

【CTFshow】misc入门总结

原创

[Sunlight_316](#) 于 2022-01-29 10:24:31 发布 371 收藏 1

分类专栏: [CTFshow](#) 文章标签: [python](#) [web安全](#) [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/weixin_51614272/article/details/122737372

版权



[CTFshow](#) 专栏收录该内容

6 篇文章 0 订阅

订阅专栏

图片篇(基础操作)

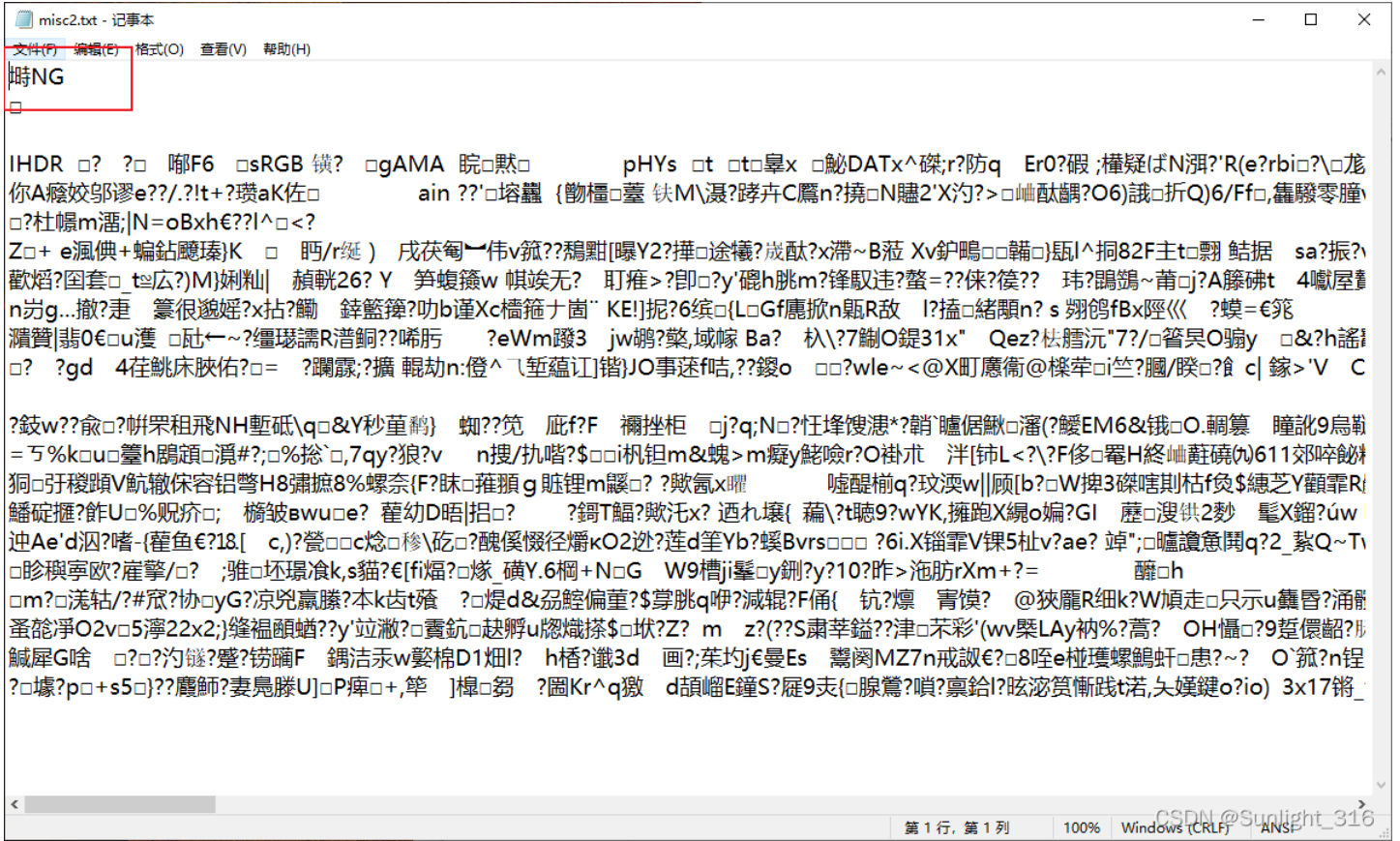
misc1

直接看图

```
ctfshow{22f1fb91fc4169f1c9411ce632a0ed8d}
```

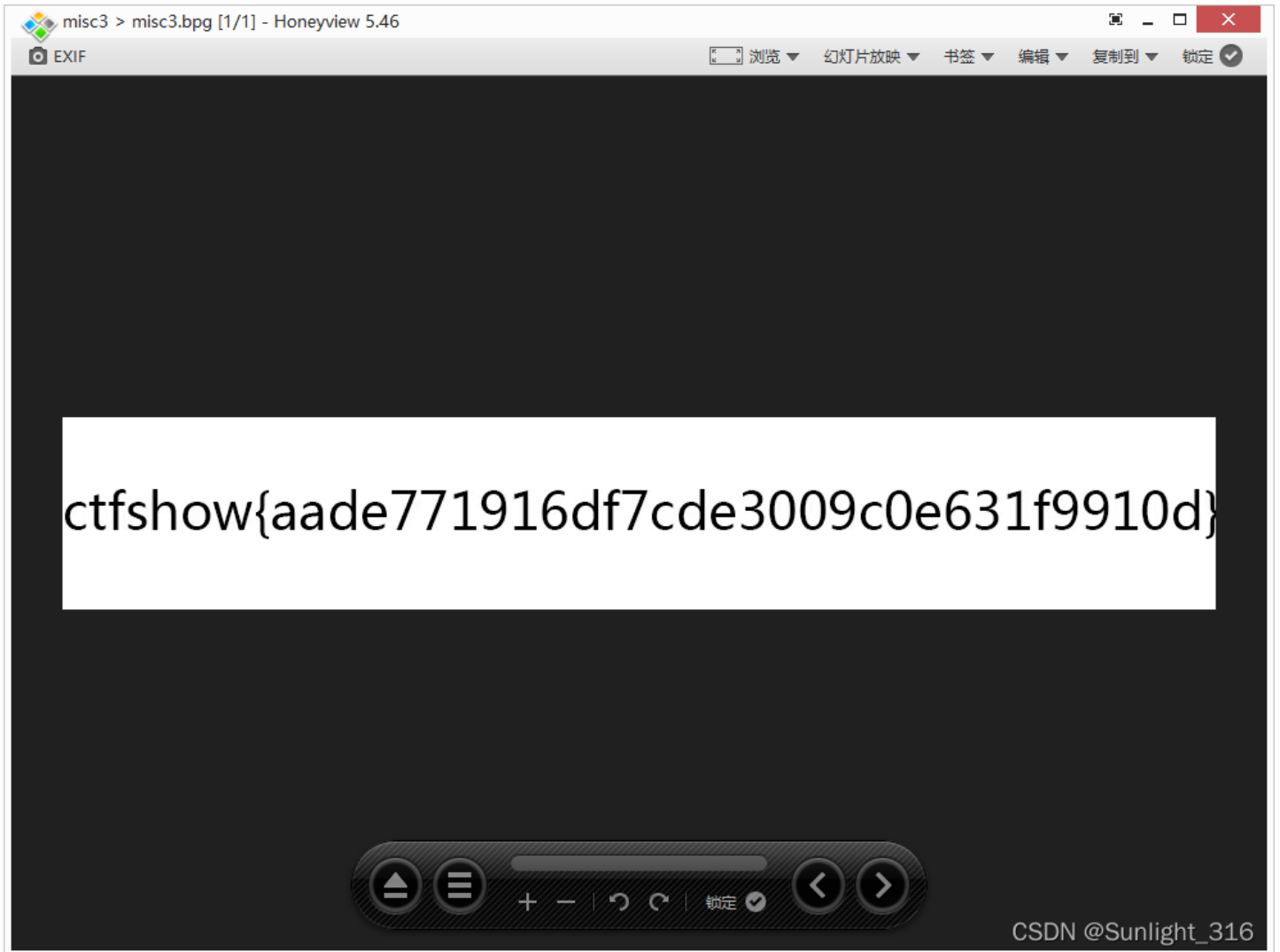
misc2

查看文件头, 发现是png文件, 将后缀改为png直接打开就是flag



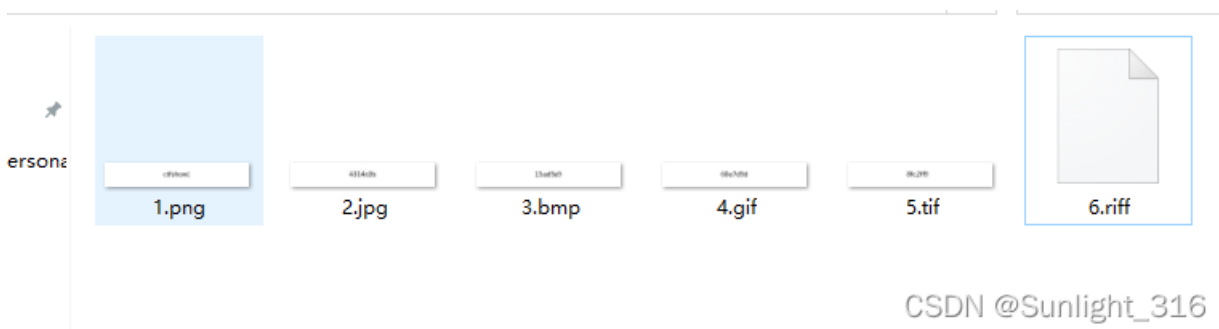
misc3

下载Honeyview直接打开bmp文件



misc4

查看文件头，发现依次是.png、.jpg、.bmp、.gif、.tif、.riff，flag文件，刚好是每张图组合起来

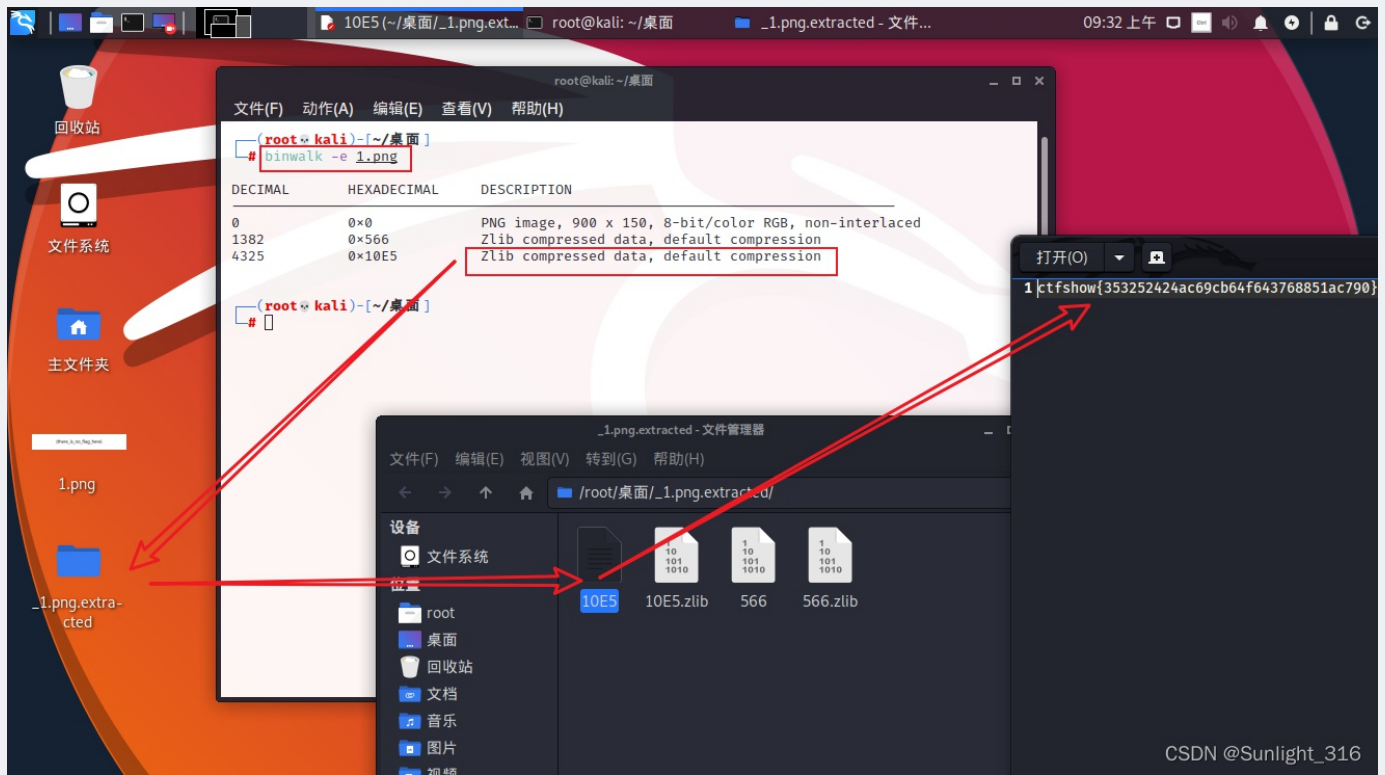


图片篇(信息附加)

misc5-7

直接用打开010打开搜索flag相关信息得到flag

binwalk分析图片结构——>分离出含有隐藏的zip信息



misc11

提示：flag在另一张图里。

这个图有两个IDAT块，而且没有隐写其他的数据，试着把第一个IDAT块的数据删除，然后另存为一张新图片，这个过程可以手动操作，也可以使用tweakpng工具。

misc12

提示：flag在另一张图里。

和上题一样的提示，所以思路是一样的。不过这题有30个IDAT块，用PNGDebugger跑了一下，发现没有出错的IDAT块...

需要删掉前8个IDAT块，用tweakpng工具更方便

misc13

一个关键的提示：这个图片的IDEN块不是0，是有数据的，查看数据得知：

```

0D70h: C8 25 FD 0E 88 1D 51 01 6B 37 8B E0 B8 FB 26 5E E%y.,.Q,k7(à.0&^
0D80h: F6 43 9F 39 23 D4 03 B4 02 D8 29 2B 2A 4E 2D C8 0CY9#0.'0)+*N-E
0D90h: 93 50 B1 8D B6 1C 40 15 43 1E 51 D1 D9 14 3C BF "Ps.#.e.c.QNU.<_
0DA0h: 9F 20 86 87 C9 25 7C CA 03 97 36 80 F3 37 9D 32 Y i±E%|É.-6e07.2
0DB0h: 28 EC 16 7C B7 E7 6D C7 02 DB 7C 67 DB C6 D0 C5 (l.|çmç.0|güEBA
