

# ctfshow - PNG隐写入门赛

原创

[H3rmesk1t](#) 于 2021-09-13 13:58:31 发布 217 收藏 1

分类专栏: [Misc](#) 文章标签: [ctfshow](#) [Misc](#) [PNG](#) [隐写](#) [LSB](#)

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/LYJ20010728/article/details/120264897>

版权



[Misc](#) 专栏收录该内容

10 篇文章 1 订阅

订阅专栏

## ctfshow - PNG隐写入门赛

前言

[One PieNG 1](#)

[One PieNG 2](#)

[One PieNG 3](#)

[One PieNG 4](#)

[One PieNG 5](#)

[One PieNG 6](#)

[One PieNG 7](#)

[One PieNG 8](#)

[One PieNG 9](#)

[One PieNG 10](#)

[One PieNG 11](#)

[One PieNG 12](#)

[One PieNG 13](#)

[One PieNG 14](#)

[One PieNG 15](#)

[One PieNG 16](#)

[One PieNG 17](#)

[One PieNG 18](#)

[One PieNG问卷调查](#)

## 前言

- [题目下载链接](#)

1. 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；
2. 所有的flag开头和结尾均为#，中间由字母、数字或下划线组成；
3. 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；
4. 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；
5. 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；
6. 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；
7. 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；
8. 取得类似#abcd\_1234#的字符串后，请计算其MD5值（包含头尾的#号）；
9. 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

- 图片的名字就是 One PieNG 1 的 Flag: `ctfshow{#St4rt_fr0m_th1s_5tr1ng#}`

## One PieNG 2

- 图片上面直接给出了 One PieNG 2 的 Flag: `ctfshow{#Th1s_i5_s0_34sy!!!#}`



## One PieNG 3

- 用蜂蜜浏览器查看图片时发现载入失败，猜测图片的宽高可能有问题，将图片修复后拿到 One PieNG 3 的 Flag: `ctfshow{#Pn9_He1gh7_6e_ch4ng3d#}`



#Pn9\_He1gh7\_6e\_ch4ng3d#

CSDN @H3rmesk1t

## One PieNG 4

- 尝试将图片的高度再拉大点，发现了 One PieNG 4 的 Flag: `ctfshow{#M4yb3_we_sh0uld_9o_d33per#}`

#Pn9\_He1gh7\_6e\_ch4ng3d#

#M4yb3\_we\_sh0uld\_9o\_d33per#

CSDN @H3rmesk1t

## One PieNG 5

- 用 `stegsolve` 查看图片，在 Blue 通道的最低位发现 Flag: `ctfshow{#You_st3gs0lved_me!!!#}`



## One PieNG 6

- 用 `stegsolve` 查看图片，在 RGB 三通道最低位发现 Flag: `ctfshow{#LSB_1s_v3ry_e4sy_righ7?#}`

Extract Preview

```

234c53425f31735f 763372795f653473 #LSB_ls_v3ry_e4s
795f72696768373f 236db6db6db624ec y_righ7? #m..$.
49db6db6db6db638 e276555ab614c155 I.m..m.8 .vUZ...U
c71c76db6db6495a 92b6db6d56a56ab6 ..v.m.IZ ...mV.j.
db6db6db6db6db6d b6db6db6db6db7f2 .m..m..m ..m..m..
7f5fb2639237246d b6db924924924924 □_c.7$m ...I$.I$
6db6db6db6db9249 2492492492492492 m..m...I $.I$.I$.
4924924924924924 9249249249249249 I$.I$.I$ .I$.I$.I
249249246db6db92 4924924924924924 $.I$m... I$.I$.I$
9249249249246db6 db9249249249246d .I$.I$m. ..I$.I$m
                
```

**Bit Planes**

Alpha  7  6  5  4  3  2  1  0

Red  7  6  5  4  3  2  1  0

Green  7  6  5  4  3  2  1  0

Blue  7  6  5  4  3  2  1  0

**Preview Settings**

Include Hex Dump In Preview

**Order settings**

Extract By  Row  Column

Bit Order  MSB First  LSB First

**Bit Plane Order**

RGB  GRB

RBG  BRG

GBR  BGR

Preview
Save Text
Save Bin
Cancel

CSDN @H3rmesk1t

## One PieNG 7

- 用 `stegsolve` 查看图片，在 RG 两通道最低位发现 Flag: `ctfshow{#5omet1mes_LSB_g0es_co1omn_f1r5t#}`

