MISC入门总结

原创

想成菜鸡的武阳 已于 2022-04-27 17:00:23 修改 1491 收藏
文章标签： 安全
于 2022-04-19 20:00:01 首次发布

《re入门到Misc精通》，哄堂大笑了，家人们。

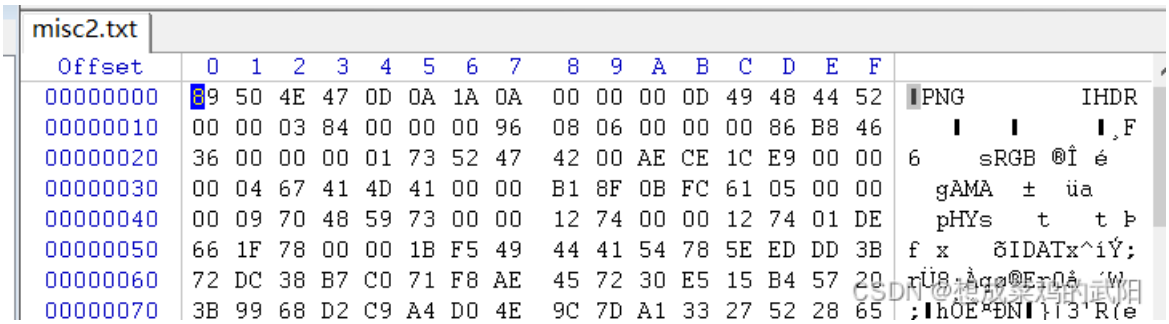
总结在后面

ctf.show https://ctf.show/challenges#misc2-1134

ctfshow misc入门题（还没写完，持续更新）

图片篇第一题 签到题 打开直接给flag qq提取文字交了

第二题 给个txt文件 winhex打开txt文件后 发现



png头 所以 换成png后缀 打开后给flag 文字识别 提交

第三题 bpg格式 cmd进入查看bgp文件中 命令为：bpgview.exe E:\ctfshow\misc3.bpg

第四题 给了四个txt winhex进入发现后缀不对 改了后每个图 对应一部分 都改成png 就可以看到了

