ctfshow-Misc入门-图片篇（持续缓慢更新）

原创

bfengj 于 2021-04-08 00:17:08 发布

分类专栏： Go

本文链接：https://blog.csdn.net/rfrder/article/details/115369600

前言

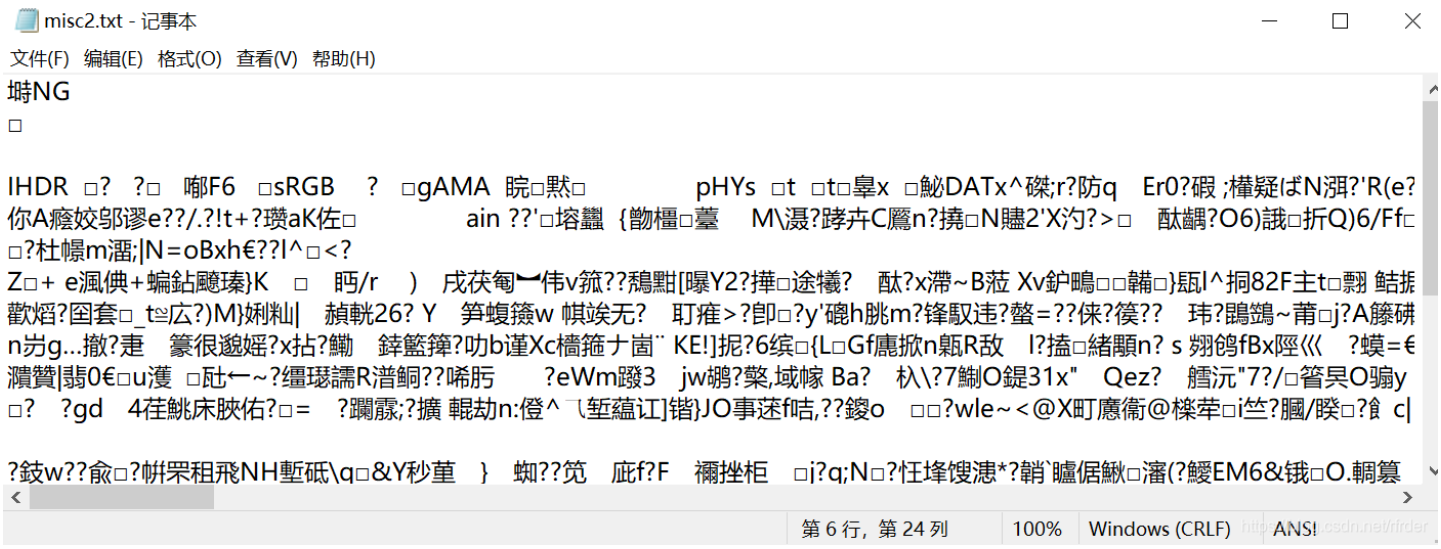
废物web狗只会web，结果就是比赛的时候web一道不会，其他方向也是一点也不会，菜死我了呜呜呜要开始学习一下MISC，跟着八神的MISC入门来慢慢学习，整篇文章可能不会有自己独立解出的题目（小白落泪），暂时只能跟着网上师傅们的WP慢慢的来学习MISC的图片篇中的各种隐写姿势，但是MISC的学习只是真的课余了，主要还是去学习web，大概会1-2天做1-2题，这样慢慢的学习，相信日积月累下去，自己的MISC也可以成长起来。

