

CTF-隐写术-心中无码

原创

湖zz 于 2019-07-24 16:40:13 发布 1613 收藏 3

文章标签: [CTF 隐写术](#)

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/zhz444614971/article/details/97137988>

版权

1.一开始使用foremost没用,winhex也看不出什么,steghide也没东西

最后用Stegsolve查看图片发现blue 0,green 0,red 0挺奇怪的

再加上题目为心中无码,而且(255,255,0)为黄色

于是设想用程序读取这张图上的点,去除黄色,然后取blue位使之变为0和1,再根据0和1对应到黑和白进行图片重构

代码:

```
from PIL import Image

lena = Image.open('Lena.png')
pic = Image.new("RGB", (300, 300))
width, height = lena.size

b0 = ""
bnum = 0

for x in range(width):
    for y in range(height):
        point = lena.getpixel((x, y))
        if (point != (255, 255, 0)): # 去除黄颜色的点
            if (int(point[2]) & int('0x01', 16)):
                bnum += 1
                b0 += "0"
            else:
                bnum += 1
                b0 += "1"

#width:384 height:698 因为去除了黄颜色的点所以最后点的个数为90000
nums = 0
```

```

for i in range(300) :
    for j in range(300) :
        if b0[nums] == "0" :
            pic.putpixel([i,j],(0,0,0))
        else :
            pic.putpixel([i,j],(255,255,255))
        nums += 1
pic.save("rs.png")

```

结果:得到一张二维码的图片

2.对二维码进行扫描

结果:

```

+++++++[->+++++++<]>+++++++.<+++++[->-----<]>--.<+++++[->+++++<]>+++++.<+++++[->
>+++++<]>+-.-----.<+++++++[->-----<]>--.<+++++++[->+++++++<]>+++++.<+++++[->---<]>--.<+++++[->
>---<]>--.<+++++[->+++<]>+-.<+++++++[->-----<]>--.<+++++[->+++++<]>+++++.<+++++[->-----<]>--.<+++++[->
>+++<]>+-.<+++++[->---<]>--.<+++++[->+++++<]>+++++.<+++++[->---<]>--.<+++++++[->-----<]>--.<+++++[->
>+++++<]>+++++.<+++++++[->-----<]>--.<+++++[->+++++<]>+++++.<+++++[->-----<]>--.<+++++[->
>++++<]>++++.<

```

3.发现这是brainfuck的代码,进行解析

brainfuk的解释:

- + : 指针指向的单元的值加1
- : 指针指向的单元的值减1
- > : 将指针移动到下一个单元(右边的元素)
- < : 将指针移动到上一个单元(左边的元素)
- . : 打印当前单元的内容的ASCII值 (比如 65 = 'A').
- , : 读取一个字符到当前的单元
- [: 如果当前单元的值是0, 则向后调到对应的]处
-] : 如果当前单元的值不是0, 则向前跳转到对应的[处

例子: ++++++++[->+++++++<]>+++++++.

过程:[8,9]

[7,18]

[6,27]

[5,36]

[4,45]

[3,54]

[2,63]

[1,72]

[0,81]

[0,89]

最后输出:89

结果:Y3Rme2xlbmFfMXNfy3V0M30=

4.可以看出这是base64加密

解密后结果:c{f{l}e{n}a_1s_cut3}