0DC0h: D5 45 CE F5 E0 DC 88 3D 64 5C 17 25 3B 7C 96 09 0EiSâU~d\,%|~
0DD0h: 82 E9 B4 57 40 FF 02 40 00 C0 00 C6 2B 64 02 ~.ú:ñiIi3@.E.#:â.:
0DE0h: D4 63 1A 74 B9 66 85 73 86 68 AA 6F 4B 77 B0 1d 0. t'f..sth0kw*(
0DF0h: 21 61 14 65 53 36 A5 65 54 33 34 65 78 61 25 34 !a.ea0reI34exa3
0E00h: DD 38 E5 66 AB 35 10 31 95 38 1F 62 82 37 BA 65 Y8ife5.1~.8..b.7°e
0E10h: 45 34 7E 32 54 64 7E 37 3A 64 E4 65 F1 36 FA 6E E4|2Td-7:daeñ6úf
0E20h: F5 34 7E 31 07 32 1D 66 54 38 F1 33 32 39 E9 61 04.1.2.fT8ñ329éa
0E30h: 6C 7D 9A 28 62 E7 A1 CA A7 24 8E 7E B8 2A AC 1F 1)~(bc|ÉSSZ~.*~.
0E40h: A1 93 E3 F 9F 13 00 AF 30 88 2A 73 79 F6 9F 49 i~âyY..0~*sy0YI
0E50h: 20 D1 85 34 93 13 F7 35 D1 85 25 55 17 06 9E EA N.,.,=5N%u..zè
0E60h: B9 59 9C 15 3F 79 B2 A6 4D C3 17 AA 7C 12 31 *Yacç.zy+|MA.*|~|
0E70h: 25 03 FE FA AB C8 63 7C BE CE 1C DB 4E D4 7D 35 %|b|w|E|ñ|ÛN0)5
0E80h: D6 43 BD B3 FF 7C 5C 1A 78 1B 7F 02 6C 79 53 32 0C%y|.|.x...ly52
0E90h: 7A 7C C4 3E 17 2E 74 B2 47 17 54 C1 A6 E5 6F ED z|A~-.t?G.TA|àoI
0EA0h: 38 C5 C8 0F 19 89 93 39 04 D5 A7 DF 27 14 58 9C 8AE.1%*9.0S8'.Xe
0EB0h: 96 4C 1F 5B D1 9C 92 92 39 AB A4 3B D3 CA 31 09 -.L.[8e''9=0:0É1.
0EC0h: C0 59 EA F3 01 5A 23 DC DC 34 C8 DE 3A 9C 35 A0 ÀYé0.Z#U04É:æ5
0ED0h: A7 AB D5 56 45 8C 5D 3F 54 50 D2 40 DD B6 14 7D $«0VEK|?TP0eY(|.}
0EE0h: FC DC FE 33 D2 32 35 C0 72 BB 97 92 BE 5C 89 23 uÜp30r5Ar~"%"#
0EF0h: 88 B8 53 8D 17 7B F9 63 1A 74 B9 66 85 73 86 68 ^ S..0üc.t'f..sth