misc1

图片的内容就是flag，利用一下QQ图片的那个提取图片中的文字，就可以免于手打了。

misc2

得到的是txt文件，主要还是看文件头，可以记事本直接打开，看到这个：



这个塒NG开头就是png文件了，也可以winhex打开：

misc2.txt

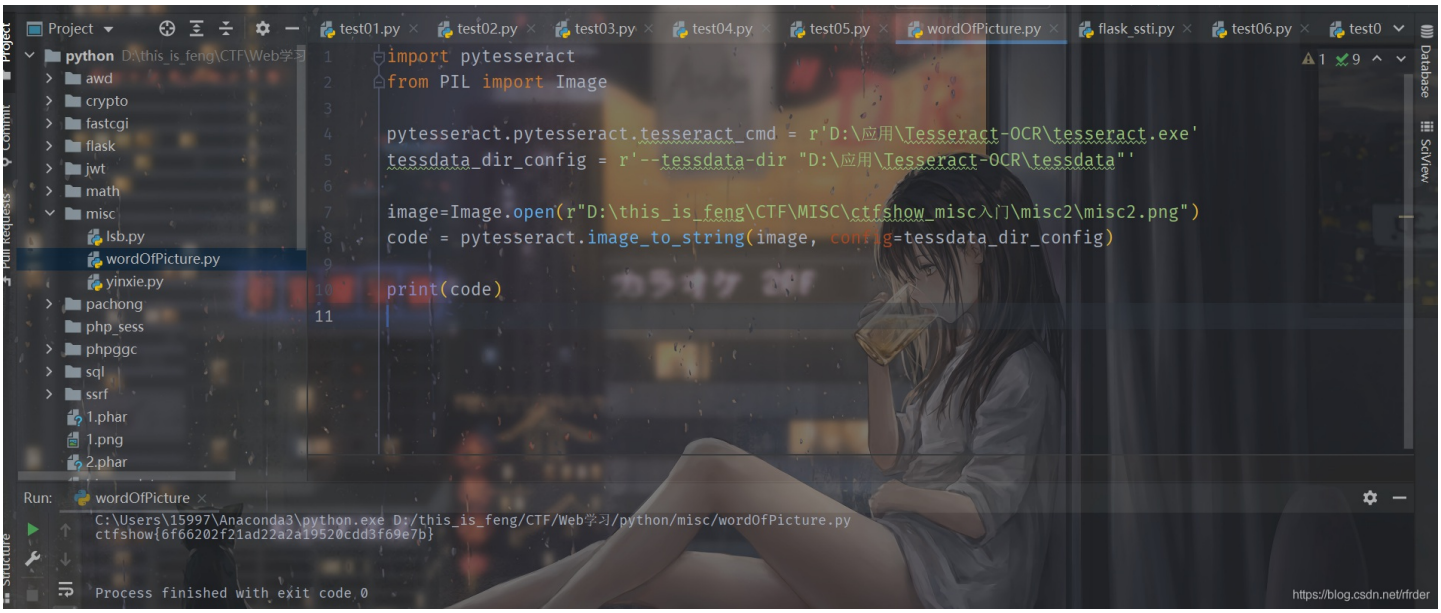
Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	‰PNG IHDR
00000010	00	00	03	84	00	00	00	96	08	06	00	00	00	86	B8	46	„ – †¸F
00000020	36	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00	6 sRGB ®Î é
00000030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	gAMA ± üa
00000040	00	09	70	48	59	73	00	00	12	74	00	00	12	74	01	DE	pHYs t t Þ
00000050	66	1F	78	00	00	1B	F5	49	44	41	54	78	5E	ED	DD	3B	f x õIDATx^íÝ;
00000060	72	DC	38	B7	C0	71	F8	AE	45	72	30	E5	15	B4	57	20	rÜ8·Àqø®Er0å ´W
00000070	3B	99	68	D2	C9	A4	D0	4E	9C	7D	A1	33	27	52	28	65	;™hÒÉ¤ÐNœ}¡3'R(e
00000080	93	3A	72	62	69	05	D2	0A	5C	13	8C	B4	97	BE	64	3F	“:rbi Ò \ Œ´—¾d?
00000090	D4	C4	93	07	24	0E	1F	C6	FF	57	C5	BA	9E	FB	A9	D9	ÔÄ“ $ ÆÿWÅºžû©Ù
000000A0	20	1E	87	38	24	9B	78	B3	6D	18	00	00	00	00	40	75	‡8$›x³m @u
000000B0	FE	EF	F0	7F	01	00	00	00	00	95	21	21	04	00	00	00	þïð •!!
000000C0	80	4A	91	10	02	00	00	00	40	A5	48	08	01	00	00	00	€J‘ @¥H
000000D0	A0	52	24	84	00	00	00	00	50	29	12	42	00	00	00	00	R$„ P) B
000000E0	A8	14	09	21	00	00	00	00	54	8A	84	10	00	00	00	00	¨ ! TŠ„
000000F0	2A	45	42	08	00	00	00	00	95	22	21	04	00	00	00	80	*EB •"! €
00000100	4A	91	10	02	00	00	00	40	A5	48	08	01	00	00	00	A0	J‘ @¥H
00000110	52	24	84	00	00	00	00	50	29	12	42	00	00	00	00	A8	R$„ P) B ¨
00000120	14	09	21	00	00	00	00	54	8A	84	10	00	00	00	00	2A	! TŠ„ *
00000130	45	42	08	00	00	00	00	95	22	21	04	00	00	00	80	4A	EB •"! €J
00000140	91	10	02	00	00	00	40	A5	48	08	01	00	00	00	A0	52	‘ @¥H R
00000150	24	84	00	00	00	00	50	29	12	42	00	00	00	00	A8	14	$„ P) B ¨
00000160	09	21	00	00	00	00	54	8A	84	10	00	00	00	00	2A	45	! TŠ„ *E
00000170	42	08	00	00	00	00	95	22	21	04	00	00	00	80	4A	91	B •"! €J‘
00000180	10	02	00	00	00	40	A5	48	08	01	00	00	00	A0	52	24	@¥H R$
00000190	84	00	00	00	00	50	29	12	42	00	00	00	00	A8	14	09	„ P) B ¨
000001A0	21	00	00	00	00	54	8A	84	10	00	00	00	00	2A	45	42	! TŠ„ *EB
000001B0	08	00	00	00	00	95	22	21	04	00	00	00	80	4A	91	10	•"! €J‘
000001C0	02	00	00	00	40	A5	48	08	01	00	00	00	A0	52	24	84	@¥H R$„
000001D0	00	00	00	00	50	29	12	42	00	00	00	00	A8	14	09	21	P) B ¨ !
000001E0	00	00	00	00	54	8A	84	10	00	00	00	00	2A	45	42	08	TŠ„ *EB
000001F0	00	00	00	00	95	22	21	04	00	00	00	80	4A	91	10	02	•"! €J‘
00000200	00	00	00	40	A5	48	08	01	00	00	00	A0	52	24	84	00	@¥H R$„
00000210	00	00	00	50	29	12	42	00	00	00	00	A8	14	09	21	00	P) B ¨ !
00000220	00	00	00	54	8A	84	10	00	00	00	00	2A	45	42	08	00	TŠ„ *EB
00000230	00	00	00	95	22	21	04	00	00	00	80	4A	91	10	02	00	•"! €J‘
00000240	00	00	40	A5	48	08	01	00	00	00	A0	52	24	84	00	00	@¥H R$„
00000250	00	00	50	A9	DF	20	21	7C	30	57	6F	DE	98	37	DD	ED	P©ß !|0WoÞ˜7Ýí
00000260	FD	8D	79	39	FC	AF	A5	BD	DC	BC	B7	BE	EB	EA	E1	F0	ý y9ü¯¥½Ü¼·¾ëêáð
00000270	3F	E0	37	34	6D	DF	C2	12	D1	07	26	F5	70	65	D5	F5	?à74mßÂ Ñ &õpeÕõ
00000280	FB	1B	6A	1A	40	A6	A9	E2	C8	A2	E3	15	E7	AE	A5	59	û j @¦©âÈ¢ã ç®¥Y

