xctf攻防世界 MISC高手进阶区 很普通的数独

原创

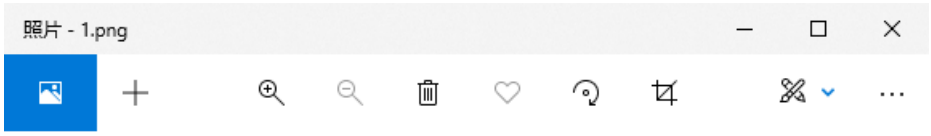
l8947943 于 2022-01-16 21:22:09 发布 5904 收藏

分类专栏： 攻防世界misc之路 文章标签： python misc 安全

本文链接：https://blog.csdn.net/l8947943/article/details/122528834

1. 进入环境，下载附件

给的一个压缩包，里面一堆图片



3		2	7	5	6	4	1	8
		8						6
		7		4	1	2		3
5		6		3	9	7		2
1		9		7	8	6		5
		4						1
2		3	5	8	4	1	6	9
9								
			9		7	3	5	4

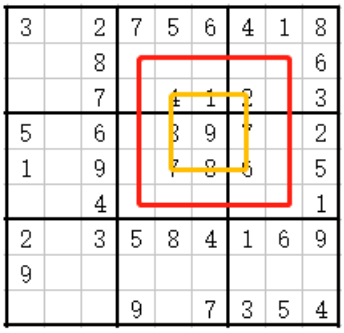
一共25张，没啥其他的提示信息

2. 问题分析

1. **猜测为二维码**

   为什么猜测是二维码？

- 图片个数是25 = 5 * 5的大小
- 数字存在的意义不是为了让你计算，应该表示占位符
- 图一有二维码定位图



是不是很像！题目分析很容易，方向其实也蛮明显的，但是用最笨的方法，就是1-25图一个个数数字，最终整理，得到45张二进制：

```
111111101010101000101000001111110000101111111
100000101100111101010011101100011001001000001
101110101110011111010011111101000101001011101
101110101101100010001010000011110001101011101
101110100011100100001111101111111011101011101
100000101100100000011000100001110100001000001
111111101010101010101010101010101011101111111
000000000011001101001000110100110011100000000
110011100100100001111111100100101000000101111
101001001011111111101110101011110101101001100
100000111100100100000110001101001101010001010
001100010011010001010011000100000010110010000
010110101010001111110100011101001110101101111
100011000100011100111011101101100101101110001
001100110100000000010010000111100101101011010
101000001011010111100110111111010011101000011
110111110111011001101100010100001110000100000
110101000010101000011101101101110101101001100
010011111110001011111010001000011011101101100
011001011001010101100011110101001100001010010
010111111111101011111111101101101111111111100
011110001100000100001000101000100100100011110
111110101110011100111010110100110100101010010
110010001011101011101000111100000011100010000
101011111011100111011111111000010101111110010
110100011000111000100111101101111101000100010
111101111110001001000011010110001111110111110
011001010101000110010100010001000101101010001
011101110101101101100100001101101000111101001
110110001001101100010101101111111010010110011 0
000011100111000000000100001010101111100010010
111010010011110011101110010100001011111010010
101001100010111111110100000100001010101010100
000010011001001101110101001111100101111101101
000010111101110001101011000001000101110100110
011110011010100010100000011011000001110010000
100110100100001101111111110110010111011111 0011
000000001111110101101000101011100100100011010
111111100011111011011010101101110011101011110
100000101110101101101000111110010001100010001
101110101011100001111111101101001000111111011
101110100110111101101000010011011000111011101
101110100000011101100001101010110010010010001
100000101011001011111011001011000011010110000
111111101010101001111011110101101110000101101
```