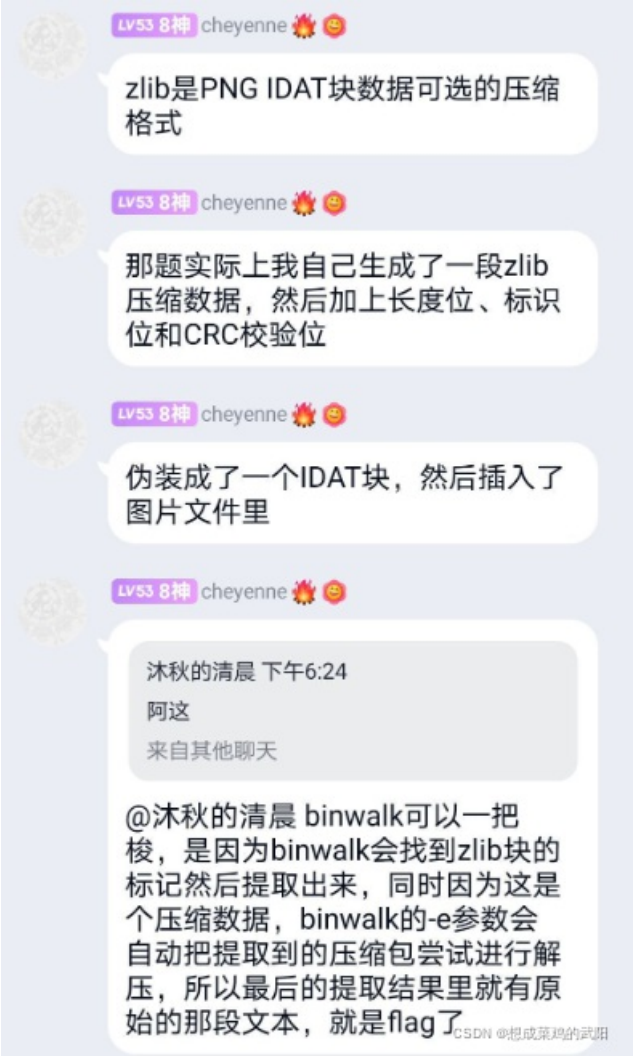
第五题 打开图片 发现假flag winhex找到后面连接的flag

第六题 打开winhex 搜文字 ctfshow即可得到

第七题：同上

第八题：**flag在图片文件中图片文件中。 kali用foremost命令分离出来** 藏了一个png 开始标志 89 50 结束标志 60 82

第九题：同第七题

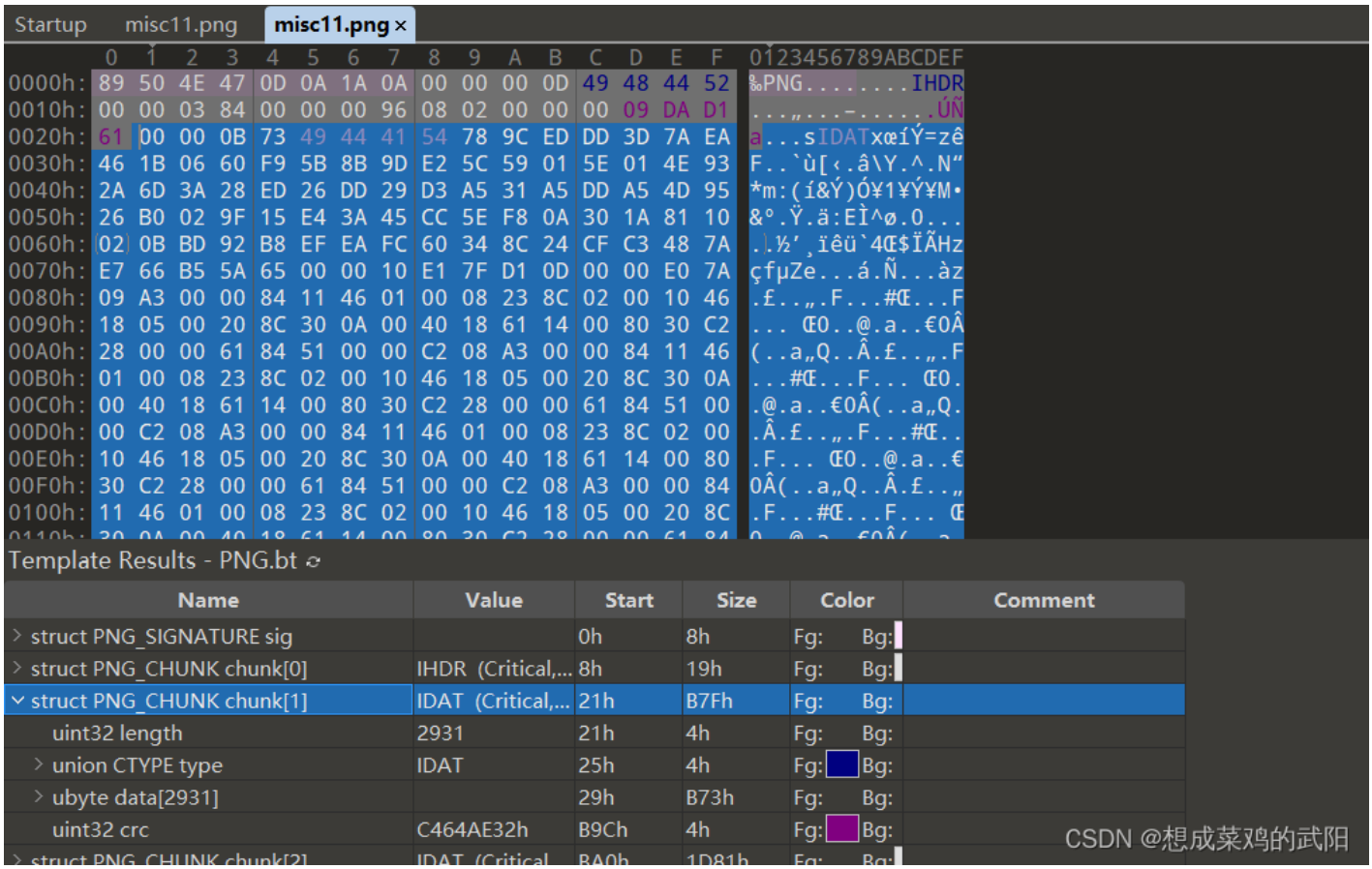


第十题：

第十一题：**flag在另一张图里**

PNG中IDAT

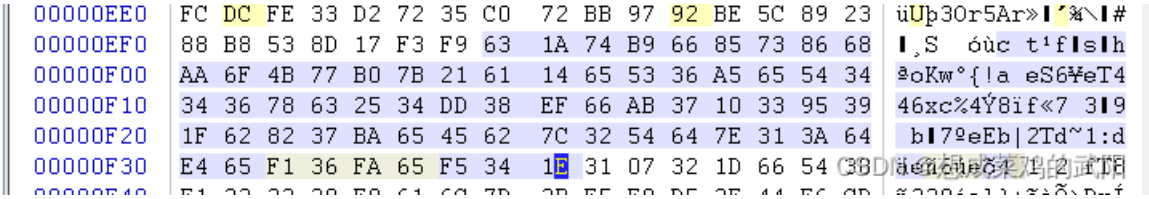
图像数据块IDAT(image data chunk)：它存储实际的数据，在数据流中可包含多个连续顺序的图像数据块。
binwalk看到的zlib就是其压缩格式 binwalk -e可以实现自动解压