可以看到PNG的文件头，把.txt改成.png就可以打开图片了，图片内容就是flag。

从别的师傅那里看到了用python获得图片中文字的方法：

```
import pytesseract
from PIL import Image

pytesseract.pytesseract.tesseract_cmd = r'D:\应用\Tesseract-OCR\tesseract.exe'
tessdata_dir_config = r'--tessdata-dir "D:\应用\Tesseract-OCR\tessdata"'

image=Image.open(r"D:\this_is_feng\CTF\MISC\ctfshow_misc入门\misc2\misc2.png")
code = pytesseract.image_to_string(image, config=tessdata_dir_config)

print(code)
```

关于pytesseract，可以参考这个：
Python3使用 pytesseract 进行图片识别



学到了，学到了。

misc3

是bpg图片，正常不能打开，需要使用能查看bpg图片的软件打开，进行下载：
bpg

然后命令行使用即可。

```
./bpgview.exe D:\this_is_feng\CTF\MISC\ctfshow_misc入门\misc3\misc3.bpg
```

misc4

这题看了第一个图片的头是png，就把所有的图片后缀都改成png，然后把每个图片中的内容拼接出来就是flag。
但是看了别的师傅的WP，其实这样并不太对，因为真正的只有第一个图片的png，其他的几个txt都不是png图片，网上查了一下：