2. **绘制二维码**

   此段代码将数据一一填充到45 * 45大小的画布上，参考链接： https://www.freesion.com/article/25531434454/

```
# -*- coding:utf-8 -*-
from PIL import Image
x = 45
y = 45

im = Image.new("RGB", (x, y))  # 创建图片
file = open('1.txt', 'r')  # 打开rbg值文件
for i in range(0, x):
    line = file.readline()  # 获取一行
    for j in range(0, y):
        if line[j] == '0':
            im.putpixel((i, j), (255, 255, 255))  # rgb转化为像素
        else:
            im.putpixel((i, j), (0, 0, 0))  # rgb转化为像素
im.show()
```

3. **换一种做法！直接代码处理**

   一步步重新写了一遍，算是过下来了，大致能看懂，也明白了做法：

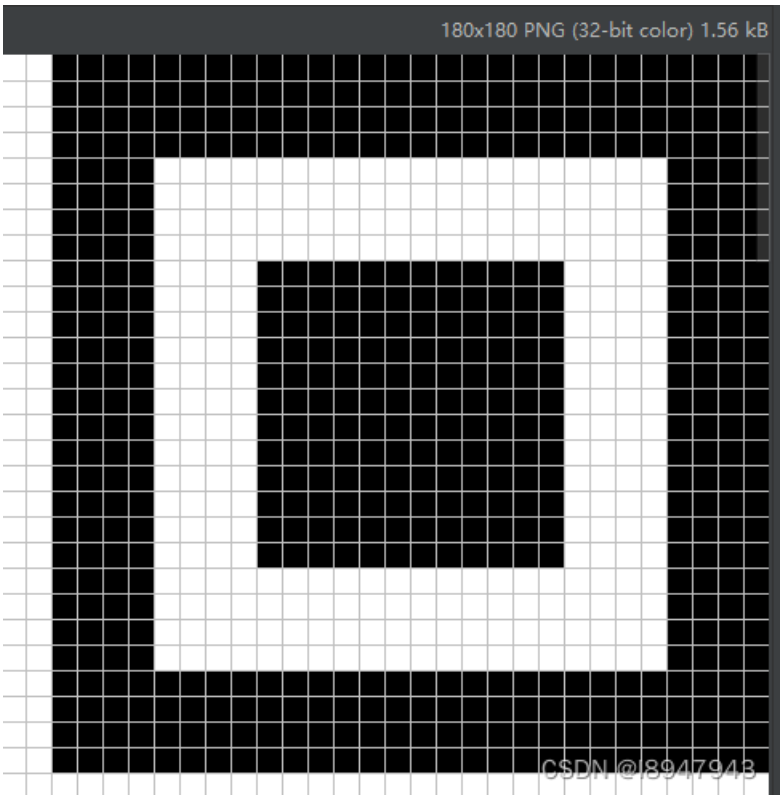
```
from PIL import Image

img0 = Image.new("RGBA", (180, 180), "white")

# 定义一个框选坐标，既能避开边框，又可以框柱数字
box0 = (3, 3, 199, 199)

for i in range(45):
    for j in range(45):
        # 得到绘制图片的编号,并打开图片
        png_num = i // 9 * 5 + j // 9 + 1
        img_temp = Image.open('pic/' + str(png_num) + '.png')
        # box定义一个可以框选出每张图片小格子的区域,定义大小是7 * 7的大小（为什么，我觉得不需要多精确，只要小框中有一个黑色像素，则认为是黑图）
        box = ((j % 9) * 22 + 11, (i % 9) * 22 + 11, (j % 9) * 22 + 18, (i % 9) * 22 + 18)
        # 先对每张图进行裁剪，避开大图的边线，其次裁剪，得到每张图的小框，为是否有数字判断做准备
        img_recognize = img_temp.crop(box0).crop(box)

        flag = False
        for pix_x in range(7):
            for pix_y in range(7):
                # 如果该像素位点包含数字，则说明在二维码上绘制出来的是黑色
                if img_recognize.getpixel((pix_x, pix_y)) != (255, 255, 255, 255):
                    flag = True
                    break
            if flag:
                break
        if flag:
            # 这儿的4是什么意思？其实就是为了画出的结果中，
            #以个黑色用4*4大小表示一个包含字符区域的块，如下图
            for x in range(4):
                for y in range(4):
                    img0.putpixel([i * 4 + x, j * 4 + y], (0, 0, 0))
img0.save("result.png")
```



