BUUCTF misc 喵喵喵

原创

Warning 于 2019-10-03 23:44:53 发布 3705 收藏 8
分类专栏： python 杂项 工具 文章标签： BUUCTF NTFSstreamseditor

本文链接：https://blog.csdn.net/destiny1507/article/details/101997730

我觉得……够我消化一下的了。涉及到的知识点：

- LSB隐写
- 二维码补全（修改图片高度）
- NTFS交换数据流隐写
- 反编译
- python简单解密（python使用）

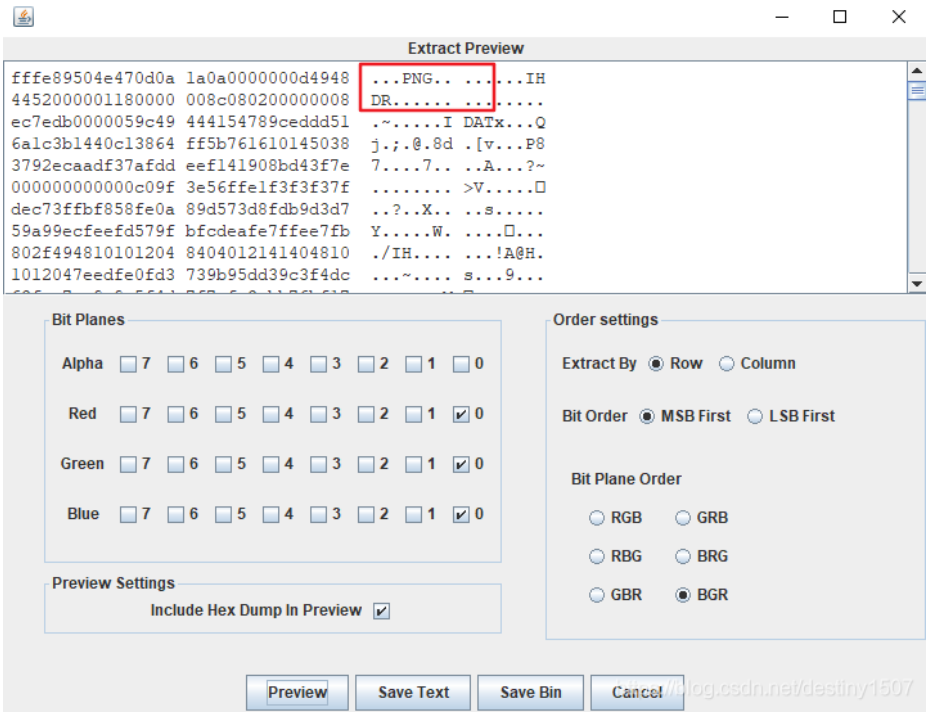
吐血而亡……下面是正经的题解：

首先给了一张图片：

看看，是不是一只可可爱爱的小猫咪？mmp

用stegsolve打开，发现最低位通道有点问题：一串可疑的白

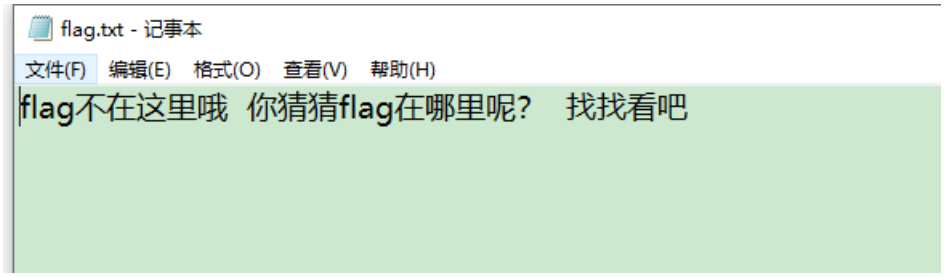
分析后发现是一个PNG图片：



导出后保存发现是半张二维码，2333终于知道题目中的扫一扫是什么意思了，修改一下高度，得到大小正常的二维码：

发现二维码的颜色不太对，黑色和白色位置反了。用stegsolve进行颜色对换后得到一张正常的二维码，扫描后下载百度网盘的文件：flag.rar

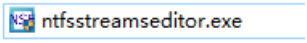
呼~是不是很感动，这解压之后应该就得到结果了吧，于是我兴致冲冲地解压了，满怀欣喜地觉得齐活了，but！



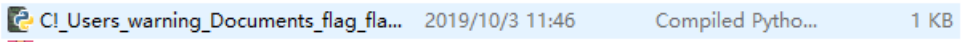
这里就要说到一个坑了，这道题的原题是在安恒月赛里的，当时的题目有三个hint，其中一个就是NTFS，但是BUUCTF里面没有提到……卑微-ing，我就看了题解。

然后发现有大佬说这个得到的flag.rar如果用winrar解压就可以报出错误，但我是用bandzip……解压过程顺畅无比……

使用工具



它可以自动识别出你的文件中NTFS隐藏的数据流，并可以直接导出，于是我发现了一个.pyc文件：



丢到在线网站上反编译 python反编译得到一段用来加密的脚本，脚本比较简单，直接写出解密的运行即可得到flag：

反编译出的加密脚本：

```
import base64

def encode():
    flag = '*************'
    ciphertext = []
    for i in range(len(flag)):
        s = chr(i ^ ord(flag[i]))
        if i % 2 == 0:
            s = ord(s) + 10
        else:
            s = ord(s) - 10
        ciphertext.append(str(s))

    return ciphertext[::-1]  #倒序一遍

ciphertext = [
    '96',
    '65',
    '93',
    '123',
    '91',
    '97',
    '22',
    '93',
    '70',
    '102',
    '94',
    '132',
    '46',
    '112',
    '64',
    '97',
    '88',
    '80',
    '82',
    '137',
    '90',
    '109',
    '99',
    '112']
```

解密脚本：

```
import base64

ciphertext = ['96','65','93','123','91','97','22', '93','70','102','94','132','46','112','64','97','88','80
ciphertext = ciphertext[::-1]

def decode():
    code = ''
    for i in range(24):
        if(i%2 == 0):
            a = int(ciphertext[i]) - 10
        else:
            a = int(ciphertext[i]) + 10
        a = i ^ a
        code = code + chr(a)
    print(code)

decode()
```

跑完得到flag：