用010把第一个idat删除了 然后另存为图 即可求出

第十二题：同上 发现idat过多 删除前8个idat就好

第十三题：提示在后面 发现



查看发现 中间隔两个字符

```
s="631A74B96685738668AA6F4B77B07B216114655336A56554333465786125340D38EF66AB35103195381
flag=""
for i in range(0,len(s),4):#相当于四个数字个循环，只要前两个
    flag += s[i]
    flag += s[i+1]
print(flag)
```

得到十六进制 转换 16进制转换，16进制转换文本字符串，在线16进制转换 | 在线工具 (sojson.com)

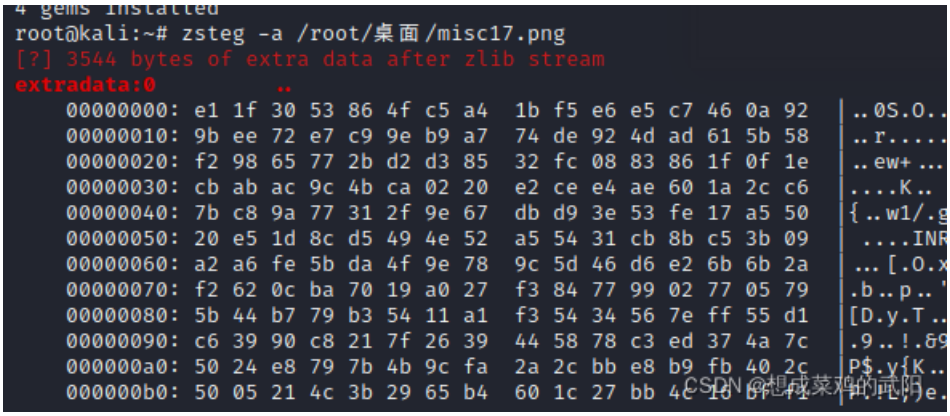
第十四题：**flag在图片文件中图片文件中。 kali用foremost命令分离出来** 藏了一个png 开始标志 89 50 结束标志 60 82

第十五题：winhex打开就发现了flag

第十六题：winhex打开时发现有大量IDAT块 提示flag在图片数据里

有zlib文件 见第十题 binwalk -e分离 得到flag

第十七题：zsteg，这是一个用于检测PNG和BMP中的隐藏数据隐藏数据的工具，可以快速提取隐藏信息 (86条消息) Kali linux下图片隐写,图片隐写信息快速检测工具——zsteg_老魏一凡的博客-CSDN博客
https://blog.csdn.net/weixin_35696092/article/details/116953571?ops_request_misc=&request_id=&biz_id=102&utm_term=kali%E5%A6%82%E4%BD%95%E8%A3%85zsteg%task-blog-2~all~sobaiduweb~default-0-116953571.142%5Ev9%5Econtrol,157%5Ev4%5Econtrol&spm=1018.2226.3001.4187



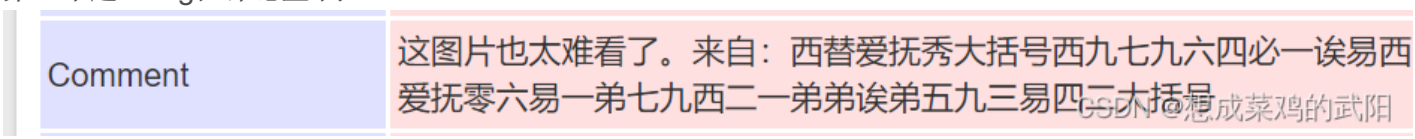
zsteg -E /root/桌面/misc17.png 'extradata:0' > 1.txt

binwalk -e /root/桌面/1.txt 得到

第十八题：提示**flag在标题、作者、照相机和镜头型号里 鼠标右键 属性**

第十九题： **flag在主机上的文档名里 属性里没** EXIF信息查看器 (tuchong.com)

第二十题：flag在评论里 同上



谐音梗扣钱 ctfshow{c97964b1aecf06e1d79c21ddad593e42}

第二十一题：flag在序列号上 同上 找到序列号一串数字

发现是数字 转下字符串 发现是十六进制 转换下

16进制转换文本 / 文本转16进制

686578285826597329

字符串转16进制 >>

hex(X&Ys)

python中hex是把十进制转十六进制 把x的分辨率与Y的分辨率再相加

十进制转十六进制| 10进制转16进制 | 在线进制转换 (sojson.com)

X分辨率	3902939465
Y分辨率	2371618619
PageName	https://ctf.show/
X定位	1082452817
Y定位	2980145261
目标Printer	ctfshow{}

四个值转换后相加 即可得到flag

3902939465 转换

进制	结果
二进制	11101000101000100010000
四进制	3220220202011021
八进制	35050420511
十进制	3902939465
十六进制	e8a22149

ctfshow{e8a221498d5c073b4084eb51b1a1686d}

第二十二题：

提示 **flag在图片里 winhex打开后搜索ctfshow 没有发现**

学习到了一个新的jpeg压缩 我好菜啊 MagicEXIF 元数据编辑器_官方电脑版 _华军纯净下载 (onlinedown.net)