JPEG
文件头：FF D8 FF
文件尾：FF D9

TGA
未压缩的前4字节 00 00 02 00
RLE压缩的前5字节 00 00 10 00 00

PNG

文件头：89 50 4E 47 0D 0A 1A 0A
文件尾：AE 42 60 82

GIF

文件头：47 49 46 38 39(37) 61
文件尾：00 3B

BMP

文件头：42 4D
文件头标识(2 bytes) 42(B) 4D(M)

TIFF (tif)

文件头：49 49 2A 00

ico

文件头：00 00 01 00

Adobe Photoshop (psd)

文件头：38 42 50 53

所以第二个是jpg,第三个是bmp，第四个是gif,第五个是tif，第六个是webp文件。

misc5

做出来了我青结，虽然是最简单的misc。。。。
用winhex打开图片搜索ctfshow{，在最后找到flag。

misc6

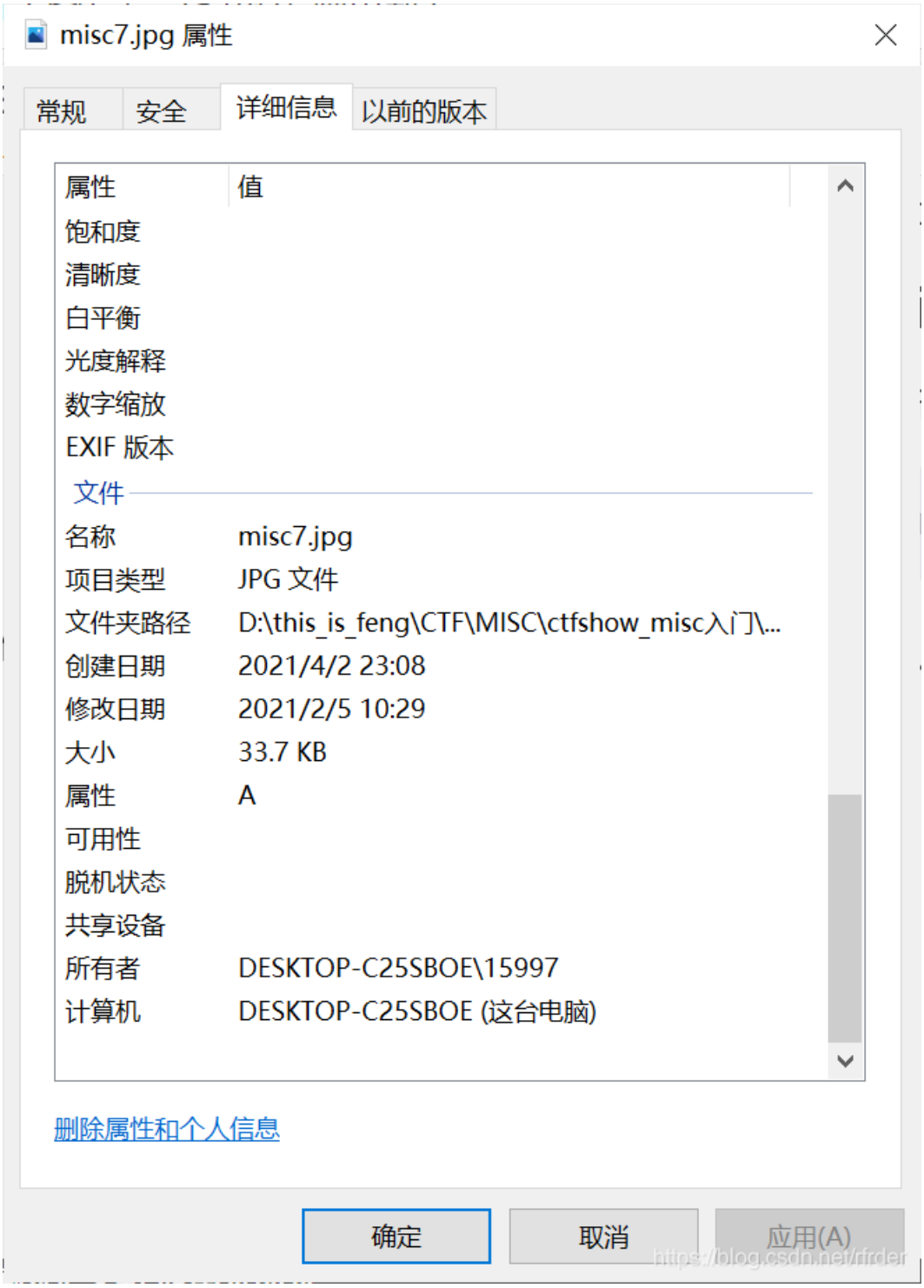
同上

misc7

提示是：

> flag在图片文件信息中。

我以为就是右键属性，藏在这里，但是发现没有：



winhex打开，搜索直接找到了flag。看了别的师傅的wp：

> 直接右键查看属性是常用的方法，不过获取不到图片的全部文件信息，也得不到这题的flag

图虫EXIF查看器

不过这玩意还是看不到flag，只是信息比较全罢了，不过先收藏着，以后或许用得到。

misc8

> flag在图片文件中图片文件中。

没太懂这提示是什么意思，看了一下WP，是图片中还隐写了其他图片：

misc8.png

位置管理器 (全部)

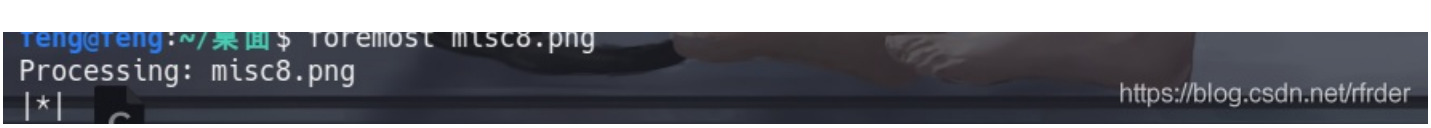
Offset ▲	搜索结果	时间
1	PNG	2021/04/02...
3893	PNG	2021/04/02...

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	ANSI ASCII
3776	29	81	10	00	00	20	52	02	21	00	00	40	A4	04	42	00	) R ! @¤ B
3792	00	80	48	09	84	00	00	00	91	12	08	01	00	00	22	25	€H „ ` "%
3808	10	02	00	00	44	4A	20	04	00	00	88	94	40	08	00	00	DJ ^"@
3824	10	29	81	10	00	00	20	52	02	21	00	00	40	A4	04	42	) R ! @¤ B
