




ctfshow 8神PNG隐写入门(土)赛 WP

原创

是Mumuzi  于 2021-03-16 20:59:35 发布  3270  收藏 6

分类专栏: [ctf ctfshow crypto](#) 文章标签: [网络安全](#)

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/qq_42880719/article/details/114825260

版权



[ctf](#) 同时被 3 个专栏收录

75 篇文章 28 订阅

订阅专栏



[ctfshow](#)

23 篇文章 8 订阅

订阅专栏



[crypto](#)

2 篇文章 1 订阅

订阅专栏

非常好玩, 考点齐全, 复习了一遍, 希望下次出需要用特别特别简单的脚本的题。记录wp顺便拿来当复习了。本来想着按照做题顺序写, 后来发现有点乱, 还是按照题目序号写

题目描述

想要我的flag吗? 如果想要的话, 那就去找吧, 我把全部的flag都放在那里。

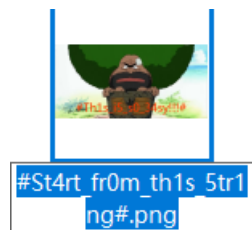
下载地址<https://ctfshow.lanzous.com/iDEiVmr8m4d>

本场比赛共有18题, 但只有1个附件文件(见第1题), 所有flag均可以从附件中获取;

- 1、所有的flag开头和结尾均为#, 中间由字母、数字或下划线组成;
- 2、本场比赛不使用任何可以设置密码的隐写方法, 包括可以将密码留空的隐写方法;
- 3、原理类似的隐写方法在确保不互相干扰的前提下可能会以多种方式使用;
- 4、如果从附件提取的隐写信息为字符串形式, 可能需要转码得到指定格式的结果;
- 5、如果从附件提取的隐写信息为另一张图片, 该图片不会再包含隐写信息, 即不存在套娃隐写;
- 6、所使用的字体均为微软雅黑, 若有字符无法分辨, 请与字体对比查看;
- 7、取得类似#abcd_1234#的字符串后, 请计算其MD5值(包含头尾的#号);
- 8、每道题目都给出了一段MD5值, 请找到MD5值匹配的题目后, 将flag包上ctfshow{}格式提交。

One PieNG 1

下载附件，图片名字即为flag



```
ctfshow{#St4rt_fr0m_th1s_5tr1ng#}
```

One PieNG 2



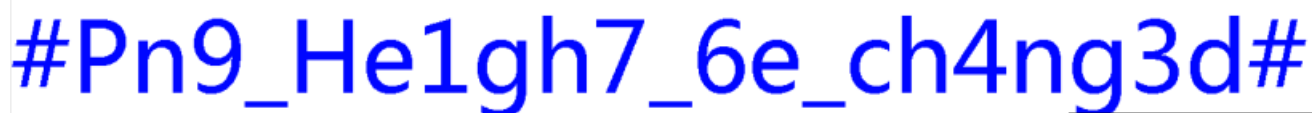
```
ctfshow{#Th1s_i5_s0_34sy!!!#}
```

One PieNG 3

修改图片高度

```
89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  
00 00 05 56 00 00 02 97 08 06 00 00 00 AB 21 2A  
35 00 00 00 16 74 45 58 74 41 72 74 69 73 74 00
```

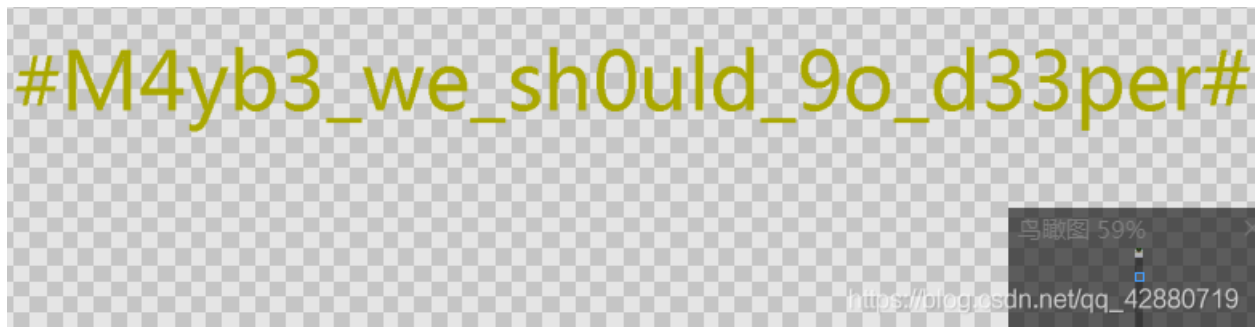
习惯直接把0改成9，就直接出了3题和4题



```
ctfshow{#Pn9_He1gh7_6e_ch4ng3d#}
```

