

PNG图片格式及隐写

原创

nginx123 于 2020-09-02 17:13:29 发布 3569 收藏 11

文章标签: [python](#) [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/nginx123/article/details/108359465>

版权

一个PNG图片的格式如下

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
0000h:	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
0010h:	00	00	08	00	00	00	04	B0	08	02	00	00	00	B3	57	D3°.....'WÓ
0020h:	7D	00	00	00	09	70	48	59	73	00	00	0B	13	00	00	0B	}...pHYs.....
0030h:	13	01	00	9A	9C	18	00	00	0A	4D	69	43	43	50	50	68	...šœ...MiCCPPh
0040h:	6F	74	6F	73	68	6F	70	20	49	43	43	20	70	72	6F	66	otoshop ICC prof

PNG图片固定以 89 50 4E 47 0D 0A 1A 0A (4个字节) 开头

接下来是IHDR区, 这个区标识着PNG文件的属性参数, 如长宽、色彩模式等, 具体如下:

08-0B: 4个字节标识IHDR区的大小(注意是去掉tag和crc32后的大小), 这里为00 00 00 0D, 即13个字节。

0C-0F: 4个字节, 固定为49 48 44 52(IHDR) 标识该区为IHDR区

10-13: 4个字节, 图片宽度, 此处为0x800=2048像素

14-17: 4个字节, 图片高度, 此处为0x04B0=1200像素

18: 1个字节 图像深度, 此处为8

19: 1个字节 颜色类型, 此处为2(真彩)

1A: 1个字节 压缩算法, 此处为0(无压缩)

1B: 1个字节 滤波方法, 此处为0(自适应滤波)

1C: 1个字节 隔行扫描, 此处为0(非隔行扫描)

1D-20: 4个字节, crc32校验值, 此处为 0xB357D37D

利用PNG图片进行隐写时, 有种方法是修改图片大小, 如将图片的大小由2048×1200修改为2048×900, 隐藏的300像素宽度里隐写了重要信息。

修改图片大小时, 往往不会修改CRC32校验值, 因此我们可以通过CRC32值, 还原出图片的原始大小。

代码如下:

```
import os
import binascii
import struct

crcbp = open("GreenScreenBG02.png", "rb").read() #打开图片
crc32frombp = int(crcbp[29:33].hex(),16) #读取图片中的CRC校验值
print(crc32frombp)

for i in range(4000): #宽度1-4000进行枚举
    for j in range(4000): #高度1-4000进行枚举
        data = crcbp[12:16] + \
            struct.pack('>i', i)+struct.pack('>i', j)+crcbp[24:29]
        crc32 = binascii.crc32(data) & 0xffffffff
        #print(crc32)
        if(crc32 == crc32frombp): #计算当图片大小为i:j时的CRC校验值，与图片中的CRC比较，当相同，则图片大小已经确定
            print(i, j)
            print('hex:', hex(i), hex(j))
```

运行结果:

```
3906191683
1920 1080
hex: 0x780 0x438
```

得到图片原始大小为1920×1080

修改png的hex值:

10-13: 0x00000780

14-17: 0x00000438

即可还原png图片