3840	00	00	80	48	09	84	00	00	00	91	12	08	01	00	00	22	€H „ ` "
3856	25	10	02	00	00	44	4A	20	04	00	00	88	52	08	FF	07	% DJ ^R ÿ
3872	33	3E	20	BA	99	89	97	04	00	00	00	00	49	45	4E	44	3> º™‰— IEND
3888	AE	42	60	82	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	®B`‚‰PNG
3904	49	48	44	52	00	00	03	84	00	00	00	96	08	02	00	00	IHDR „ –
3920	00	09	DA	D1	61	00	00	00	09	70	48	59	73	00	00	12	ÚÑa pHYs
3936	74	00	00	12	74	01	DE	66	1F	78	00	00	1D	EF	49	44	t t Þf x ïID
3952	41	54	78	9C	ED	DD	4F	88	24	57	1D	C0	F1	8A	D7	D9	ATxœíÝO^$W ÀñŠ×Ù
3968	28	28	AC	82	88	B0	26	B5	88	B8	01	45	90	24	20	E9	((¬‚^°&µ^¸ E $ é
3984	0D	6C	D8	B3	B3	DB	9B	53	60	03	2B	F4	9C	66	60	3D	lØ³³Û›S` +ôœf`=
4000	68	CF	61	B3	5E	02	33	A7	59	48	B0	03	39	C5	B5	5B	hÏa³^ 3§YH° 9Åµ[
4016	BC	08	6B	0F	E8	88	60	0F	42	10	99	C9	21	64	A5	07	¼ k è^` B ™É!d¥
4032	24	8A	F6	1C	82	9A	EE	73	79	78	E6	F9	9B	57	55	AF	$Šö ‚šîsyxæù›WU¯
4048	5E	D5	7B	5D	AF	77	F3	FD	9C	6A	77	AA	5F	BF	7A	FF	^Õ{]¯wóýœjwª_¿zÿ
4064	EA	D7	F5	E7	BD	27	B2	2C	4B	00	00	00	80	18	3E	13	ê×õç½'²,K € >
4080	3B	03	00	00	00	F8	F4	22	18	05	00	00	40	34	04	A3	; øô" @4 £
4096	00	00	00	88	86	60	14	00	00	00	D1	10	8C	02	00	00	^†` Ñ Œ

