# png文件隐写——不负责任的总结

原创

RayLee23333 于 2019-11-12 16:36:05 发布 1171 收藏

分类专栏： ctf MISC



本文链接：https://blog.csdn.net/qq_33458986/article/details/103032337



ctf 同时被 2 个专栏收录

6 篇文章 0 订阅

订阅专栏

MISC

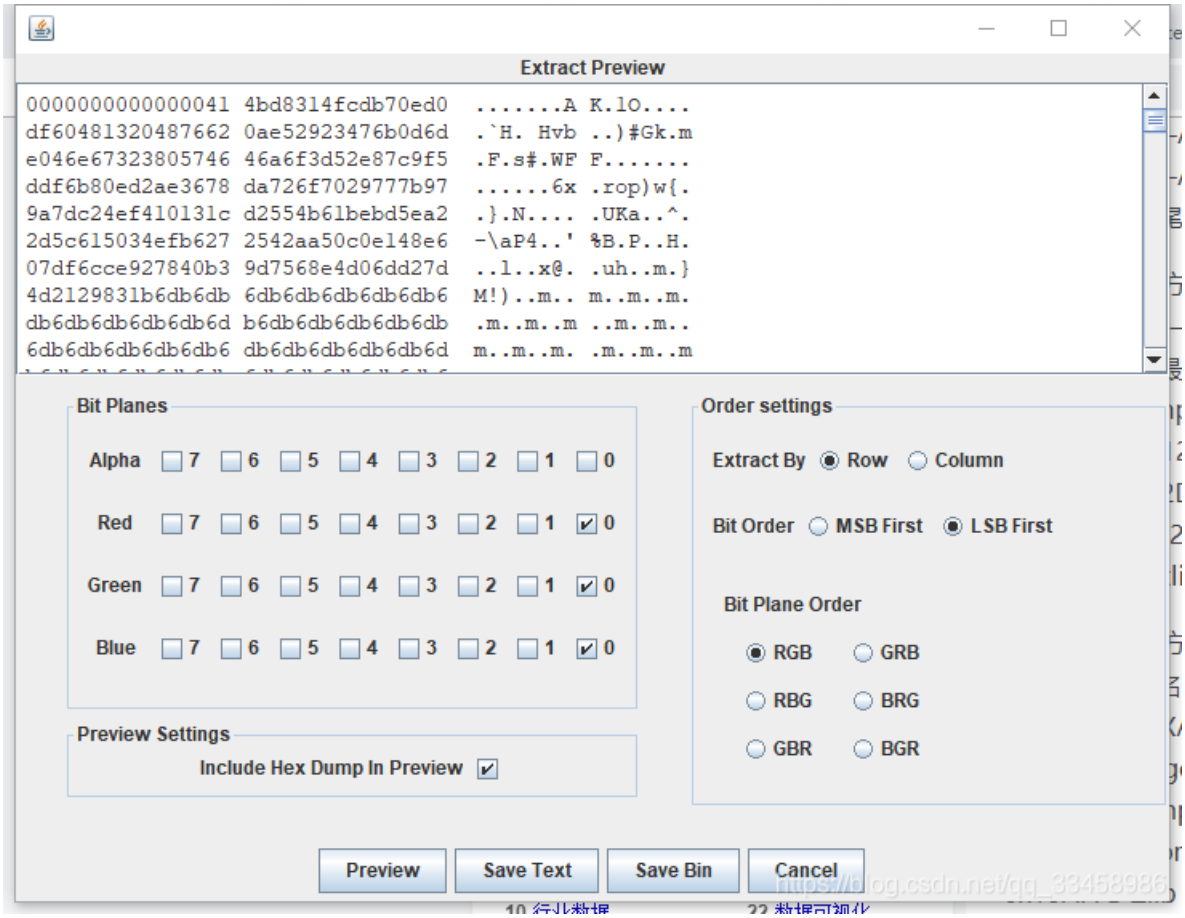
1 篇文章 0 订阅

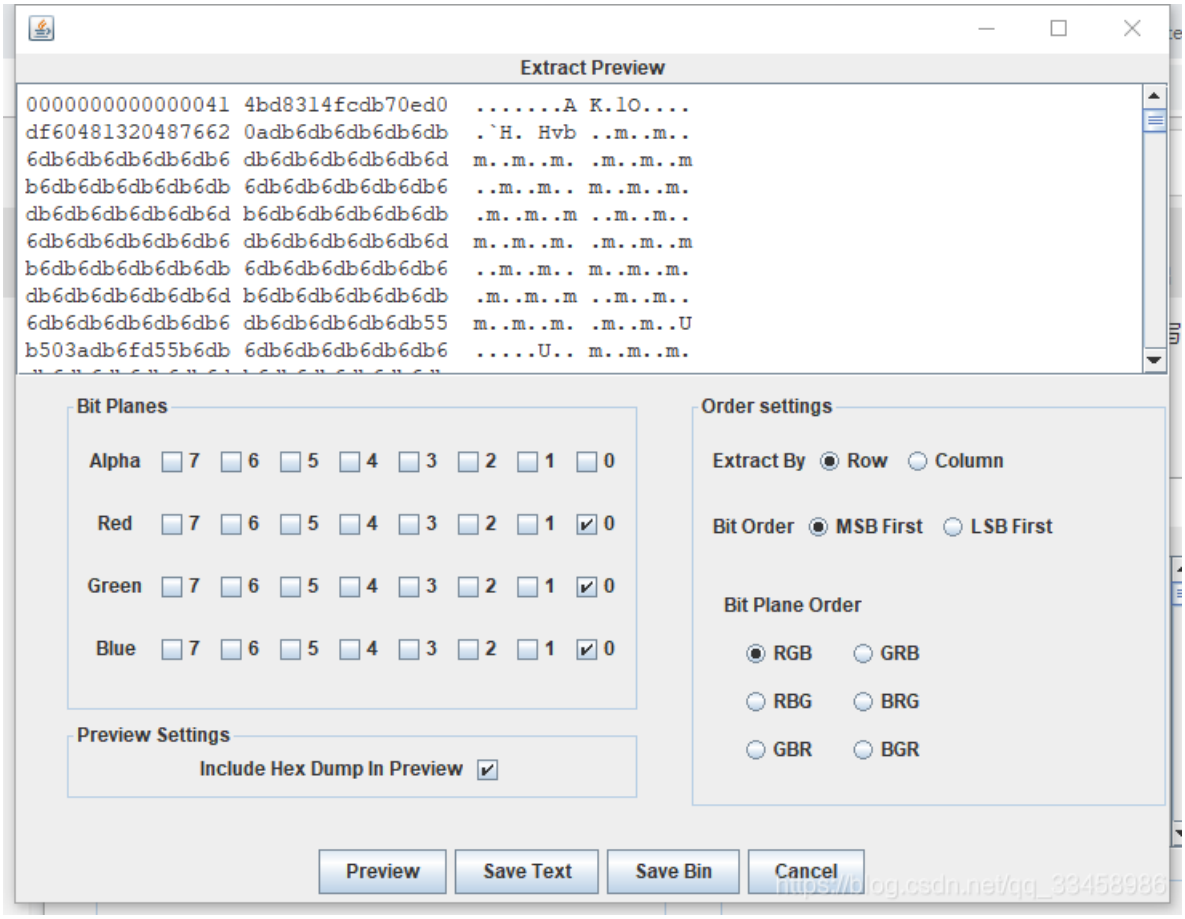
订阅专栏

为什么说不负责任呢，因为我自己也不是很了解这些隐写算法的内在逻辑，只是总结使用方法以及隐写后的图像特征（该总结过程并不是很严谨，希望各位看客也可以指出鄙人的错误）