下载后打开就出来了

ctfshow{dbf7d3f84b0125e833dfd3c80820a129}

第二十三题：**flag在时间里**

给了一个psd文件

EXIF信息查看器无需安装软件，只需上传照片即可查看完整EXIF信息，包括机身、镜头型号、拍摄时间、相机快门次数，支持JPEG、TIFF、CR2、NEF、XMP等多种图片格式。无需下载，比Exif Show, ExifPro更好用的EXIF查看器！ https://exif.tuchong.com/查看时间

XMP-photoshop

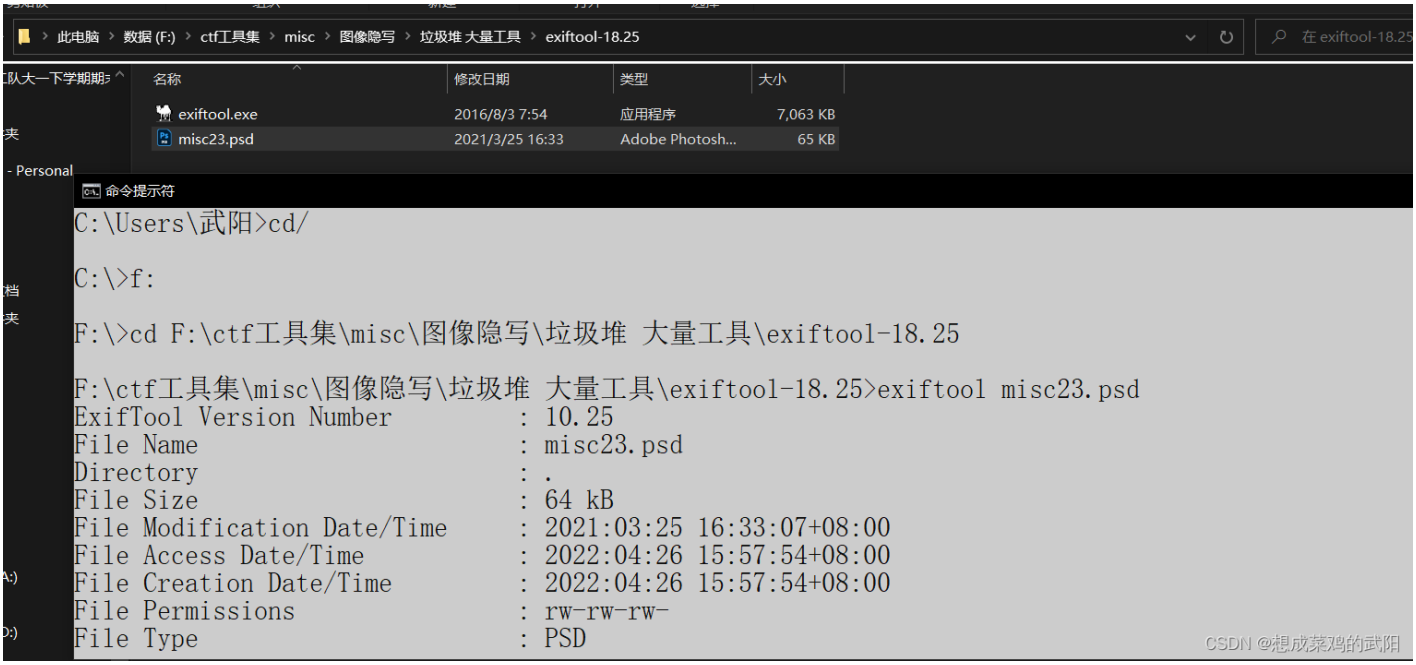
色彩模式	RGB
TextLayerName	{there is no flag here}
TextLayerText	{there is no flag here}

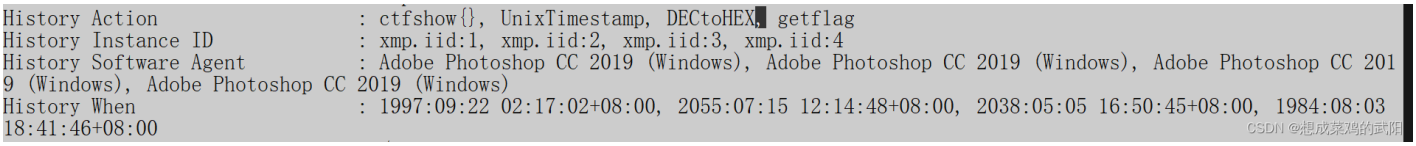
XMP-xmp

创建日期	2021:03:25 15:45:24+08:00
Creator工具	Adobe Photoshop CC 2019 (Windows)
元数据Date	2021:03:25 16:02:50+08:00
修改日期	2021:03:25 16:02:50+08:00