可以使用binwalk或者foremost，参考文章：

MISC中图片隐藏文件分离

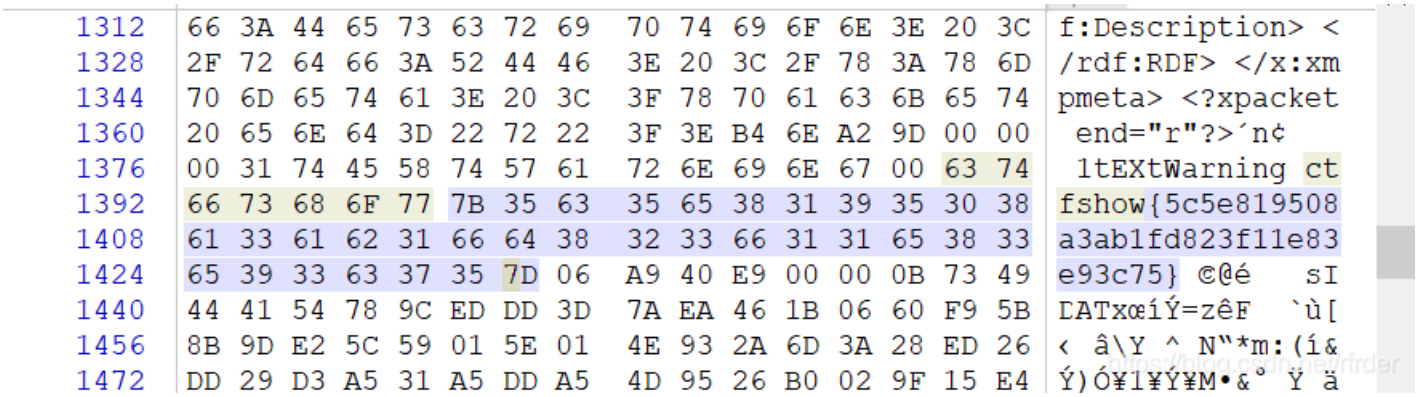
```
feng@feng:~/桌面$ binwalk misc8.png

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
```



再去output目录里面去看分离出来的所有文件，有一张打开就可以看到flag，再转文字即可。
学到了学到了，binwalk和foremost这两个还要再学习学习，熟练一下。

misc9

提示在图片块里，师傅们说是也就是数据块。但是其实这题直接winhex打开搜索也可以直接发现flag。



用010Editor看一下的话，这个 chunk估计就是所谓的数据块了。

Template Results - PNG.bt

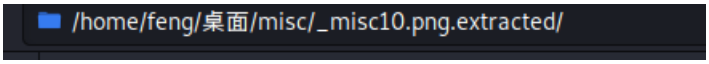
Name	Value	Start	Size	Color	Comment
struct PNG_SIGNATURE sig		0h	8h	Fg: Bg	
struct PNG_CHUNK chunk[0]	IHDR (Critical, Public, Unsafe to Copy)	8h	19h	Fg: Bg	
struct PNG_CHUNK chunk[1]	pHYs (Ancillary, Public, Safe to Copy)	21h	15h	Fg: Bg	
struct PNG_CHUNK chunk[2]	iTXt (Ancillary, Public, Safe to Copy)	36h	528h	Fg: Bg	
struct PNG_CHUNK chunk[3]	tEXt (Ancillary, Public, Safe to Copy)	55Eh	3Dh	Fg: Bg	
uint32 length	49	55Eh	4h	Fg: Bg	
union CTYPE type	tEXt	562h	4h	Fg: Bg	
struct PNG_CHUNK_TEXT text	Warning = ctfshow{5c5e819508a3ab1fd823f11e83e93c75}	566h	31h	Fg: Bg	
uint32 crc	6A940E9h	597h	4h	Fg: Bg	
struct PNG_CHUNK chunk[4]	IDAT (Critical, Public, Unsafe to Copy)	59Bh	B7Fh	Fg: Bg	
struct PNG_CHUNK chunk[5]	IEND (Critical, Public, Unsafe to Copy)	111Ah	Ch	Fg: Bg	