Extract Preview

```

23356f6d6574316d 65735f4c53425f67 #5ometlm es_LSB_g
3065735f636f316f 6d6e5f6631723574 0es_colo mn_flr5t
233fffffff00000000 0000000000000000 #?.....
ffffffffffff30c fffffc3f00000000 .....?....
000000000000ffff 0000ffff3c3f .....<?
fff00000000ffff 0000fff03c003c0 .....
fc3fffffffff0000 0000fff00000000 .?.....
0000000000055695 aaaaa96a000030cf .....V. ...j..0.
aaffff5556566559 00ff693caaaa3c3f ...UVVeY ..i<...?
a663ff55000a035a 5a95aaaa955a5555 .c.U...Z Z....ZUU

```

**Bit Planes**

Alpha  7  6  5  4  3  2  1  0

Red  7  6  5  4  3  2  1  0

Green  7  6  5  4  3  2  1  0

Blue  7  6  5  4  3  2  1  0

**Preview Settings**

Include Hex Dump In Preview

**Order settings**

Extract By  Row  Column

Bit Order  MSB First  LSB First

**Bit Plane Order**

RGB  GRB

RBG  BRG

GBR  BGR

Preview Save Text Save Bin Cancel

CSDN @H3rmesk1t

## One PieNG 8

- 用 `stegsolve` 查看图片，发现 R,G,B,A 通道都能看到左上角有问题，在最高位发现 Flag: `ctfshow{#zsteg_do35_no7_a1w4ys_w0rk#}`

Extract Preview

```

237a737465675f64 6f33355f6e6f375f #zsteg_d o35_no7_
6131773479735f77 30726b23fffffffff alw4ys_w 0rk#....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....
fffffffffffffffff ffffffffffffffffff .....

```

**Bit Planes**

Alpha  7  6  5  4  3  2  1  0

Red  7  6  5  4  3  2  1  0

Green  7  6  5  4  3  2  1  0

Blue  7  6  5  4  3  2  1  0

**Preview Settings**

Include Hex Dump In Preview

**Order settings**

Extract By  Row  Column

Bit Order  MSB First  LSB First

**Bit Plane Order**

RGB  GRB

RBG  BRG

GBR  BGR

Preview Save Text Save Bin Cancel

CSDN @H3rmesk1t

## One PieNG 9

- 用 `stegsolve` 查看图片，提取 RGB 的 1,2 通道，得到一个压缩包，将其 `save bin` 保存下来后打开发现 Flag: `ctfshow{#Wh4t_1s_6it_0rder_4nd_y0u_c4n_LSB_b1nd4ta_to0#}`



Extract Preview

```

504b030414000000 080088714c529664 PK..... ..qLR.d
2eca310000002f00 000006000000070777 ..l.../. .....pw
2e747874530ecf30 2989372c8e37cb2c .txtS..0 ).7,.7.,
8937284a492d8a37 c94b89af34288d4f .7(JI-.7 .K..4(.O
36c98bf709768a4f 32cc4b3129498c2f 6....v.O 2.Kl)I./
c937500600504b01 021f0014000000008 .7P..PK. ....
0088714c5296642e ca310000002f0000 ..qLR.d. .l.../..
0006002400000000 0000002000000000 ...$. ....
00000070772e7478 740a002000000000 ...pw.tx t...
0001001800b7e72a 030601d70146499b .....* .....FI.