0F00h: AA 6F 4B 77 B0 7B 21 61 14 65 53 36 A5 65 54 34 *okw*{|a.eS6veT4
0F10h: 34 36 78 63 25 3A DD 38 EF 66 AB 37 10 33 95 39 46xc%4Y81f:7.3~9
0F20h: 1F 62 82 37 BA 65 45 62 7C 32 54 64 7E 31 3A 64 .b.7°eEb|2Td-1:d
0F30h: E4 65 F1 36 FA 65 15 34 1E 31 07 32 1D 66 54 38 aen6ue04.1.2.fT8
0F40h: F1 33 32 39 E9 61 0C 7D 2B F5 E0 D5 3E 44 E6 D0 ñ329éa1)+0ä0~DæI
0F50h: C8 C8 F3 A5 2F 79 3 96 FE 41 76 F9 6E 49 E4 BA EÉ0V/y3-þAvùnIa°
0F60h: BD 00 D8 92 68 B2 89 27 62 57 3E 21 AF BB 6C 65 %..0'h'%'bw|'!>le
0F70h: A3 0E 80 43 5D 0A 69 24 E7 E4 5A 22 9B ED AF 59 f.cC).i$çqz"~i Y
0F80h: 05 06 CE C7 BE 74 EB 1C 6F 9F 06 1E C9 81 5F 16 ..iCtete0V..E...
0F90h: F6 3F BF 7C 4F DE 00 3A 07 65 92 89 3B 5A 5A 3B 07z|0b~*.e'k%ZZ:
0FA0h: 23 C3 06 1D 2D 74 79 C8 B9 68 8E 8A 87 3E A5 93 #Á..Ûye'kZ$~+~"
0FB0h: 1A F1 32 CF 8F 85 A6 78 E6 62 EC FE 9D CE 44 AC .ñ2I...xabi1.ID~
0FC0h: 8C BF D5 98 1A 8D 7C 10 26 A4 91 0B 35 8D 6F 21 Gz0'..|.&~'.5.o!
0FD0h: 7B A4 C6 A4 D0 7D 1B 79 44 0C 4D BE B2 9F 1C 86 {#E#0}.yA.Mk*Y.t
0FE0h: E2 43 39 1C C4 C8 3B F7 D6 73 8F F3 4D 7A 86 91 äC9..ÄE:þs.0Mz1'
0FF0h: 1D 37 F3 36 7F 16 CC 44 CE 0C 88 3E FA 5E E1 7E .706..IDE.'>u'á~
1000h: 07 56 CD 4B A4 15 E0 B7 EE 3B D1 94 C0 E9 A0 06 .VIK0.ä.i:N'Aé .
1010h: 0E 3B 9E 0B 23 CE 12 B9 99 17 39 62 A5 41 B5 37 .:z.#I.'~9bVAp7
1020h: 7A 00 00 00 02 49 45 4E 44 0D E1 67 7D 8B 8F z....IEND.ä;}<.