## LSB隐写-LSB-Steganography工具

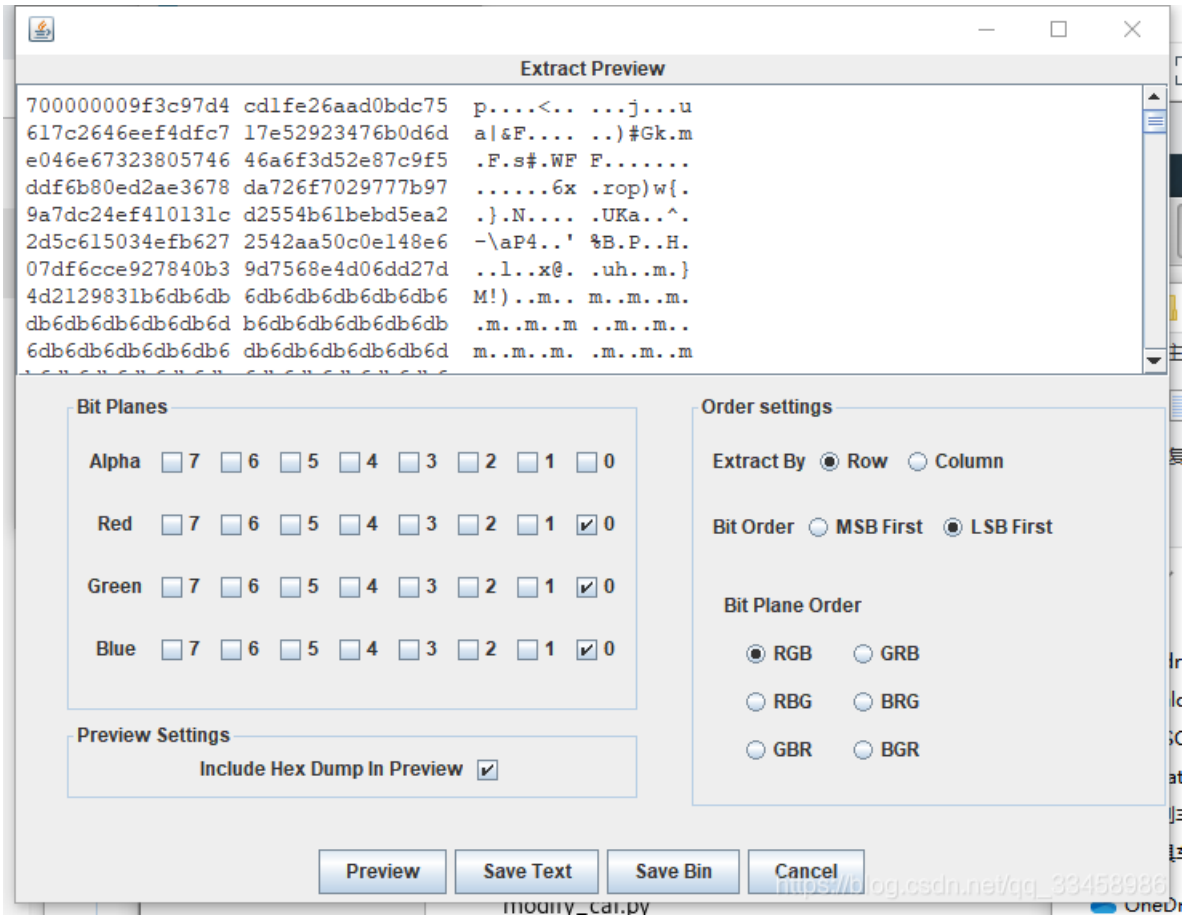
使用该隐写算法对两张图片分别进行隐写处理，再用stegSolve观察LSB得到下面两张图的结果，可以看出这种算法的特点就是前面全是0，所以比赛看到这种类型的时候就用这个方法试一试吧。