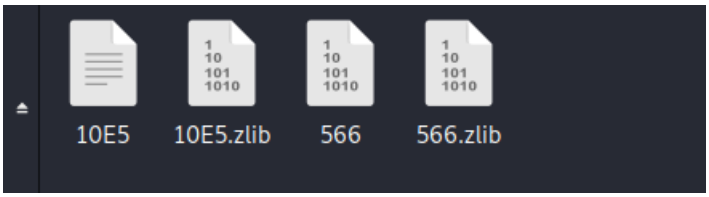
misc10

知识盲区，提示flag在图片数据里。图片数据又是个什么鬼。。。。
查了一下。

> PNG定义了两种类型的数据块：一种是PNG文件必须包含、读写软件也都必须要支持的关键块（critical chunk）；另一种叫做辅助块（ancillary chunks），PNG允许软件忽略它不认识的附加块。这种基于数据块的设计，允许PNG格式在扩展时仍能保持与旧版本兼容。
>
> 关键数据块中有4个标准数据块：
>
> - 文件头数据块IHDR（header chunk）：包含有图像基本信息，作为第一个数据块出现并只出现一次。
> - 调色板数据块PLTE（palette chunk）：必须放在图像数据块之前。
> - 图像数据块IDAT（image data chunk）：存储实际图像数据。PNG数据允许包含多个连续的图像数据块。
> - 图像结束数据IEND（image trailer chunk）：放在文件尾部，表示PNG数据流结束。