```

CSDN @Sunlight_316

那ODE1的这个位置的flag不就是正确的flag了吗
隔一个16进制数取一个

脚本:

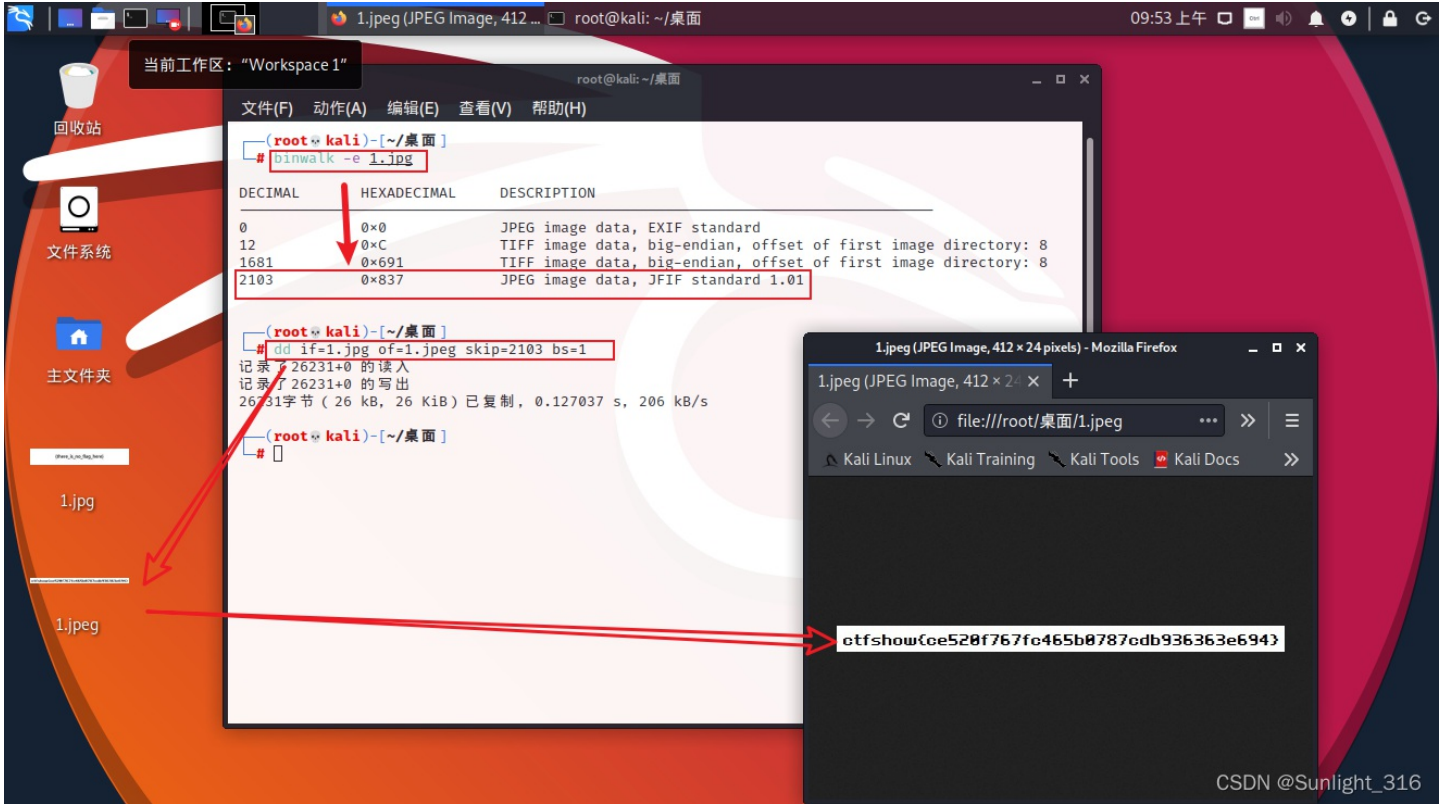
```

a="631A74B96685738668AA6F4B77B07B216114655336A5655433346578612534DD38EF66AB35103195381F628237BA6545347C3254647E3
73A64E465F136FA66F5341E3107321D665438F1333239E9616C7D"
r = ''
s=bytes.fromhex(a)
for i in range(0,len(s),2):
    r+=chr(s[i])
print(r)
#fromhex函数把一串16进制字符，每两位转换成16进制的对象