One PieNG 4

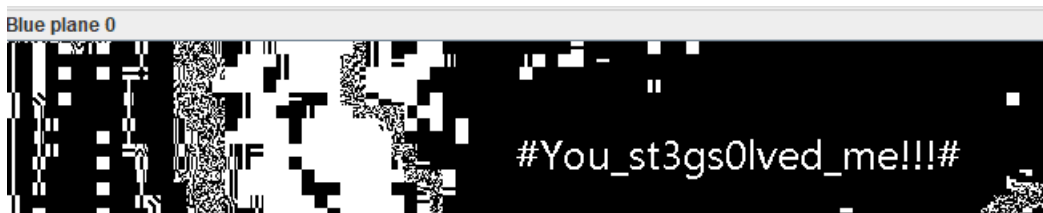
再往下拉就能发现flag



```
ctfshow{#M4yb3_we_sh0uld_9o_d33per#}
```

One PieNG 5

先把高度调回去，然后用stegsolve打开
B通道最低位隐写

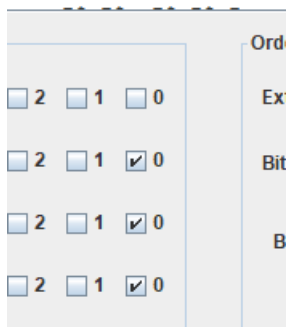


```
ctfshow{#You_st3gs0lved_me!!!#}
```

One PieNG 6

LSB隐写，使用stegsolve的data extract，选上RGB的0通道

```
#LSB_ls_v3ry_e4s  
y_righ7? #m..m.$.  
I.m..m.8 .vUZ...U  
..v.m.IZ ...mV.j.  
.m..m..m ..m..m..  
[]_c.7$m ...I$.I$.  
m..m...I $.I$.I$.  
I$.I$.I$. .I$.I$.I  
$.I$m... I$.I$.I$.  
.I$.I$m. ..I$.I$m
```

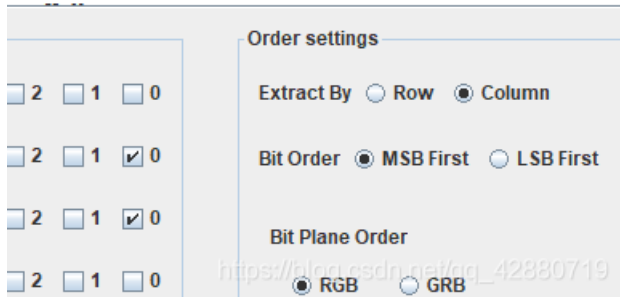


```
ctfshow{#LSB_1s_v3ry_e4sy_righ7?#}
```

One PieNG 7

考虑column，先选上RGB和column，发现可能存在flag，然后尝试去掉一些通道，最后在RG通道找到flag

```
#5ometlm es_LSB_g  
0es_colo mn_flr5t  
#?.....  
.....?....  
.....<?  
.....  
.?.....  
.....V. ...j..0.  
...UVVeY ..i<...?  
.c.U...Z Z....ZUU
```

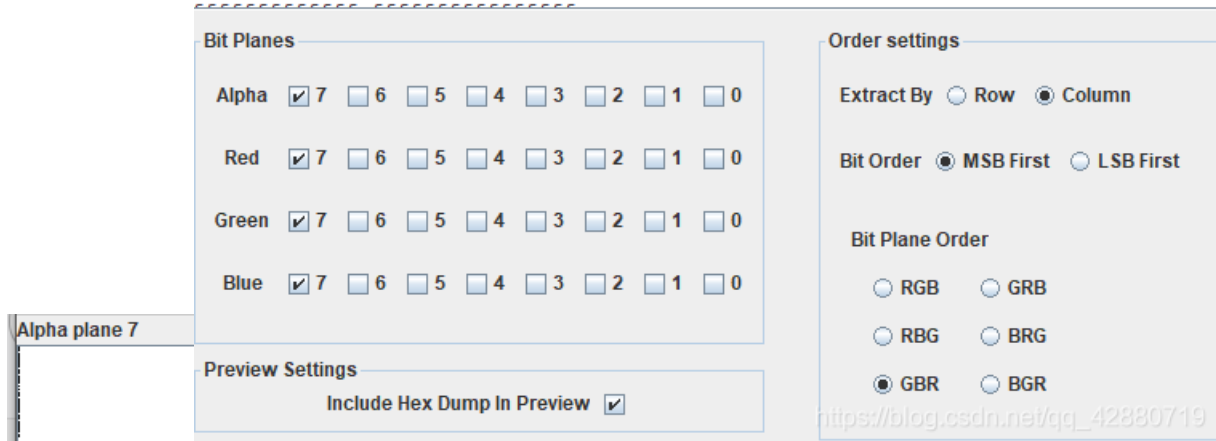


```
ctfshow{#5omet1mes_LSB_g0es_co1omn_f1r5t#}
```

