[BUGKU] [MISC] 猫片

原创

三无提督w 于 2020-07-10 00:02:36 发布 96 收藏

~~这题是bugku从安恒那偷的，就是那个只有签到题的比赛~~

提示：难度与BUGKU同分数的题目相比较高，完全不是一个水平，如果是新手建议别看writeup硬啃，先做其他题吧。大佬随意

下载，发现一个文件

`binwalk`走一下，发现真的就是png，改后缀，发现图片是只猫

因为是没人解出来的杂项题，所以比赛才会放出hints，有了hints后题目难度大大下降了，起码有了明确的方向

拉进`stegsolve`

左右切一切色道，发现Blue plane 0，red plane 0和green plane 0头上都有一条，怀疑是隐写

选 analyse > Data Extract

左侧勾上Blue plane 0，red plane 0和green plane 0，右侧根据提示勾上LSB和BGR

preview一下，看到数据头，那么明显的png，保存bin，名字直接写成11.png

但是png开头是89504E47 ，这里要把多余的删去，拉进`winhex`，删去头上的fffe

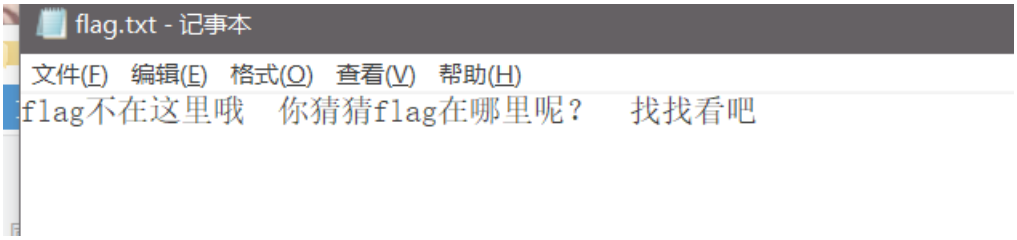
之后就能看到png了，是半张二维码

拉进linux，crc校检报错，去改宽高，二维码是方的，所以把高写成和宽一样

之后扫描一下

> https://pan.baidu.com/s/1pLT2J4f

下载下来一个压缩包，打开flag



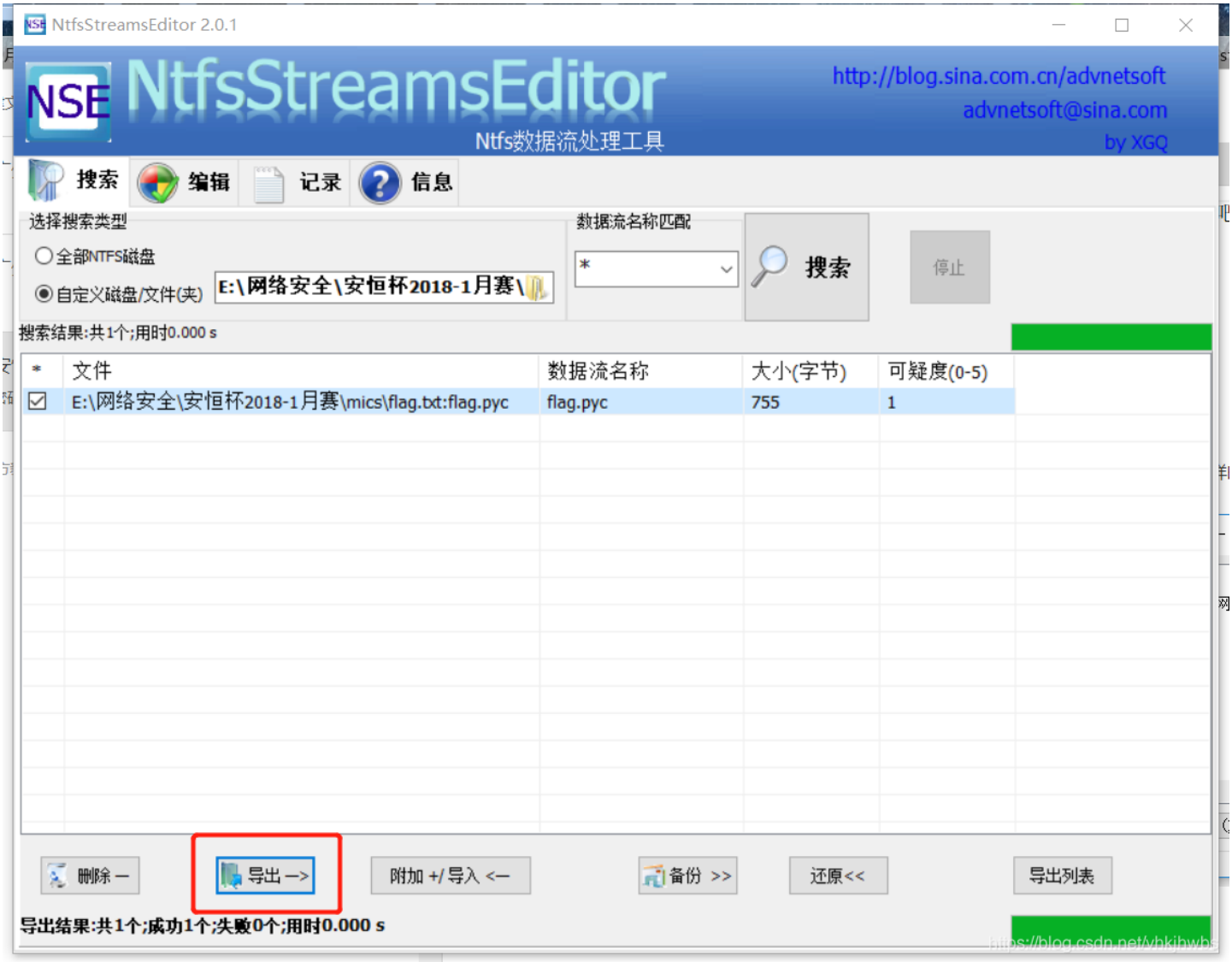
心态炸了

看看提示，还有一个NTFS没用上

搜了一下NTFS隐写，发现了NTFS交换数据流的隐写方式

打开`NtfsStreamsEditor`

选择文件夹进行搜索，记得把文本文件解压出来



导出，出现了pyc文件

在网上摘抄了一段py和pyc的文章

> 原来Python的程序中，是把原始程序代码放在.py文件里，而Python会在执行.py文件的时候。将.py形式的程序编译成中间式文件（byte-compiled）的.pyc文件，这么做的目的就是为了加快下次执行文件的速度。
> 所以，在我们运行python文件的时候，就会自动首先查看是否具有.pyc文件，如果有的话，而且.py文件的修改时间和.pyc的修改时间一样，就会读取.pyc文件，否则，Python就会读原来的.py文件。
> 其实并不是所有的.py文件在与运行的时候都会产生.pyc文件，只有在import相应的.py文件的时候，才会生成相应的.pyc文件

接下来回到题目，可以用在线工具解密pyc

```
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]


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']
```

这就变成了普通的算法题了。

写个解密脚本

```
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']
flag = ''
ciphertext.reverse()

for i in range(len(ciphertext)):
    if i % 2 == 0:
        s = int(ciphertext[i]) - 10
    else:
        s = int(ciphertext[i]) + 10
    s = chr(i ^ s)
    flag += s

print(flag)
```

逻辑大概是这样，很简单就不赘述了

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-HjBF8dl5-1594310513220)(https://s1.ax1x.com/2020/06/30/NlVGWT.png)]

---

flag{Y@e_Cl3veR_C1Ever!}