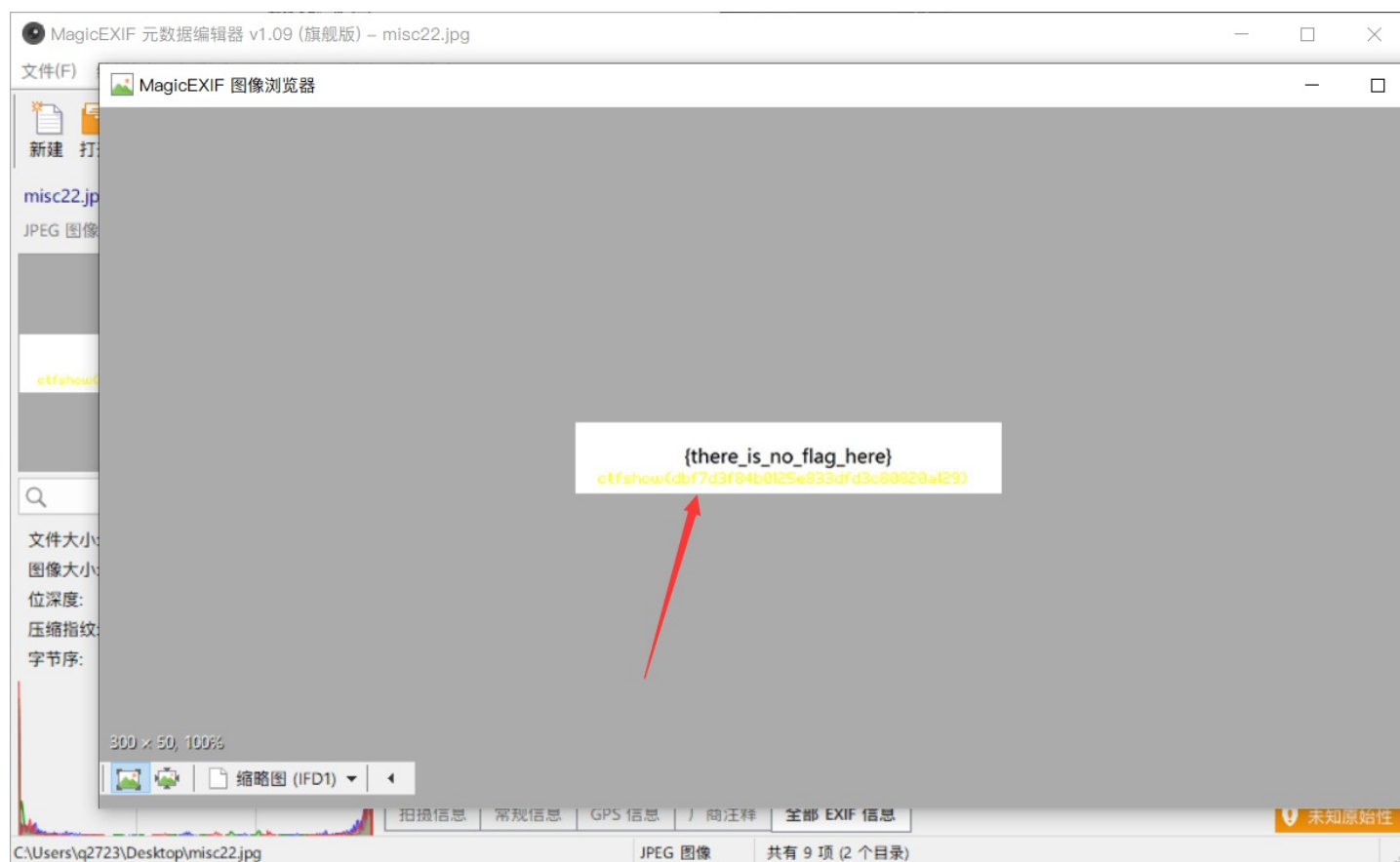
再用ctfshow{}包裹

misc22-缩略图隐写

提示: flag在图片里

缩略图竟然和图片不一样, 学到了

这是magicexif



利用exiftool工具导出图片

```
exiftool.exe -ThumbnailImage -b misc22.jpg > 1.jpg
```

misc23-属性转换

提示: flag在时间里

使用exiftool命令查看信息

```
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
```

发现四个时间, 将时间转换成时间戳网站<https://www.zxgj.cn/g/unix>

再用21题的方法, 进行hex编码

misc41-抽象

提示:

H4ppy Apr1l 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。

愚人节限定题, 下载得到misc41.jpg, 用winhex打开, 发现是jpg的文件尾, 但是文件头对不上, 本来想以这个方向为突破点, 没得到结果... 后面看了套神的wp才知道, 提示中的第二句说的就是我...

第一句提示的F001才是真突破点, 这个位置有大量F001, 看起来组成了某种形状

我的思路是, 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。最后变成下图的8*125的“图”, 其实如果会用CyberChef会更方便, 不过我不太习惯。

```
0 0 0 0
0 0 00

00
0 000
000 0
0 0
00 000

0 0
0 0
000 000
0 0 0 0
000 000

0 0 000
0 0 0
000 000
0 0
0 000

000
```

依稀看出flag: ctfshow{fcbd427caf4a52f1147ab44346cd1cdd}

来自https://blog.csdn.net/weixin_45696568/article/details/115261347

参考链接

[z.volcano](#)