```

**Bit Planes**

Alpha  7  6  5  4  3  2  1  0

Red  7  6  5  4  3  2  1  0

Green  7  6  5  4  3  2  1  0

Blue  7  6  5  4  3  2  1  0

**Preview Settings**

Include Hex Dump In Preview

**Order settings**

Extract By  Row  Column

Bit Order  MSB First  LSB First

**Bit Plane Order**

RGB  GRB

RBG  BRG

GBR  BGR

Preview Save Text Save Bin Cancel

CSDN @H3rmesk1t

## One PieNG 10

- 用 `zsteg` 查看图片发现 `One PieNG 10` 的 Flag: `ctfshow{#A_k3y_1n_exif#}`

```

meta Artist .. text: "#A_k3y_1n_exif#"
meta XML:com.adobe.xmp.. Traceback (most recent call last):
  18: from /usr/local/bin/zsteg:23:in <main>
  17: from /usr/local/bin/zsteg:23:in `load'
  16: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/bin/zsteg:8:in <top (required)>
  15: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg.rb:30:in `run'
  14: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:151:in `run'
  13: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:151:in `each_with_index'
  12: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:151:in `each'
  11: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:158:in `block in run'
  10: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:158:in `each'
   9: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:162:in `block (2 levels) in run'
   8: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/cli/cli.rb:245:in `check'
   7: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:72:in `check'
   6: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:156:in `check_metadata'
   5: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:156:in `each'
   4: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:159:in `block in check_metadata'
   3: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:284:in `process_result'
   2: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker.rb:362:in `data2regid'
   1: from /var/lib/gems/2.7.0/gems/zsteg-0.2.4/lib/zsteg/checker/zlib.rb:24:in `check_data

```

CSDN @H3rmesk1t

## One PieNG 11

- 根据 `One PieNG 10` Flag 内容的提示，查看一下图片的 EXIF，将 `City` 选项 base58 解密拿到 `One PieNG 11` 的 Flag: `ctfshow{#An0th3r_key_1n_3xif#}`

**Recipe**

From Base58

Alphabet  
123456789ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuv...

Remove non-alphabet chars

**Input** length: 29  
lines: 1

```
3AjtPrXQ3uhFwguK7nqu4ZpsqMLwU
```

**Output** time: 1ms  
length: 21  
lines: 1

```
#An0th3r_key_1n_3xif#
```

CSDN @H3rmesk1t

## One PieNG 12

- 根据 **One PieNG 10** Flag 内容的提示，查看一下图片的 EXIF，将 **Document Ancestors** 选项十六进制转字符串拿到 **One PieNG 12** 的 Flag: `ctfshow{#A_key_fr0m_Ph0t0sh0p#}`

加密或解密字符串长度不可以超过10M 当前长度: 44

```
1 23415F6865795F6672306D5F50683074307368307023
```