One PieNG 8

考虑到R,G,B,A通道都能看到左上角有问题，并且是明显的问题，可能7通道存在flag

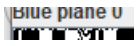
```
a737465675f64 6f33355f6e6f375f #zsteg_d o35_no7_  
1773479735f77 30726b23ffffffff alw4ys_w 0rk#....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....  
ffffffffffffff ffffffffffffffff .....
```



```
ctfshow{#zsteg_do35_no7_a1w4ys_w0rk#}
```

One PieNG 9

可以观察到，0通道的左上角的LSB隐写长度

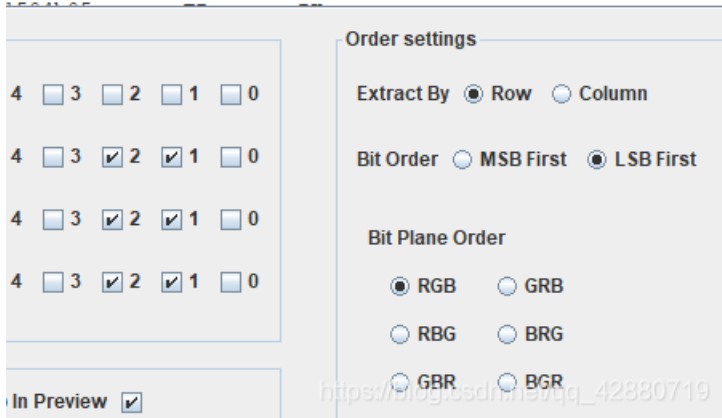


然后发现1通道和2通道长度明显比这个长，

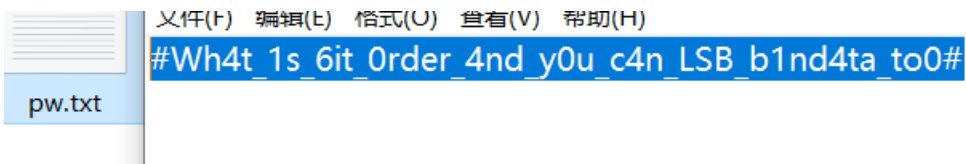


提取1、2通道，得到pk头，点击save bin，保存为1.zip

```
c529664 PK..... .qLR.d
0007077 ..l.../. .....pw
e37cb2c .txtS..0 ).7,.7.,
4288d4f .7(JI-.7 .K..4(.O
9498c2f 6....v.O 2.Kl)I./
0000008 .7P..PK. ....
02f0000 ..qLR.d. .l.../..
0000000 ...$. ....
0000000 ...pw.tx t. ....
146499b .....* .....FI.
```



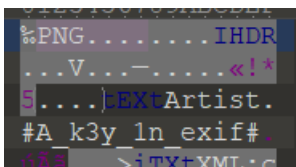
得到1.zip，解压即可得到flag



```
ctfshow{#Wh4t_1s_6it_Order_4nd_y0u_c4n_LSB_b1nd4ta_to0#}
```

One PieNG 10

010打开，得到flag



ctfshow{#A_k3y_1n_exif#}

One PieNG 11

在线去找一个EXIF查看器（也可以用010，也可以用PS）

<https://exif.tuchong.com/>

XMP-photoshop

DocumentAncestors	23415F6B65795F6672306D5F50683074307368307023
城市	b58/3AjtPrXQJuhFwguK7nqu4ZpsqMLwU

