VNCTF2022_Misc_复现

原创

M3ng@L 于 2022-02-24 20:07:09 发布

分类专栏： CTF比赛复现 文章标签： python Misc

本文链接：https://blog.csdn.net/qq_51999772/article/details/123119743

VNCTF2022_Misc_复现

仔细找找

放大图片可以看见有很小的与周围颜色不同的像素点均匀的分布在图片中

先寻找像素点之间的间隔，然后拼接在一起生成新的图片

代码实现

```
from re import L
from PIL import Image
import numpy

pic = Image.open("C:\\Users\\Menglin\\Desktop\\flag.png")
# 寻找像素间隔
# array = numpy.array(pic)
# print(array.shape[0])
# print(array.shape[1])
# for i in range(array.shape[1]):
#     for j in range(16,array.shape[0]):
#         if pic.getpixel((i,j)) != (0,0,0):
#             print(i,j)
#             break
widith, height = [], []
for i in range(pic.size[0]):
    if pic.getpixel((i,46)) == (255,255,255):
        widith.append(i)
for i in range(pic.size[1]):
    if pic.getpixel((24,i)) == (255,255,255):
        height.append(i)
image = Image.new("RGB",(len(widith),len(height)),(255,255,255))
for i in range(len(widith)):
    for j in range(len(height)):
        image.putpixel((i,j),pic.getpixel((widith[i],height[j])))
image.show()
```

Prize wheel

通过抽奖小程序获得flag.zip的解压密码（不断输1就可，PS：获奖概率几乎为0的话就反编译）

得到一张混沌的图片，可以看出是以正方形的形式（**这里的正方形是以像素值为单位的**）被打乱的像素原本应该在的地方

那么就按照正方形的路径遍历每个像素点，将每个像素点进行一步一步的移位，直到

> hint：图片上存在可以定位的东西

直到定位的东西被放回了原位

可以看到图片上有不同于其他地方的白点，而且每个正方形的路径上只有一个白点

可以想到的利用白点来定位的方式就是把白点排成一条线（因为也没有其他提示了）

解析官方wp解题代码

我们已经知道图片需要通过正方形的路径来遍历像素值，再进行调整使其归位

那么需要规定遍历的每个正方形的起点

```
for count in tqdm(range(3,width + 1,2)): # count是所遍历正方形一边的长度，比如3、5、7...
    d = count // 2
    for i in range((count - 1) * 4): # 所遍历正方形的周长
        p_x = centre_x - d
        p_y = centre_y - d
```

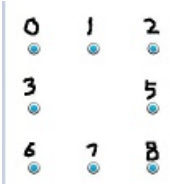
其中`p_x,p_y`就是遍历的每个正方形的起点（也就是每个正方形的左下角），且这里是从图片中心的小正方形开始进行遍历调整像素值

判定是否正确归位的条件，注意这里用的是RGBA作为像素点的属性（可以通过getpiexl函数查看）

```
if img.getpixel((width // 2,centre_y - count // 2 )) == (255,255,255,255):
 break
```

当在图片在以宽度的中点，以高度的中点以上是白点（最后也就连成了一条白线）时，切换到下一个正方形

进行相邻像素值的转换，这里以边长为像素值3的正方形为例，以`逆时针`的方向来进行遍历



首先从6处开始（备份6处的像素值），将7处的像素值移到6处，8处的像素值移到7处（前三个for循环有`length-1`个动作所以`range(count-1)`），至此一个for循环就结束了；继续下一个for循环，将5处的像素值移到8处，2处的像素值移到5处，结束这个for循环；以此类推，到最后一个for循环，将3处的像素值移到0处（因为只需要这一个动作就可以达到效果，所以`range(count-2)`），结束for循环；注意到3处此时的像素值没有改变，应该是6处的像素值，那么就用之前备份的6处的像素值赋值给3处即可（注意`count==length==正方形边长`）

```
temp = img.getpixel((p_x,p_y))
for j in range(count - 1):
    img.putpixel((p_x,p_y),(img.getpixel((p_x + 1,p_y))))
    # 将下一个像素点的像素值赋值给当前像素点
    p_x += 1
    # 切换目标像素点
for j in range(count - 1):
    img.putpixel((p_x,p_y),(img.getpixel((p_x,p_y + 1))))
    p_y += 1
for j in range(count - 1):
    img.putpixel((p_x,p_y),(img.getpixel((p_x - 1,p_y))))
    p_x -= 1
    # 每个地方p_x,p_y进行的操作不同的原因是通过逆时针进行遍历正方形
for j in range(count - 2):
    img.putpixel((p_x,p_y),(img.getpixel((p_x,p_y - 1))))
    p_y -= 1
img.putpixel((p_x,p_y),temp)
```

