攻防世界pure_color

原创

String vs Array 已于 2022-03-03 20:24:40 修改 174 收藏

文章标签： python opencv

于 2022-03-03 20:14:28 首次发布

本文链接：https://blog.csdn.net/SStringss/article/details/123262733

WriteUp全是工具没得情感看的时候我都惊呆了，37度的手指竟然能敲出这么冰冷的文字

首先看到这种图片隐写的时候就要考虑是否存在分离文件的可能，并且图片隐写可能采用的空域LSB, Diamond，一些原理上是LSB的变体， 变换域DCT, DWT 以及一些可能我们脑回路想不到甚至解不出来的自适应变换等等

在本题中首先用2进制工具查看了一遍，1.没有附加文件 2. CRC与宽高匹配说明没有对宽高做手脚 3.查找没有flag匹配字符串

那么根据难易程度排行第四点我们就要找LSB了。

```
>>>import cv2
>>>n = cv2.imread('./stego.png')    #这里我改了照片的名字
>>>print(n)
>>>[[[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [255 255 255]
  [255 255 255]
  [255 255 255]]

 [[255 255 255]
  [255 255 255]
  [255 255 255]
  ...
  [255 255 255]
  [255 255 255]
  [255 255 255]]]
```

打印出来乍一看没什么，好像都是255结尾的，我们继续往下翻

```
>>>count = 0
>>>for k in n :
    for p in k:
        if p[0] != 255:
            print('b:' + str(p[0]))
               count += 1
        elif p[1] != 255:
            print('g:' + str(p[1]))
               count += 1
        elif p[2] != 255:
            print('r' + str(p[2]))
               count += 1

b:254
b:253
b:253
b:252
b:253
b:253
b:253
...
b:254
>>>count
22150
```

这个时候就可以确定了以RGB读的png在B通道进行了隐写的操作。问题又来了，他隐写的是什么数据，是文本，是图片，Or Others?

我们来看一下他隐写变换了多少个像素吧：根据count的计算，一共更改了22150位。折合成字符（Byte, 暂不考虑Word 和 2Word)，一共有2768Byte的字节流。远远超出flag的长度，所以我们PASS 字符的长度，考虑图像的可能。

因为R,G通道本身不涉及LSB，我们直接将他们抛弃。剩一个通道的图片将转化为灰度图，像素值就是B通道的值。为了扩大这四种（B的值在252-255之间）像素点肉眼的区分度，我们采用如下的方法：

```
>>> k = n[:,:,0]
>>> for i in range(279):
    for p in range(1024):
        k[i][p] = (k[i][p] - 252) * 85
>>> cv2.imwrite('flag.png',k)                    #保存图像
True
```

查看我们得到的图像

没想到Flag就出来了，不然就要尝试从字符或者其他图片方法入手了