Composite

https://blog.csdn.net/qq_42880719

直接白给2个flag

b58/3AjtPrXQJuhFwguK7nqu4ZpsqMLwU即base58后面的内容

转换前:

3AjtPrXQJuhFwguK7nqu4ZpsqMLwU

编码Base58>

解码Base58>

转换后:

#An0th3r_key_1n_3xif#

ctfshow{#An0th3r_key_1n_3xif#}

One PieNG 12

同上，23415F6B65795F6672306D5F50683074307368307023转换

1	23415F6B65795F6672306D5F50683074307368307023
---	--

16进制转字符	字符转16进制	测试用例	清空结果	复制结果
---------	---------	------	------	------

PayPal

一个账户，收款全球。0费用开户，享卖家保障，赢逾2亿

1	#A_key_fr0m_Ph0t0sh0p#	https://blog.csdn.net/qq_42880719
---	------------------------	---

```
ctfshow{#A_key_fr0m_Ph0t0sh0p#}
```

One PieNG 13

010查看变量窗口（打开方式：视图-检查器窗口-变量，需要下载png模板，点击模板-模板储存库-png模板）

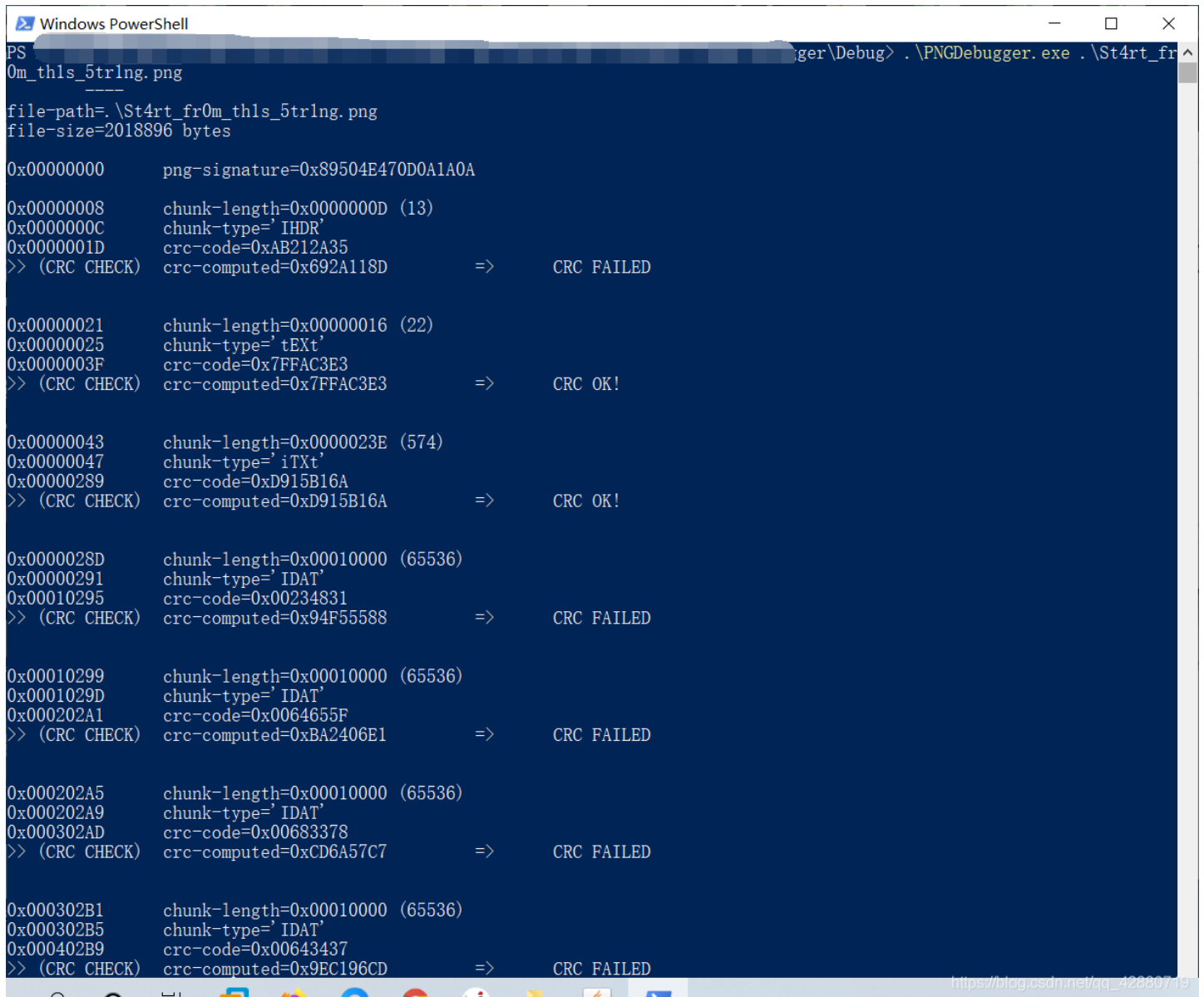
> struct PNG SIGNATURE ...		0h	8h	Fq:	Bq:	
> struct PNG CHUNK chu...	IHDR (Critical,...	8h	19h	Fq:	Bq:	
> struct PNG CHUNK chu...	tEXt (Ancillar...	21h	22h	Fq:	Bq:	
> struct PNG CHUNK chu...	iTXt (Ancillary...	43h	24Ah	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	28Dh	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	10299h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	202A5h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	302B1h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	402BDh	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	502C9h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	602D5h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	702E1h	1000Ch	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	802EDh	2652h	Fq:	Bq:	
> struct PNG CHUNK chu...	tEXt (Ancillar...	8293Fh	30h	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	8296Fh	F857h	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	921C6h	10000h	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	A21C6h	10000h	Fq:	Bq:	
> struct PNG CHUNK chu...	IDAT (Critical,...	B21C6h	10000h	Fq:	Bq:	

```
.... u..B}4...
..$tXEt.....
.#Ju5t_a_lonely
tEXt_chunk#-n$<<.
```

```
ctfshow{#Ju5t_a_1one1y_tEXt_chunk#}
```