看他人做法 又学到了新的工具 exiftools

ExifTool完全入门指南 (rmnof.com)https://www.rmnof.com/article/exiftool-introduction/





可以看到action 和history

要求是先转成unix时间戳 然后DEC十进制转HEX十六进制就得到flag

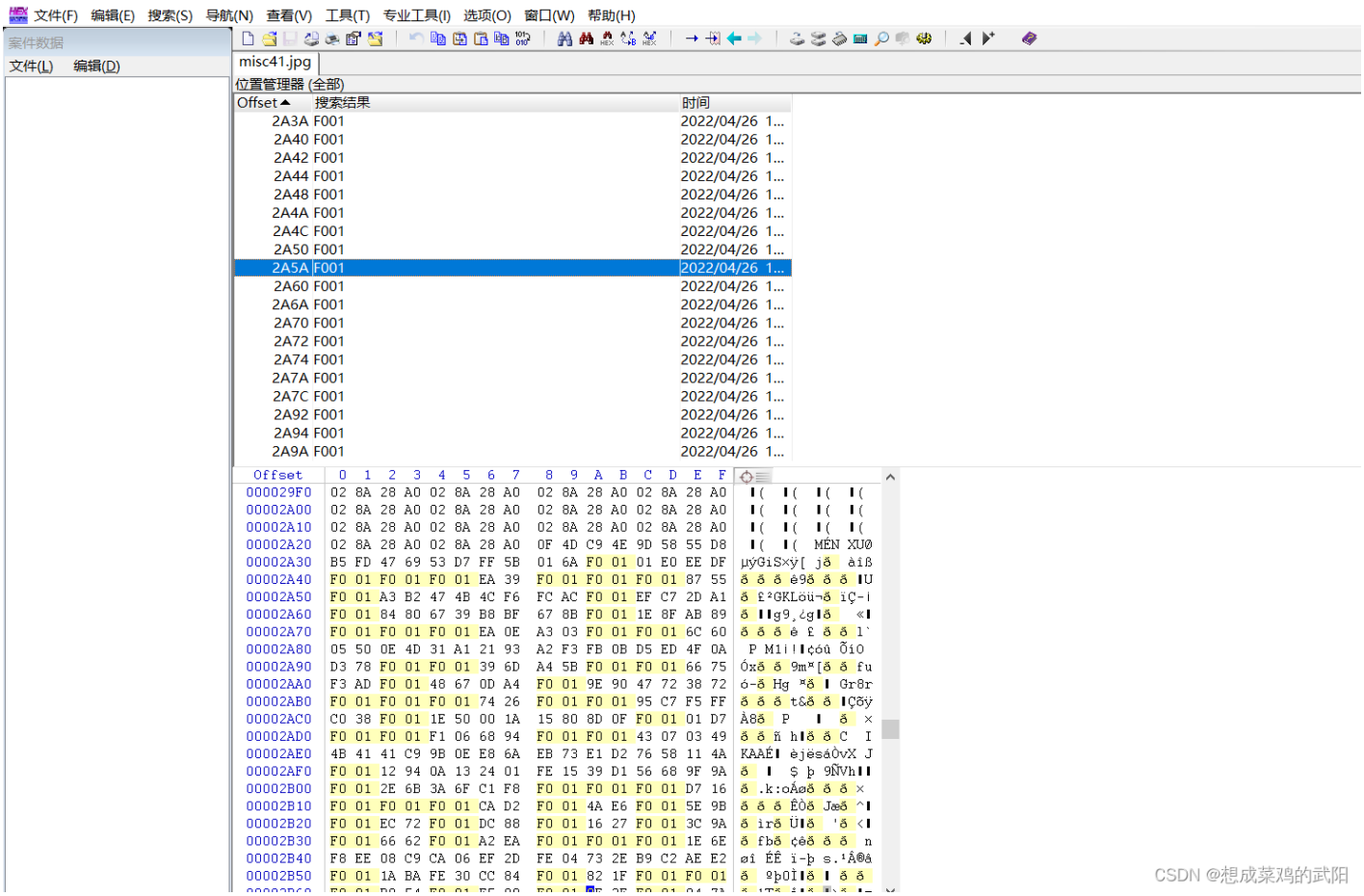
时间戳(Unix timestamp)转换工具 - 在线工具 (tool.lu)https://tool.lu/timestamp/

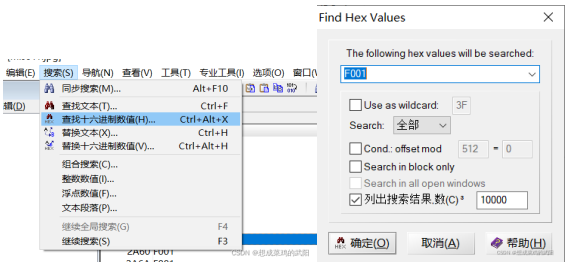
十进制转十六进制| 10进制转16进制 | 在线进制转换 (sojson.com)
https://www.sojson.com/hexconvert/10to16.html