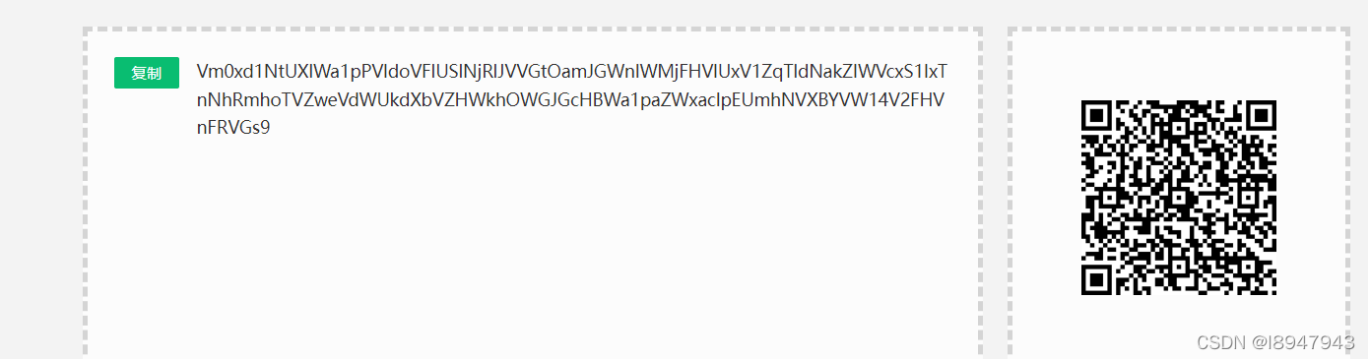
运行代码后发现二维码定位块不对，如图：

图片1，5，21定位符有问题，回到原图，修改三者之间的名称 1->5, 5->21, 21->1后重新运行，得到二维码：

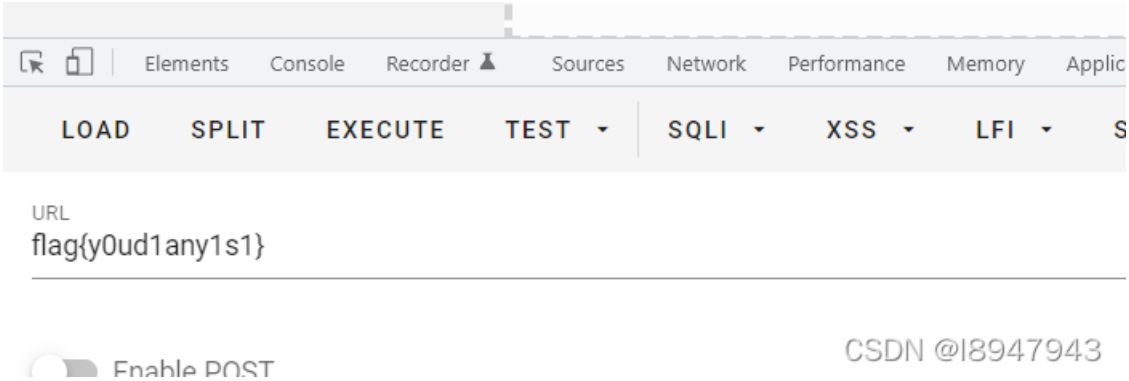
3. **在线扫描二维码**

https://jiema.wwei.cn/

得到扫码结果：



很像base64编码，使用hackbar进行解码，特么的就离谱，还要解码7次才有结果。。。如图：



得到最终答案：`flag{y0ud1any1s1}`

3. 总结

- misc方向的思路奇特，需要脑洞，但是代码非常重要
- 参考其他博主的wp，跑完代码发现bug，自己尝试做了修改
- 多次解码是我没想到的，就离谱到家了
- 代码中的裁剪参数真的很不好理解，做了可视化尝试才知道了为什么那样做，希望原博不是用尺子量出来的

这个题代码研究的时间有点长，心累。欢迎交流哈~~~