用binwalk看一下，发现有2个Zlib compressed data。
Binwalk工具的详细使用说明
使用`binwalk -e`来提取一下文件。





打开第一个文件就可以得到flag。
八神关于原理的解释：

LV53 8神 cheyenne

zlib是PNG IDAT块数据可选的压缩格式

LV53 8神 cheyenne

那题实际上我自己生成了一段zlib压缩数据，然后加上长度位、标识位和CRC校验位

LV53 8神 cheyenne

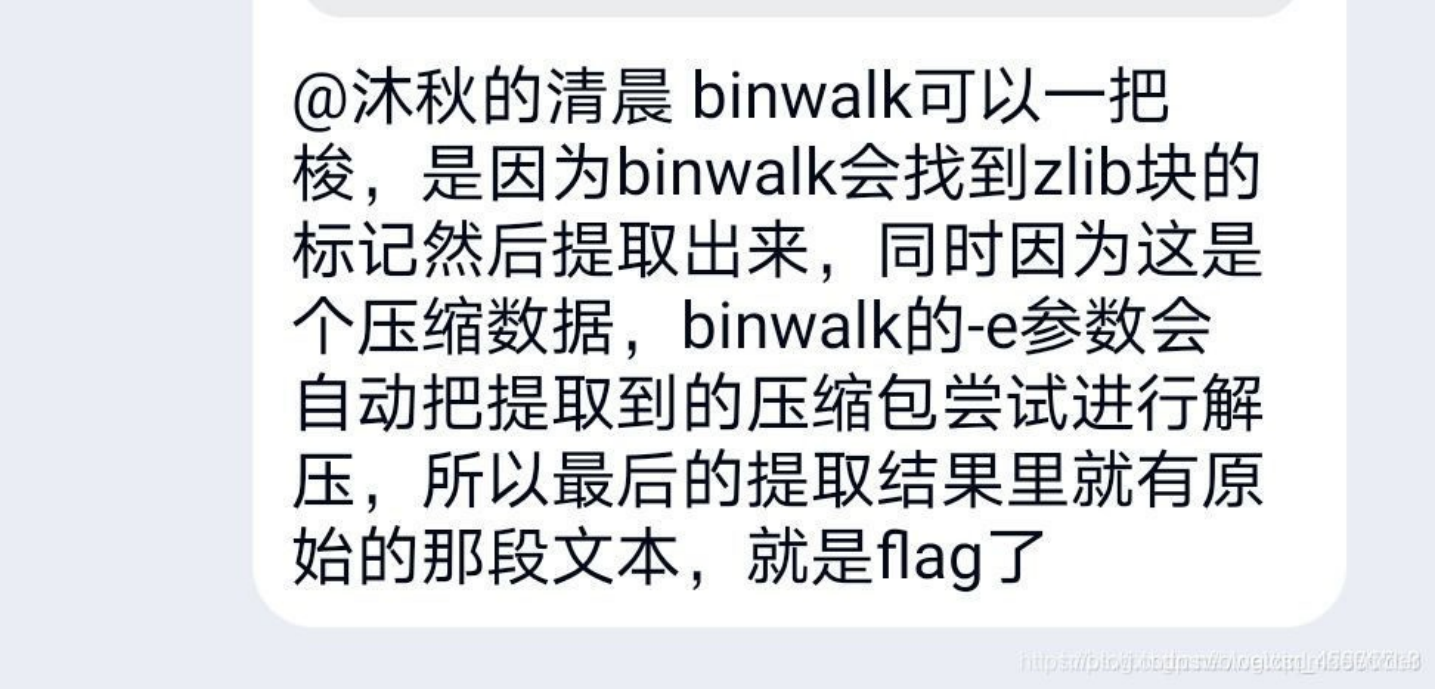
伪装成了一个IDAT块，然后插入了图片文件里

LV53 8神 cheyenne

沐秋的清晨 下午6:24

阿这

来自其他聊天



学到了，学到了。

misc11

又学到新东西了，提示是：

> flag在另一张图里。

拿010 Editor打开看一下，发现和上一题一样有2个IDAT块，我想既然是另一张图，以为还是之前那样，把IDAT块取出来然后解压就可以得到另一张图片了，但是发现没有。。。
看了一下别的师傅的WP，这题原来是要删除第一个IDAT块，这样得到的新的图片就可以得到flag了。

删除IDAT块比较方便的是使用tweakpng这个工具：
Tweakpng

只需要右击第一个IDAT块，然后delete就行了。

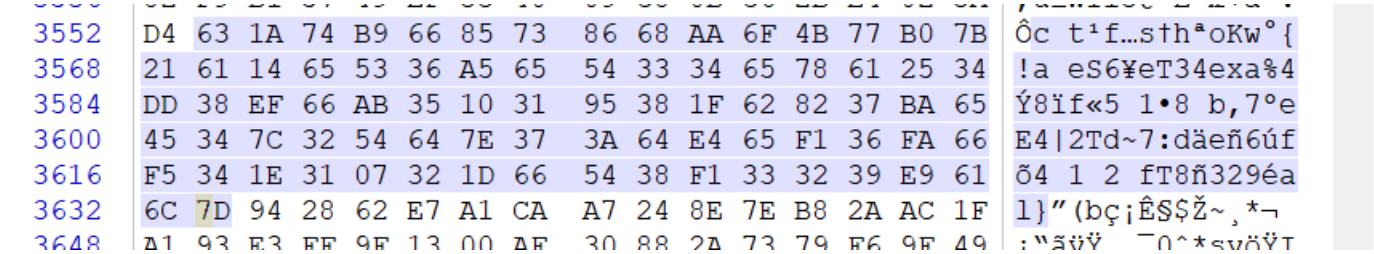
misc12

做法同上，删掉前8个IDAT数据块就可以了。

misc13

> flag位置在图片末尾。

我一开始以为是图片的IEND块有问题，但是改了发现还是没flag。。。。
看了一下WP，原来是图片末尾有这么一块：



3552	D4	63	1A	74	B9	66	85	73	86	68	AA	6F	4B	77	B0	7B	Ôc t¹f…sth ªoKw°{
3568	21	61	14	65	53	36	A5	65	54	33	34	65	78	61	25	34	!a eS6¥eT34exa%4
3584	DD	38	EF	66	AB	35	10	31	95	38	1F	62	82	37	BA	65	Ý8ïf«5 1•8 b‚7ºe
3600	45	34	7C	32	54	64	7E	37	3A	64	E4	65	F1	36	FA	66	E4|2Td~7:däeñ6úf
3616	F5	34	1E	31	07	32	1D	66	54	38	F1	33	32	39	E9	61	õ4 1 2 fT8ñ329éa
3632	6C	7D	94	28	62	E7	A1	CA	A7	24	8E	7E	B8	2A	AC	1F	l}”(bç¡Ê§$Ž~¸*¬
3648	A1	93	E3	FF	9F	13	00	AF	30	88	2A	73	79	F6	9F	49	¡“ãÿŸ ¯0ˆ*syöŸI

可以看到ctfshow，隔一个字符取一个，写一个脚本跑出来：

```
a="631A74B96685738668AA6F4B77B07B216114655336A56554333465786125 34DD38EF66AB35103195381F628237BA6545347C3254647E373A64E465F136FA66F5341E3107321D665438F1333239E9616C7D"

flag=""
for i in range(0,len(a),4):
    hexStr=a[i:i+2]
    flag+=chr(int("0x"+hexStr,16))
print(flag)
```