合并就好 ctfshow{3425649ea0e31938808c0de51b70ce6a}

misc41：

太坑了 竟然是





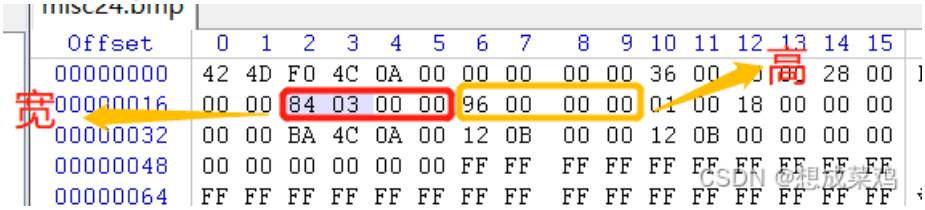
H4ppy Apr1l F001's D4y！ F001

得到的是ctfshow{fcbd427caf4a52f1147ab44346cd1cdd}

misc24

flag在图片上面。

(90条消息) 位图(bmp)文件格式分析_aidem_brown的博客-CSDN博客_bmp位图
https://blog.csdn.net/aidem_brown/article/details/80500637



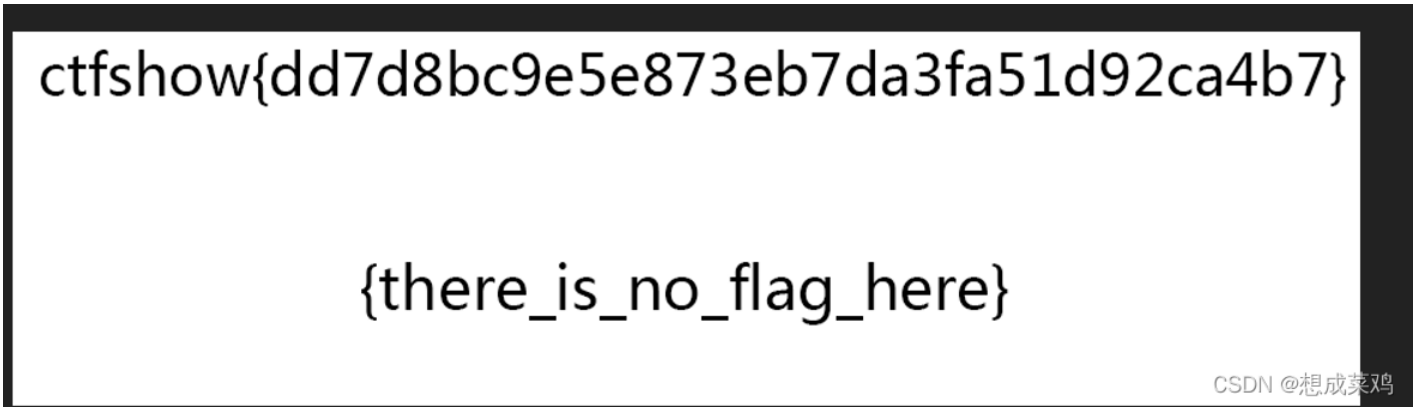
转换后得到宽为900

最后得出目前文件是900*150=135000个像素大小 同时文件头占53字节

文件尾的位置在675053字节处(后面两个字节是windows的”**补0**”)，又因为**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`

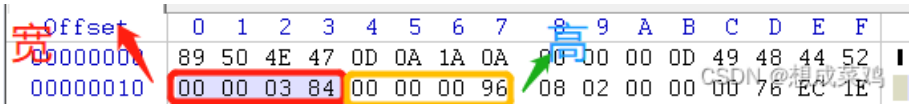
提示高 所以正确的高度是225000/900=250

即改为FA 00 00 00



misc25

png格式与bmp不同，他的宽高部分不一样 且不需要大端优先 也没那么复杂 直接改



详解PNG文件结构 - Angel_Kitty - 博客园 (cnblogs.com)https://www.cnblogs.com/ECJTUACM-873284962/p/8986391.html**flag在图片下面。 改高度 00 00 00 96改成00 00 00 F2就行**

misc26

flag在下面 但多下面 需要CRC爆破

对一张正常的图片，通过修改其宽度或者高度隐藏信息，使计算出的CRC校验码与原图的CRC校验码不一致；windows的图片查看器会忽略错误的CRC校验码，因此会显示图片，但此时的图片已经是修改过的，所以会有显示不全或扭曲等情况，借此可以隐藏信息。