One PieNG 14

也是常规考点，我先用PNGdebugger跑了一下



```
Windows PowerShell
PS C:\Users\user> .\PNGDebugger.exe .\St4rt_fr
0m_thls_5trlng.png
file-path=. \St4rt_fr0m_thls_5trlng.png
file-size=2018896 bytes
0x00000000      png-signature=0x89504E470D0A1A0A
0x00000008      chunk-length=0x0000000D (13)
0x0000000C      chunk-type=' IHDR'
0x0000001D      crc-code=0xAB212A35
>> (CRC CHECK)  crc-computed=0x692A118D      =>      CRC FAILED
0x00000021      chunk-length=0x00000016 (22)
0x00000025      chunk-type=' tEXt'
0x0000003F      crc-code=0x7FFAC3E3
>> (CRC CHECK)  crc-computed=0x7FFAC3E3      =>      CRC OK!
0x00000043      chunk-length=0x0000023E (574)
0x00000047      chunk-type=' iTXt'
0x00000089      crc-code=0xD915B16A
>> (CRC CHECK)  crc-computed=0xD915B16A      =>      CRC OK!
0x0000028D      chunk-length=0x00010000 (65536)
0x00000291      chunk-type=' IDAT'
0x00010295      crc-code=0x00234831
>> (CRC CHECK)  crc-computed=0x94F55588      =>      CRC FAILED
0x00010299      chunk-length=0x00010000 (65536)
0x0001029D      chunk-type=' IDAT'
0x000202A1      crc-code=0x0064655F
>> (CRC CHECK)  crc-computed=0xBA2406E1      =>      CRC FAILED
0x000202A5      chunk-length=0x00010000 (65536)
0x000202A9      chunk-type=' IDAT'
0x000302AD      crc-code=0x00683378
>> (CRC CHECK)  crc-computed=0xCD6A57C7      =>      CRC FAILED
0x000302B1      chunk-length=0x00010000 (65536)
0x000302B5      chunk-type=' IDAT'
0x000402B9      crc-code=0x00643437
>> (CRC CHECK)  crc-computed=0x9EC196CD      =>      CRC FAILED
```

然后记得备份一份，使用tweakpng删掉前面出错的9个IDAT块（这9个IDAT块就是一打开图片看到的那个）

IDAT	65536	94f555...	critical	PNG image data
IDAT	65536	ba240...	critical	PNG image data
IDAT	65536	cd6a5...	critical	PNG image data
IDAT	65536	9ec19...	critical	PNG image data
IDAT	65536	1d1c5...	critical	PNG image data
IDAT	65536	d41fca...	critical	PNG image data
IDAT	65536	655d5...	critical	PNG image data
IDAT	65536	cb187...	critical	PNG image data
IDAT	9798	19fe70...	critical	PNG image data
tXEt	36	2d6ea...	ancillary, safe to c...	unrecognized chunk type
IDAT	63563	0639e...	critical	PNG image data
IDAT	65524	0b24d...	critical	PNG image data
IDAT	65536	74110...	critical	PNG image data