```

misc14

binwalk分析图片结构，-e解不出来，直接用dd命令分离出文件:

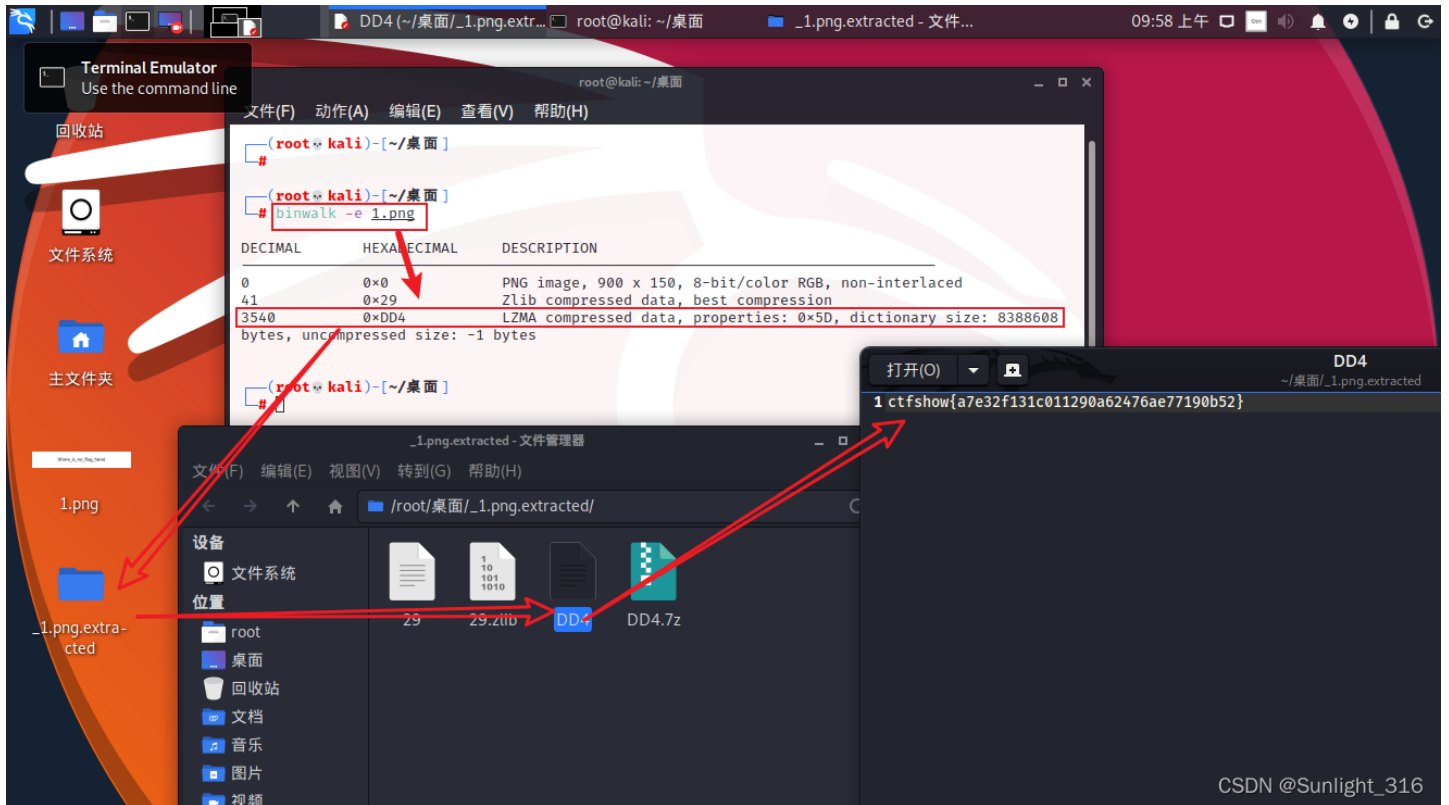


misc15

直接用打开010打开搜索flag相关信息得到flag

misc16

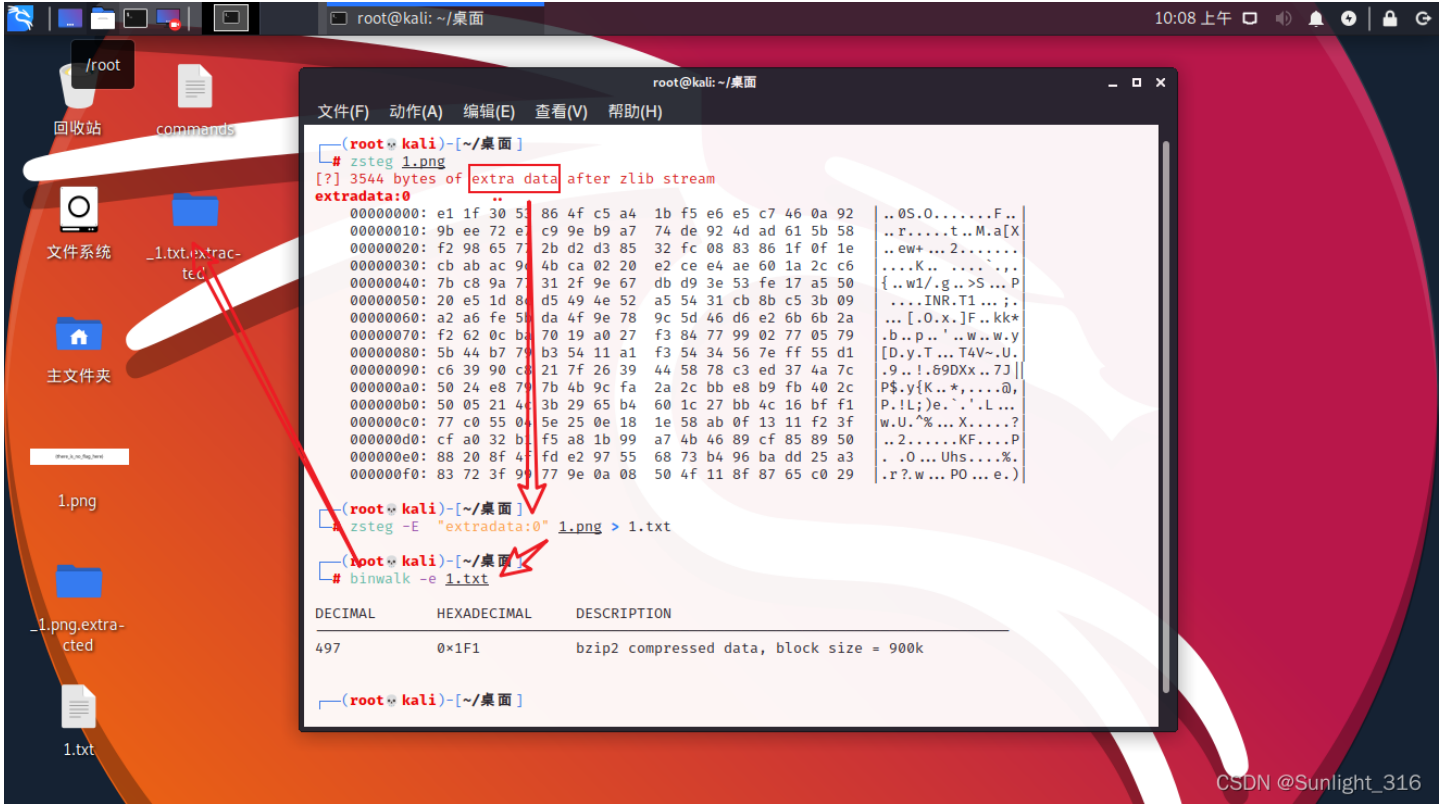
binwalk分析图片结构——>分离出含有隐藏的LZMA信息，直接可以打开就是flag



CSDN @Sunlight_316

misc17

binwalk分析得出有一个bzip，但是什么都分离不出来
用zsteg试试，发现有额外隐藏的数据
steg -E "extradata:0" misc17.png > 1.txt 把隐藏的数据分离到1.txt
用binwalk分离1.txt得到flag



CSDN @Sunlight_316

misc18-23

用 exiftool 查看图片的具体信息，有的点开属性也可以

misc21: 提示：flag在序号里。

序号ASCII码转成字符：hex(X&Ys)

所以flag是信息中X和Y一段一段转成16进制再组合起来：

misc22: 缩略图隐写

exiftool -ThumbnailImage -b misc22.jpg > 1.jpg //缩略图隐写

misc23: 时间中的+08:00指的是时间要加8个小时，改成东八区时间

misc41

F001是突破点，这个位置有大量F001，看起来组成了某种形状，O10打开图片，搜索F001，全部高亮！形状就是flag

图片篇(文件结构)

misc24-36

用tweakpng打开发现CRC校验错误，说明要改宽高了
 修改png, jpg, gif, bmp四种图片文件的宽高
 png: 前四位是宽，后四位是高
 jpg: 前两位是高，后两位是宽
 bmp: 前四位是宽，后四位是高，但是是倒着写
 03 B6——>写成B6 03

先打开属性，确定目前宽高数据，再转换成十六进制在010中找，就很容易找到

- 用脚本直接爆破出png, jpg正确的宽高
- 原理就是根据CRC32算出宽高来

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	B9	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	PNP IHDR
00000010	00	00	01	F4	00	00	01	A4	08	06	00	00	00	CB	D6	DF	ó × EÖB
00000020	8A	00	00	00	09	70	48	59	73	00	00	12	74	00	00	12	pHYs t
00000030	74	01	DE	66	1F	78	00	00	0A	4D	69	43	43	50	50	68	t þf x MiCCPPh
00000040	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66	otoshop ICC prof
00000050	69	6C	65	00	00	78	DA	9D	53	77	58	93	F7	16	3E	DF	ile xÚ SwX ÷ >B