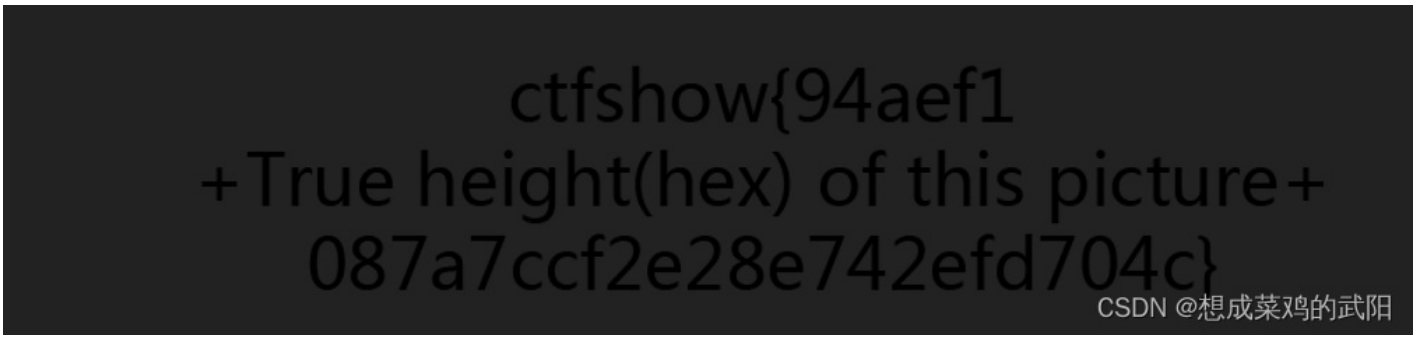
而Linux下的图片查看器不会忽略错误的CRC校验码，因此用Linux打开修改过宽或高的png图片时，会出现打不开的情况

爆破图片修改前的宽和高来匹配CRC校验码，并用正确的宽和高来修复图片

```
import zlib
import struct

filename = 'misc26.png'
with open(filename, 'rb') as f:
    all_b = f.read()
    crc32key = int(all_b[29:33].hex(),16)
    data = bytearray(all_b[12:29])
    n = 4095            #理论上0xffffffff,但考虑到屏幕实际/cpu，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]
            crc32result = zlib.crc32(data)
            if crc32result == crc32key:
                print("宽为：",end="")
                print(width)
                print("高为：",end="")
                print(height)
                exit(0)
```

然后winhex修改 得到



ctfshow{94aef125e087a7ccf2e28e742efd704c}

misc27 **flag在图片下面**

给了个jpg 所以没法爆破 直接改高 他的和png还不一样 需要查找

右键点击图片 选择属性 打开

他的高度为150像素 改成16进制为96 宽改后为03 84 winhex搜96 0384

Offset ▲	搜索结果	时间
9E	96	2022/04/27 1...
9F	0384	2022/04/27 1...
165	96	2022/04/27 1...
21A	96	2022/04/27 1...
825	96	2022/04/27 1...
8BB	96	2022/04/27 1...
8C5	96	2022/04/27 1...
96E	96	2022/04/27 1...
AD3	96	2022/04/27 1...
BC7	96	2022/04/27 1...
BCB	96	2022/04/27 1...
C5E	96	2022/04/27 1...
C71	96	2022/04/27 1...
DE9	96	2022/04/27 1...
F99	96	2022/04/27 1...
FFF	96	2022/04/27 1...
1062	96	2022/04/27 1...
10ED	96	2022/04/27 1...
1440	96	2022/04/27 1...

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	FF	D8	FF	EE	00	0E	41	64	6F	62	65	00	64	40	00	00
00000010	00	01	FF	DB	00	84	00	02	02	02	02	02	02	02	02	02
00000020	02	03	02	02	02	03	04	03	02	02	03	04	05	04	04	04
00000030	04	04	05	06	05	05	05	05	05	05	06	06	07	07	08	07
00000040	07	06	09	09	0A	0A	09	09	0C	0C	0C	0C	0C	0C	0C	0C
00000050	0C	0C	0C	0C	0C	0C	0C	01	03	03	03	05	04	05	09	06
00000060	06	09	0D	0A	09	0A	0D	0F	0E	0E	0E	0E	0F	0F	0C	0C
00000070	0C	0C	0C	0F	0F	0C	0C	0C	0C	0C	0C	0F	0C	0C	0C	0C
00000080	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
00000090	0C	0C	0C	0C	0C	0C	0C	0C	FF	C0	00	11	08	00	96	03
000000A0	84	03	01	11	00	02	11	01	03	11	01	FF	DD	00	04	00
000000B0	71	FF	C4	01	A2	00	00	00	07	01	01	01	01	01	00	00
000000C0	00	00	00	00	00	00	04	05	03	02	06	01	00	07	08	09

把96前的00块改为01得到 ctfshow{5cc4f19eb01705b99bf41492430a1a14}

总结：

- 1.查看图像属性详细信息是否有隐藏内容 或者看exif信息 EXIF信息查看器 (tuchong.com)
- 或者用exiftools
- 2.利用winhex或nodepad++打开搜索ctf,CTF，flag,key等关键字是否存在相关信息
- 搜下有没有txt 有的话 直接扔kali里分解
- 3.检查图像的开头标志和结束标志是否正确，若不正确修改图像标志恢复图像，打开查看是否有flag或ctf信息，（往往gif属于动图，需要分帧查看各帧图像组合所得数据 若不是直接的ctf或flag信息 需要考虑将其解码）看标志位是否损坏 没有的话加 winhex里：右键-》编辑-》粘贴0字节-》插入所需要的位数-》修改那些插入的0字节
- jpg图像开始标志：FF D8 结束标志 ：FF D9
- gif图像开始标志：47 49 46 38 39 61 (GIF89)结束标志：01 01 00 3B
- bmp图片开始标志：42 4D //92 5B 54 00 00 00 00 00 结束标志：00
- png图片开始标志：89 50 结束标志：60 82
- 4.将图片放置在kail系统中，执行binwalk xxx.jpg 查看图片中是否是多个图像组合或者包含其他文件（若存在多幅图像组合用binwalk来找 语法：binwalk -e 文件路径，再执行foremost xxx.jpg会自动分离；若检测出其他文件修改其后缀名即可，如zip）