---

≡

16进制转字符
字符转16进制
测试用例
清空结果
复制结果

```
1 #A_key_fr0m_Ph0t0sh0p#
```

CSDN @H3rmesk1t

## One PieNG 13

- 用 010editor 打开，查看每一个块，发现 **One PieNG 13** 的 Flag: `ctfshow{#Ju5t_a_1one1y_tEXt_chunk#}`

```

8:2870h: 00 20 12 56 01 00 00 00 00 22 61 15 00 00 00 00 . .V....."a.....
8:2880h: 20 12 56 01 00 00 00 00 22 61 15 00 00 00 00 20 . .V....."a.....
8:2890h: 12 56 01 00 00 00 00 22 61 15 00 00 00 00 20 12 . .V....."a.....
8:28A0h: 56 01 00 00 00 00 22 61 15 00 00 00 00 20 12 56 V....."a..... .V
8:28B0h: 01 00 00 00 00 22 61 15 00 00 00 00 20 12 56 01 . .V....."a.....
8:28C0h: 00 00 00 00 22 61 15 00 00 00 20 12 56 01 00 . .V....."a.....
8:28D0h: 00 00 00 22 61 15 00 00 00 00 20 12 56 01 00 . .V....."a.....
8:28E0h: 00 00 22 61 15 00 00 00 00 20 12 56 01 00 00 . .V....."a.....
8:28F0h: 00 22 61 15 00 00 00 00 20 12 56 01 00 00 00 . .V....."a.....
8:2900h: 22 61 15 00 00 00 00 20 12 56 01 00 00 00 22 "a..... .V....."
8:2910h: 61 15 00 00 00 00 20 12 56 01 00 00 00 22 61 a..... .V....."a
8:2920h: 15 00 00 00 00 20 12 56 01 00 00 00 00 22 61 15 . .V....."a
8:2930h: 00 00 00 00 20 FA 1F 0C 42 7D 34 00 52 43 23 00 . .V....."a
8:2940h: 00 00 24 74 58 45 74 00 00 00 00 00 00 00 00 . .V....."a
8:2950h: 00 23 4A 75 35 74 5F 61 5F 31 6F 6E 65 31 79 5F . .V....."a
8:2960h: 74 45 58 74 5F 63 68 75 6E 6B 23 2D 6E A7 AB 00 tEXt_chunk#-n9q.
8:2970h: 00 F8 4B 43 44 41 54 78 5E EC BD 09 80 24 49 55 .øKIDA;x^i%.€$IU
8:2980h: 3E FE EA EE BB 67 A6 67 76 66 EF 03 70 77 41 04 >þêi»g|gvfi.pwA.
8:2990h: 61 11 41 01 41 39 04 44 90 95 53 51 11 44 6E B9 a.A.A9.D.·SQ.Dn'
8:29A0h: 45 94 1B 3C 38 04 94 CB BF 28 8A A0 82 88 80 80 E".<8."É¿(Š ,,"€€
8:29B0h: 80 DC F0 43 40 0E 11 D8 03 77 D9 7B E7 9E BE BB €ÜòC@..ø.wÜ{çž¼»
8:29C0h: EB FE BF EF 45 BE AA A8 E8 88 CC AA CA AE EE 9E ëþ¿iE¼ª"è`IªÉ@iž
8:29D0h: 99 FC 66 5E 67 66 9C 2F 5E BC 88 CC 78 F5 32 32 ¢üf^gfe/^%`ixð22
8:29E0h: F7 BE 2F 1E 6E 53 86 0C 19 32 64 C8 90 21 43 86 ÷¼/.nSt..2dÉ.!Cf
8:29F0h: 0C 19 32 64 C8 90 21 43 86 0C 19 4E 49 3C F6 5E ..2dÉ.!Cf..NI<ó^

```

CSDN @H3rmesk1t

## One PieNG 14

- 由于 `zsteg` 提示数据块异常，用 `pngdebuger` 跑一下，发现九个出错的数据块，先将其提取出来，将图片的这九个数据块删去，拿到 `One PieNG 14` 的 Flag: `ctfshow{#eXtr4_IDAT_of_an0th3r_Pn9#}`



```

>> (CRC CHECK) crc-computed=0xd915b16a => CRC OK!

0x00000280 chunk-length=0x00010000 (65536)
0x00000291 chunk-type='IDAT'
0x00010295 crc-code=0x00234831
>> (CRC CHECK) crc-computed=0x94f55588 => CRC FAILED

0x00010299 chunk-length=0x00010000 (65536)
0x00010290 chunk-type='IDAT'
0x000202A1 crc-code=0x0064655f
>> (CRC CHECK) crc-computed=0x8a2406f1 => 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

0x000402BD chunk-length=0x00010000 (65536)
0x000402C1 chunk-type='IDAT'
0x000502C5 crc-code=0x00615f31
>> (CRC CHECK) crc-computed=0xd1c151cc => CRC FAILED

0x000502C9 chunk-length=0x00010000 (65536)
0x000502CD chunk-type='IDAT'
0x000602D1 crc-code=0x006e5f63
>> (CRC CHECK) crc-computed=0xd41fcad9 => CRC FAILED

0x000602D5 chunk-length=0x00010000 (65536)
0x000602D9 chunk-type='IDAT'
0x000702DD crc-code=0x0068756e
>> (CRC CHECK) crc-computed=0x655d563d => CRC FAILED

0x000702E1 chunk-length=0x00010000 (65536)
0x000702E5 chunk-type='IDAT'
0x000802E9 crc-code=0x00685f43
>> (CRC CHECK) crc-computed=0xcb1875fd => CRC FAILED

0x000802ED chunk-length=0x00002646 (9798)
0x000802F1 chunk-type='IDAT'
0x0008293B crc-code=0x00524323
>> (CRC CHECK) crc-computed=0x19fe78d3 => CRC FAILED

0x0008293f chunk-length=0x00000024 (36)
0x00082943 chunk-type='IDAT'
0x0008296B crc-code=0x2d6ea7ab
>> (CRC CHECK) crc-computed=0x2d6ea7ab => CRC OK!