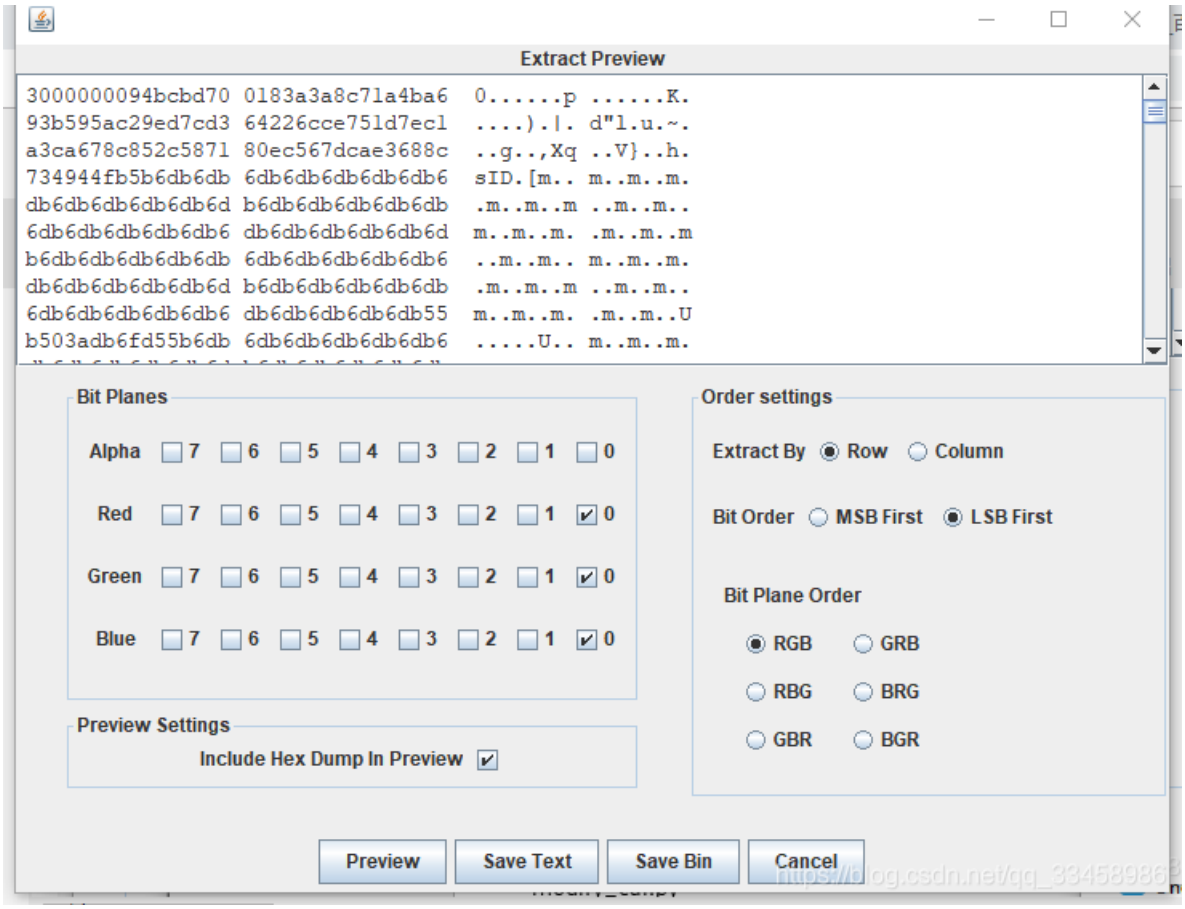





## LSB隐写-cloacked-pixel工具

使用该算法得到的结果如下






可以看到这个算法的特点呢，是开头有个数字，根据观察，这个数字应该是有用的数据的长度，然后才是很多个0，接着是相应长度的数据，上面这两张截图中下面这张图隐写的信息比较少，同时，该算法需要密码。

**IHDR隐写**

一般就是修改图片高度，IHDR：crc error

```
import os
import binascii
import struct

img = open("cai.png", "rb").read()

for w in range(1024):
    for h in range(1024):
        data = img[0xc:0x10] + struct.pack('>i',w) + struct.pack('>i',h) + img[0x18:0x1d]
        crc32 = binascii.crc32(data) & 0xffffffff
        if crc32 == struct.unpack('>i',img[0x1d:0x21])[0] & 0xffffffff:
            print w, h
            print hex(w), hex(h)
            open("cai.png", "wb").write(img[:0xc] + data + img[0x1d:])
            exit()
```

参考链接：ctf-stego汇总