IDAT 0524 /d1c0... critical
IDAT 65524 03708 critical

https://blog.csdn.net/qq_42880719
PNG image data
PNG image data

删掉tXEt和前面的IDAT，然后save保存



ctfshow{#eXtr4_IDAT_of_an0th3r_Pn9#}

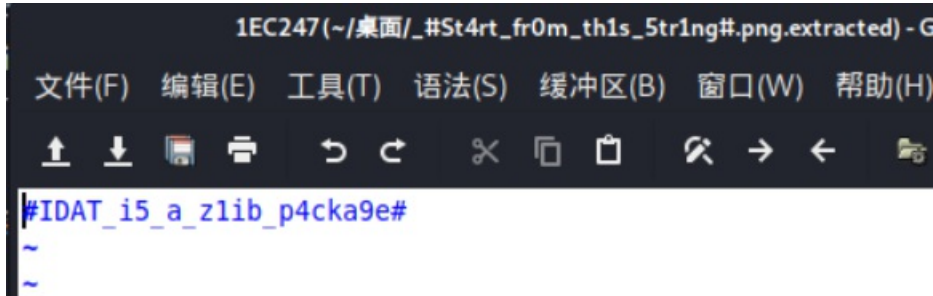
One PieNG 15

使用binwalk，得到的文件我是一个个翻的

官解<https://www.w3.org/TR/PNG/>

更正后的原理：

zlib不是一个压缩算法，说到“zlib压缩数据”之类的概念时，里面zlib的意思更接近指代一种压缩数据的存放格式。同时，PNG的压缩算法虽然预留了相应标记位，但是目前仍只支持一种算法，同时也是zlib库目前唯一支持的算法，LZ77的变种，DEFLATE所以IDAT块中的数据是以LZ77变种算法DEFLATE压缩后以zlib格式储存的



```
ctfshow{#IDAT_i5_a_zlib_p4cka9e#}
```

One PieNG 16

这题太坑了，告辞。

之前出错的IDAT块

```
le=0x00234831
puted=0x94F55588      =>      CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
le=0x0064655F
puted=0xBA2406E1      =>      CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
le=0x00683378
puted=0xCD6A57C7      =>      CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
le=0x00643437
puted=0x9EC196CD      =>      CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
le=0x00615F31
puted=0x1D1C51CC      =>      CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
le=0x006E5F63
puted=0xD41FCAD9      =>      CRC FAILED
```

```
length=0x00010000 (65536)
type=' IDAT'
file=0x0068756E
computed=0x655D563D => CRC FAILED

length=0x00010000 (65536)
type=' IDAT'
file=0x006B5F43
computed=0xCB1875FD => CRC FAILED
```

能发现都是00开头，后面几乎都是6，并且是16进制，猜想转ASCII
一转就出flag

23483164655F683378643437615F316E5F6368756E6B5F43524323

```
1 23483164655F683378643437615F316E5F6368756E6B5F43524323|
```

16进制转字符 字符转16进制 测试用例 清空结果 复制结果

亿速云服务器免备案CN2高速直连

亿速云，CN2高速稳定独享带宽目前还有优惠活动低至29元每月，香港服务器

```
1 #H1de_h3xd47a_1n_chunk_CRC#
```

https://blog.csdn.net/qq_42880719

```
ctfshow{#H1de_h3xd47a_1n_chunk_CRC#}
```

One PieNG 17

文件尾

```
END# #HexEditor
r_will_b3_helpfu
1#%PNG.....IH
DR.....}
```

```
ctfshow{#HexEditor_wi11_b3_he1pfu1#}
```

One PieNG 18

foremost分离出另一张图片（虽说刚刚binwalk分过了）



#He110_I_4m_Tw0_PieNG#

0000000.png

00003937.png

#He110_I_4m_Tw0_PieNG#

```
ctfshow{#He110_I_4m_Tw0_PieNG#}
```

****One PieNG问卷调查 ****

```
ctfshow{套娃终有报，天道好轮回。不信抬头看，苍天饶过谁。}
```

建议出进阶，建议用python，建议使用工具，不建议套娃