0000h:	FF	D8	FF	EE	00	0E	41	64	6F	62	65	00	64	40	00	00	y@yi..Adobe.de..
0010h:	00	01	FF	DB	00	84	00	02	02	02	02	02	02	02	02	02	..yU.....
0020h:	02	03	02	02	02	03	04	03	02	02	03	04	05	04	04	04
0030h:	04	04	05	06	05	05	05	05	05	05	06	06	07	07	08	07
0040h:	07	06	09	09	0A	0A	09	09	0C	0C	0C	0C	0C	0C	0C	0C
0050h:	0C	0C	0C	0C	0C	0C	0C	01	03	03	03	05	04	05	09	06
0060h:	06	09	0D	0A	09	0A	0D	0F	01	0E	0E	0F	0F	0F	0C	0C
0070h:	0C	0C	0C	0F	0F	0C	0C	0C	0C	0C	0C	0F	0C	0C	0C	0C
0080h:	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C
0090h:	0C	0C	0C	0C	0C	0C	0C	0C	FF	C0	00	11	06	00	96	03yA.....
00A0h:	84	03	01	11	00	02	11	01	03	11	01	FF	DD	00	04	00yY...
00B0h:	71	5F	C4	01	A2	00	00	00	07	01	01	01	01	01	00	00	qyA.C.....
00C0h:	00	00	00	00	00	04	05	03	02	06	01	00	00	00	08	09
00D0h:	0A	0B	01	00	02	02	03	01	01	01	01	01	00	00	00	00
00E0h:	00	00	00	00	00	02	03	04	05	06	07	08	09	0A	0B	10
00F0h:	00	02	01	03	00	02	04	02	06	07	03	04	02	06	02	73s
0100h:	01	02	03	11	04	00	05	21	12	31	00	51	06	13	61	22!..1AQ..a"
0110h:	71	81	14	32	91	A1	15	B1	40	23	C1	52	D1	E1	33		q..2'j..±B#ARÑá3
0120h:	16	62	F0	24	72	82	F1	25	34	53	92	A2	B2	63	73		.bð\$r,ñ%C4S'c²cs
0130h:	C2	35	44	27	93	A3	B3	36	54	64	74	C3	D2	E2	08		ÅSD'£³6.TdtÅ0â.
0140h:	26	83	09	0A	18	19	84	94	45	46	A4	B4	56	D3	55	28	&f....."EFª VÓU(
0150h:	1A	F2	E3	F3	C4	D4	E4	F4	65	75	85	95	A5	B5	C5	D5	..ððA0ãøeu...YµA0
0160h:	E5	F5	66	76	86	96	A6	B6	C6	D6	E6	F6	37	47	57	67	ãøfvñ-!¶Æ0æø7Gwg
0170h:	77	87	97	A7	B7	C7	D7	E7	F7	38	48	58	68	78	88	98	wñ-š·Ç×ç=8HXhx~
0180h:	A8	B8	C8	D8	E8	F8	29	39	49	59	69	79	89	99	A9	B9	· Èøèø)9IYiy&™ø†
0190h:	C9	D9	E9	F9	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	ÈUèù*:JZjzŠšªøÈU
01A0h:	EA	FA	11	00	02	02	01	02	03	05	05	04	05	06	04	08	èù.....
01B0h:	03	03	6D	01	00	02	11	03	04	21	12	31	41	05	51	13	..m.....!..1A.Q.
01C0h:	61	22	06	71	81	91	32	A1	B1	F0	14	C1	D1	E1	23	42	a".q."2j±ð.AÑª#B
01D0h:	15	52	62	72	F1	33	24	34	43	82	16	92	53	25	A2	63	..Rbrñ3\$4C,..'S'cc
01E0h:	B2	C2	07	73	D2	35	E2	44	83	17	54	93	08	09	0A	18	*Å.s05ãDf.T".....
01F0h:	19	26	36	45	1A	27	64	74	55	37	F2	A3	B3	C3	28	29	..&6E.'dtU7ðE³Å(
0200h:	D3	E3	F3	84	94	A4	B4	C4	D4	E4	F4	65	75	85	95	A5	ðãø,"ª A0ãøeu...Y
0210h:	B5	C5	D5	E5	F5	46	56	66	76	86	96	A6	B6	C6	D6	E6	µA0ãøFVfvñ-!¶Æ0æ
0220h:	F6	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7	38	48	58	øGwgwñ-š·Ç×ç=8HX
0230h:	68	78	88	98	A8	B8	C8	D8	E8	F8	39	49	59	69	79	89	· Èøèø)9IYiy&™ø†
0240h:	99	A9	B9	C9	D9	E9	F9	2A	3A	4A	5A	6A	7A	8A	9A	AA	ÈUèù*:JZjzŠšªøÈU

chall	misc30.bmp															ANSI	ASCII	
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
00000000	42	4D	50	87	06	00	00	00	00	00	36	00	00	00	28	00	BMP+	6 (
00000016	00	00	B6	03	00	00	96	00	00	00	01	00	18	00	00	00	█	-
00000032	00	00	1A	87	06	00	12	0B	00	00	12	0B	00	00	00	00	+	
00000048	00	00	00	00	00	00	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Y	Y
00000064	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Y	Y
00000080	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Y	Y
00000096	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Y	Y
00000112	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	Y	Y

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

misc37

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

misc38

用APNG Disassembler来把每一帧分离出来，9、17、36、40帧中藏有flag

misc39

不同帧之间的间隔时间来隐写
 提取命令: identify -format "%T " misc39.gif > 1.txt, 得到的一串36和37
 36换成0,37换成1, 得到一串01
 由于无法整除8, 就7个一组