完整官方wp解题代码

```python
from PIL import Image
from tqdm import tqdm

img = Image.open("flag.png")
width, height = img.size

centre_x = width // 2
centre_y = height // 2

for count in tqdm(range(3,width + 1,2)):
    d = count // 2
    for i in range((count - 1) * 4):
        p_x = centre_x - d
        p_y = centre_y - d
        if img.getpixel((width // 2,centre_y - count // 2 )) == (255,255,255,255):
            break
        temp = img.getpixel((p_x,p_y))
        for j in range(count - 1):
            img.putpixel((p_x,p_y),(img.getpixel((p_x + 1,p_y))))
            p_x += 1
        for j in range(count - 1):
            img.putpixel((p_x,p_y),(img.getpixel((p_x,p_y + 1))))
            p_y += 1
        for j in range(count - 1):
            img.putpixel((p_x,p_y),(img.getpixel((p_x - 1,p_y))))
            p_x -= 1
        for j in range(count - 2):
            img.putpixel((p_x,p_y),(img.getpixel((p_x,p_y - 1))))
            p_y -= 1
        img.putpixel((p_x,p_y),temp)
img.show()
```

strage flag

流量包是蚁剑流量，提取流量中所包含的文件数据

流量包导出第8个tcp流的原始数据，进行Gunzip转换（对原始数据的操作cyberchief的magic可以一步到位）

得到一组以`tree`形式展现的文件夹；

```
`-- New\ folder
    |-- New\ folder
    |   |-- New\ folder
    |   |-- New\ folder\ (2)
    |   |-- New\ folder\ (3)
    |   `-- New\ folder\ (4)
    `-- New\ folder\ (2)
        |-- New\ Folder\ (3)
        |   |-- New\ folder
        |   |   |-- New\ folder
        |   |   |   |-- New\ folder
        |   |   |   |-- New\ folder(2)
        |   |   |   |   `-- New\ folder
        |   |   |   |-- New\ folder(3)
        |   |   |   |   `-- New\ folder
        |   |   |   `-- New\ folder(4)
        |   |   |       `-- New\ folder
        |   |   `-- New\ folder(2)
```


这是folders语言（一种esolang），相关介绍Folders - Esolang (esolangs.org)

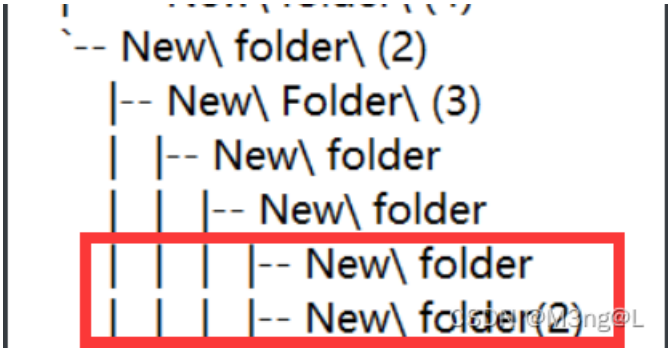
> C#编译：GitHub - rottytooth/Folders: A language where the code is written with folders
> 或者直接安装folders：`pip install Folders`

- 根据文件夹的相对位置来创建文件夹

详情看官方wp（我没试过）[VNCTF 2022 Official WriteUp.pdf (tonycrane.cc)]
(https://note.tonycrane.cc/assets/images/writeups/vnctf2022/VNCTF 2022 Official WriteUp.pdf)

- 根据folders语言的原理将文件夹直接转换为flag字符

比如：文件夹甲内有文件夹乙，则甲转换为1，乙则不作替换（不管）；文件夹丙内无新的文件夹，则丙转换为0；注意：并不是所有文件夹都可以进行转换，在这里只有倒数第二层的文件夹可以转换，原因是其他层次的文件夹是folders语言的命令语句



只有红框里的文件夹可以转换

转换脚本

```
from Crypto.Util.number import *

flag = ""
N_data = []
with open("download.txt") as f:
    for line in f.readlines():
        temp = str(line).index("N")
        N_data.append(temp)
        # print(line)
print(N_data)
for i in range(len(N_data)-1):
    if N_data[i] == 24:
        if N_data[i] >= N_data[i + 1]:
            flag += "0"
        else:
            flag += "1"
# print(flag)
flag = int(flag,2)
print(long_to_bytes(flag))
```

得到的是

> v903879df57n503879bcdfc1efc141fe}tf{d23

很显然是错位了的，个人估计可能是因为folders语言的嵌套语句之内的原因；总而言之，得到的字符确实是flag出现的字符，大体的顺序也是对的，但是需要重新手动排位（尤其是n,c两个字符）

> vnctf{d23903879df57503879bcdf1efc141fe}

simple macos

在压缩包里面寻找有用的文件

> simple macos.7z\Users\scr1pt\Librarys\Mail\V9\AC26459E-8824-4F93-8FF1-DC6AB35E8B0D[Gmail].mbox\已删除邮件.mbox\EF4FC717-2856-44B2-B23B-303D44FDC243\Data\Messages\603.emlx

其中有提示（base64转码得到）

> i hide the secret flag in the profile picture
> please clean your computer after reading , be careful !!!

重点在某张图片里面

继续寻找其他可疑文件

> simple macos.7z\System\Volumes\Preboot\79FABCCE-3636-4266-A6CF-8E3BB40332B4\var\db\CryptoUserInfo.plist

有一段文字以及base64编码

> our secret need a password

其中base64转码得到一张图像的数据

010editor打开图片文件末尾有flag后半段

由提示：oursecret进行解密（注意要把图像文件中的flag后半段删除，不然oursecert提示没有隐藏数据）

> oursecret下载：文件加密软件-文档加密工具(Our Secret)v2.5.5.0 绿色版-东坡下载 (uzzf.com)

解密得到flag前半段

base64转码得到一张图像的数据

010editor打开图片文件末尾有flag后半段

由提示：oursecret进行解密（注意要把图像文件中的flag后半段删除，不然oursecert提示没有隐藏数据）

> oursecret下载：文件加密软件-文档加密工具(Our Secret)v2.5.5.0 绿色版-东坡下载 (uzzf.com)

解密得到flag前半段