binwalk -e filename

- 5.使用StegSolve对图像进行分通道扫描，查看是否为LSB隐写
- 6.在kail下切换到F5-steganography，在java Extract运行
- 命令：java Extract 123456.jpg图片的绝对地址 -p 123456
- 判断是否为F5算法隐写
- 7.在kali系统中使用outguess-master工具（需要安装），检测是否为guess算法隐写
- 8.用winhex改变像素
- 其他人的思路：

2.docx文件类型：
（1）文档中含有隐藏文字，选项中设置。
（2）在kali下改为压缩包，看一下含有的隐藏信息。
3.jpg图片文件：
（1）查看属性，notepad++打开，16进制打开看格式，看看有没有关键字。
（2）备份一份，改为zip，看看是否包含其余的文件。
（3）jpg文件kali查看文件，kali下的命令：binwalk 文件名 分离文件：foremost -e 文件名
。
outguess隐写：outguess -r angrybird.jpg angrybird.txt。
steghide：查看隐藏在文件中的信息：steghide info 文件名，分离文件：steghide extract -sf 文件名。
（4）使用stegsolve，查看不同通道，不同偏移量是否含有其余信息。
（5）jpg图片可以使用stegdetect -tjopi -s 10.0 文件名查看是什么隐藏方式。
（6）jpg文件，jphide可以使用steghide解密（jphs），命令：steghide extract -sf 文件名（要密码）
（7）jpg文件下的F5隐写，进入F5-steganography-james文件夹，在空白处 ctrl+shift+鼠标右键->在此处打开命令窗口，在cmd中输入命令：java Extract 文件名 -p 密码；kali下也可以使用java Extract /root/文件名 -p 密码提取F5隐写文件，在F5文件夹中可以找到output.txt
（8）jpg文件可能会用到brainftools分离，命令：bftools.exe decode braincopter 文件名 --output out.jpg之后运行：bftools.exe run out.jpg
4.png文件：
（1）查看属性，notepad++打开，16进制打开看格式，看看有没有关键字。
（2）kali查看文件：binwalk 文件名
分离文件：foremost -e 文件名

（3）使用stegsolve，查看不同通道，不同偏移量是否含有其余信息。
（4）brainftools分离，命令：bftools.exe decode braincopter 文件名 --output out.txt，如果有BrainFuck代码可以运行：bftools.exe decode braincopter 文件名 --output out.jpg，之后运行bftools.exe run out.jpg
（5）steganography软件可以提取文件，选择decrypt为密码提取。（web版steganography：http://www.atool.org/steganography.php）
（6）tweakpng判断是不是png格式，可能校验位有问题。
（7）在十六进制编辑器中修改高度（二行六列）查看隐藏信息。
5.bmp文件：
（1）查看属性，notepad++打开，16进制打开看格式，看看有没有关键字。
（2）kali查看文件：binwalk 文件名
分离文件：foremost -e 文件名

（3）使用stegsolve，查看不同通道，不同偏移量是否含有其余信息。
（4）bmp文件隐写可能是LSB，利用Wbstego解决，生成一个is文件，文本编辑器打开看看。
（5）把图片放到画图里，改成png格式保存，再利用png的隐写查看隐藏信息。
6.gif文件：
（1）notepad++查看文件头GIF8，十六进制打开：修改头文件
（2）使用stegsolve，逐帧查看。
7.zip文件：
（1）密码爆破
8.音频文件：
（1）查看属性，notepad++打开，16进制打开看格式，看看有没有关键字。
（2）MP3stego命令：Decode.exe -X -P 密码 文件名

（3）利用Audacity分析音频文件。
（4）kali查看文件：binwalk 文件名
分离文件：foremost -e 文件名

9.stegsolve

（1）下面右选项表示不同的色素的通道

（2）数据提取，analysis，date extract，按照选项选完（RGB）

（3）图片合成，先打开一张，用中间的analysis，combine，二维码一般四角为黑色，可以进行反色操作（点下面的左右）
10.其余文件：
（1）查看属性，notepad++打开，16进制打开看格式，搜索关键字（flag，ctf，key）。
（2）两张以上的文件可能会使用stegsolve进行合成。
（3）修改zip，查看隐藏文件。
（4）复杂图片可以分离图层。

WinHex: Hex Editor & Disk Editor, Computer Forensics & Data Recovery Software
http://www.winhex.com/winhex/BPG Image format (bellard.org) https://bellard.org/bpg/