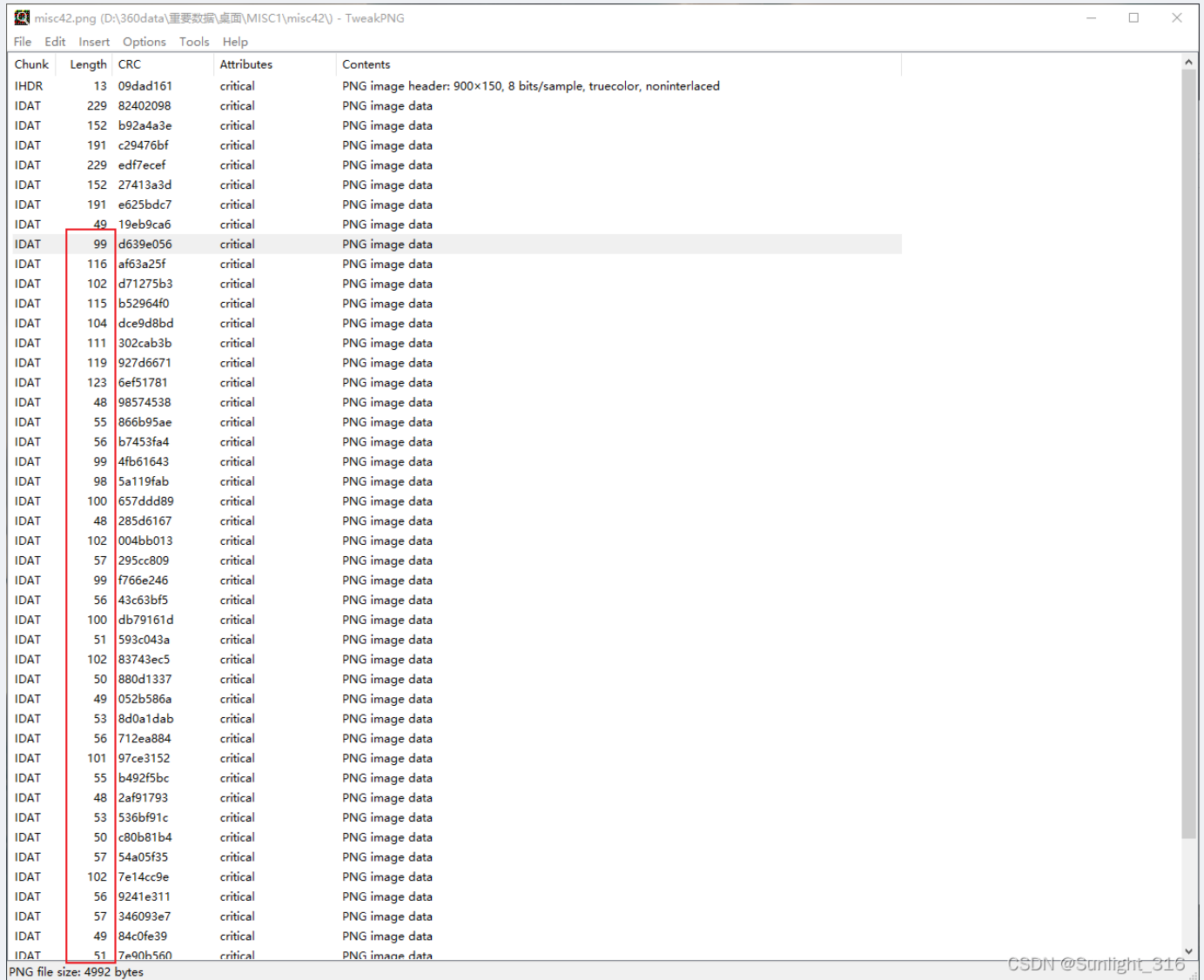
misc40

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

```
flag=""
for i in range(28,69): #flag内容从28位开始
    f = open('apngframe'+str(i)+'.txt')
    s = f.read()
    flag += chr(int(s.split("/")[0][6:]))
print(flag)
```

misc42

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ACILL转换为字符是ctfshow，将后面的接着转换成字符即可得到flag



Chunk	Length	CRC	Attributes	Contents
IHDR	13	09dad161	critical	PNG image header: 900x150, 8 bits/sample, truecolor, noninterlaced
IDAT	229	82402098	critical	PNG image data
IDAT	152	b92a4a3e	critical	PNG image data
IDAT	191	c29476bf	critical	PNG image data
IDAT	229	edf7ecef	critical	PNG image data
IDAT	152	27413a3d	critical	PNG image data
IDAT	191	e625bdc7	critical	PNG image data
IDAT	49	19eb9ca6	critical	PNG image data
IDAT	99	d639e056	critical	PNG image data
IDAT	116	af63a25f	critical	PNG image data
IDAT	102	d71275b3	critical	PNG image data
IDAT	115	b52964f0	critical	PNG image data
IDAT	104	dce9d8bd	critical	PNG image data
IDAT	111	302cab3b	critical	PNG image data
IDAT	119	927d6671	critical	PNG image data
IDAT	123	6ef51781	critical	PNG image data
IDAT	48	98574538	critical	PNG image data
IDAT	55	866b95ae	critical	PNG image data
IDAT	56	b7453fa4	critical	PNG image data
IDAT	99	4fb61643	critical	PNG image data
IDAT	98	5a119fab	critical	PNG image data
IDAT	100	657dd89	critical	PNG image data
IDAT	48	285d6167	critical	PNG image data
IDAT	102	004bb013	critical	PNG image data
IDAT	57	295cc809	critical	PNG image data
IDAT	99	f766e246	critical	PNG image data
IDAT	56	43c63bf5	critical	PNG image data
IDAT	100	db79161d	critical	PNG image data
IDAT	51	593c043a	critical	PNG image data
IDAT	102	83743ec5	critical	PNG image data
IDAT	50	880d1337	critical	PNG image data
IDAT	49	052b586a	critical	PNG image data
IDAT	53	8d0a1dab	critical	PNG image data
IDAT	56	712ea884	critical	PNG image data
IDAT	101	97ce3152	critical	PNG image data
IDAT	55	b492f5bc	critical	PNG image data
IDAT	48	2af91793	critical	PNG image data
IDAT	53	536bf91c	critical	PNG image data
IDAT	50	c80b81b4	critical	PNG image data
IDAT	57	54a05f35	critical	PNG image data
IDAT	102	7e14cc9e	critical	PNG image data
IDAT	56	9241e311	critical	PNG image data
IDAT	57	346093e7	critical	PNG image data
IDAT	49	84c0fe39	critical	PNG image data
IDAT	51	7a90h560	critical	PNG image data

PNG file size: 4992 bytes

CSDN@Sunlight_316

misc43

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

misc44

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

misc45

转成.bmp格式后，用binwalk提取即可，考察点是png和bmp像素点的读取方式（？）

misc46

提取出GIF的详细信息
identify misc46.gif > message.txt

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()
```

misc47

misc48

用010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

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

再分别转换成hex

```
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

把FFE后面的那个字符提取出来，再连接在一起，一共32位()，这就是flag

图片篇(颜色通道)

misc50

直接Stegsolve查看颜色通道