```

CSDN @H3rmesk1t



#eXtr4\_IDAT\_of\_an0th3r\_Pn9#

CSDN @H3rmesk1t

## One PieNG 15

- 由于 binwalk 发现了异常数据块，将其分离出来查看，发现 One PieNG 15 的 Flag: `ctfshow{#IDAT_i5_a_z1ib_p4cka9e#}`

```

kali@kali ~/Desktop/_1.png.extracted$ ls
1EC247 1EC247.zlib 1EC2BB 1EC2BB.zlib 295 295.zlib 82977 82977.zlib
kali@kali ~/Desktop/_1.png.extracted$ cat 1EC247
#IDAT_i5_a_z1ib_p4cka9e#
kali@kali ~/Desktop/_1.png.extracted$

```

## One PieNG 16

- 发现前面提取出来的出错的 CRC 数据块拼接起来和之前的十六进制很像，转一下拿到 One PieNG 16 的 Flag: `ctfshow{#H1de_h3xd47a_1n_chunk_CRC#}`

```
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

```
1 #H1de_h3xd47a_1n_chunk_CRC#
```



CSDN @H3rmesk1t

## One PieNG 17

- 用 `zsteg` 查看图片发现 One PieNG 17 的 Flag: `ctfshow{#HexEditor_wi11_b3_he1pfu1#}`

```
extradata:0  ..
00000000: 23 48 65 78 45 64 69 74 6f 72 5f 77 69 31 31 5f #HexEditor_wi11_
00000010: 62 33 5f 68 65 31 70 66 75 31 23 89 50 4e 47 0d b3_he1pfu1#.PNG.
00000020: 0a 1a 0a 00 00 00 0d 49 48 44 52 00 00 02 18 00 .....IHDR.....
00000030: 00 00 3a 08 02 00 00 00 7d 1d b7 53 00 00 0b 85 ..:.....}..S....
00000040: 49 44 41 54 78 5e ed 9c 3d 8f dc bc 0e 85 df ff IDATx^..=.....
00000050: ff a7 d2 06 5b 2e b0 d5 36 29 02 04 48 10 6c b1 ...[ ... 6) ..H.l.
00000060: 41 52 a4 c8 eb b1 2d 89 a2 25 8a d4 87 e5 99 3d AR....-..%.....=
00000070: 0f 54 dc 3b 6b cb 14 79 c8 e3 99 e0 de ff fe 01 .T.;k..y.....
00000080: 00 00 00 0d c0 48 00 00 00 34 01 23 01 00 00 d0 ....H ... 4.#....
00000090: 04 8c 04 00 00 40 13 30 12 00 00 00 4d c0 48 00 ....@.0....M.H.
000000a0: 00 00 34 01 23 01 00 00 d0 04 8c 04 00 00 40 13 ..4.#.....@.
000000b0: 30 12 00 00 00 4d c0 48 00 00 00 34 01 23 01 00 0....M.H...4.#..
000000c0: 00 d0 04 8c 04 00 70 ef fc 79 fd f4 ed f3 6d bd .....p..y...m.
000000d0: fd dc 3f 01 a7 52 6d 24 7f bf 3e a5 2b f7 f3 65 ..?..Rm$..>.+..e
000000e0: fb fc c7 d7 5f fb 27 07 7c d5 bf 7d 7e 7a ff bd ...._.'|..}~z..
000000f0: 7f a8 c1 3f 54 de 7f 20 eb e9 5a c5 5a bf c9 af ...?T.. ..Z.Z...
```

CSDN @H3rmesk1t

## One PieNG 18

- `binwalk` 发现还有一张图片, 分离出来上面的内容就是 One PieNG 18 的 Flag: `ctfshow{#He110_I_4m_Tw0_PieNG#}`

# #He110\_I\_4m\_Tw0\_PieNG#

CSDN @H3rmesk1t

## One PieNG问卷调查

- 没意思

ctfshow{套娃终有报，天道好轮回。不信抬头看，苍天饶过谁。}

---