VNCTF2022 WriteUp

原创

是Mumuzi 于 2022-02-13 20:00:00 发布 1254 收藏 5

分类专栏： ctf buuctf 文章标签： 信息安全

文章目录

Crypto
ezmath
Web
GameV4.0
Reverse
BabyMaze
Misc
问卷
仔细找找
Strange flag
simple macos
prize wheel

Crypto

ezmath

限制了60s 我只能说我这垃圾电脑边玩MC边跑根本跑不完777次我草

如果2^N-1能被15整除，那么自然数N应取那些值 百度搜(2^n-1) % 15就有

```python
from pwn import *
context.log_level='debug'
import re
def sha256(enc,sec):
    table = string.ascii_letters+string.digits
    for i in table:
        for j in table:
            for k in table:
                for n in table:
                    s = i+j+k+n
                    s1 = s+sec
                    m = hashlib.sha256(s1.encode())
                    mi = m.hexdigest()
                    if(enc == mi):
                        return s

p = remote('node4.buuoj.cn',28865)
rec = p.recv()
sec = rec[16:32].decode()
sha = rec[37:-23].decode()
print(sec,sha)
result = sha256(sha,sec)
print(result)
p.sendline(result.encode())
i = 0
while 1:
    r = p.recv().decode()
    que = re.search('plz give me the (.*)\(n\)',r).group(1)[:-3]
    ans = str(int(que)*4).encode()
    print(i)
    p.sendline(ans)
    i += 1
    if(i == 777):
        p.recv()
        p.recv()
        p.recv()
    p.recvuntil('are so smart.\n')
```

Web

GameV4.0

找到/js/data.js。拉到最下面即可看到base64编码后的flag

```
VNCTF{Welcome_to_VNCTF2022}
```

Reverse

BabyMaze

直接打印出字节码，用老方法

```python
import dis, marshal, sys

header_sizes = [
    # (size, first version this applies to)
    # pyc files were introduced in 0.9.2 way, way back in June 1991.
    (8,  (0, 9, 2)),  # 2 bytes magic number, \r\n, 4 bytes UNIX timestamp
    (12, (3, 6)),     # added 4 bytes file size
    # bytes 4-8 are flags, meaning of 9-16 depends on what flags are set
    # bit 0 not set: 9-12 timestamp, 13-16 file size
    # bit 0 set: 9-16 file hash (SipHash-2-4, k0 = 4 bytes of the file, k1 = 0)
    (16, (3, 7)),     # inserted 4 bytes bit flag field at 4-8
    # future version may add more bytes still, at which point we can extend
    # this table. It is correct for Python versions up to 3.9
]
header_size = next(s for s, v in reversed(header_sizes) if sys.version_info >= v)

with open('BabyMaze.pyc', "rb") as f:
    metadata = f.read(header_size)  # first header_size bytes are metadata
    code = marshal.load(f)          # rest is a marshalled code object

dis.dis(code)
```

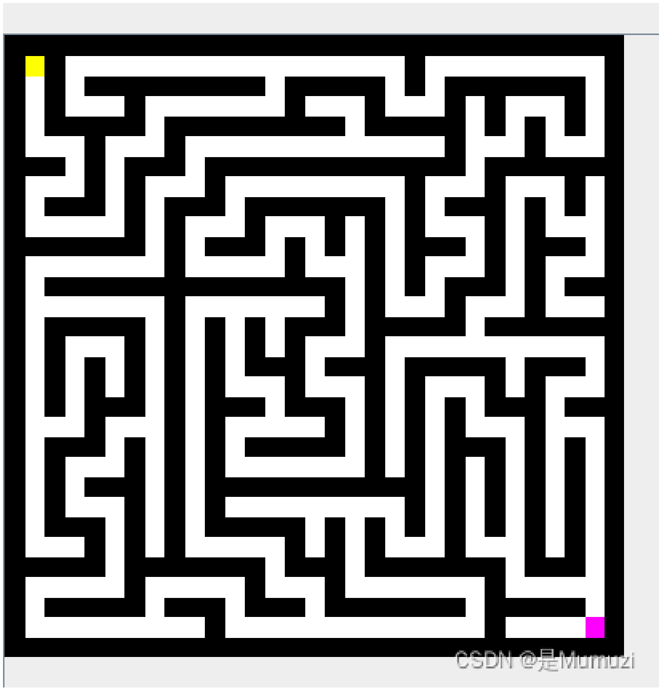
```
C:\Users\mumuzi\PycharmProjects\pythonProject\venv\Scripts\python3.exe E:/火狐下载/az.py
  1           0 JUMP_ABSOLUTE            4
        >>    2 JUMP_ABSOLUTE            6
        >>    4 JUMP_ABSOLUTE            2
        >>    6 LOAD_CONST               0 (1)
              8 LOAD_CONST               0 (1)
             10 LOAD_CONST               0 (1)
             12 LOAD_CONST               0 (1)
             14 LOAD_CONST               0 (1)
             16 LOAD_CONST               0 (1)
             18 LOAD_CONST               0 (1)
             20 LOAD_CONST               0 (1)
             22 LOAD_CONST               0 (1)
             24 LOAD_CONST               0 (1)
             26 LOAD_CONST               0 (1)
             28 LOAD_CONST               0 (1)
             30 LOAD_CONST               0 (1)
             32 LOAD_CONST               0 (1)
             34 LOAD_CONST               0 (1)
             36 LOAD_CONST               0 (1)
             38 LOAD_CONST               0 (1)
             40 LOAD_CONST               0 (1)
             42 LOAD_CONST               0 (1)
             44 LOAD_CONST               0 (1)
             46 LOAD_CONST               0 (1)
             48 LOAD_CONST               0 (1)
             50 LOAD_CONST               0 (1)
             52 LOAD_CONST               0 (1)
             54 LOAD_CONST               0 (1)
             56 LOAD_CONST               0 (1)
             58 LOAD_CONST               0 (1)
```


可以看到得到了很多0,2和1个1和1个3,。

而且很明显是31*31

处理一下，画个图

```
s = '0000000000000000000000000000000010222222222222222220222222222002020000000002000002020000000200202220222222222
0222022220202222020020000020000000002000002020202002220222022222222222222022202220000202000200000000000000200000000 2
2202022202222222222022202220200200020200020000000202000202020022222020222022220202022202022200000000020200020202020
0020200020022222220222220222020222020222002000000000000000202020002020000222222202222222202022202220222002000000 20
2020200020000020000000020222020202020222022222222222222002020202020002000200000200020020202020202220222022220202
2200202000202000200020202020002000022202220202222202020202220222002000202020200000202020002020200202220202022222
2202020202020200202000202000000000000202020202002022202020222222222020202020200200020202000002020202020202020200222
0202022222020202220222020200000020000000200020000000200020022222202222220222022222220222220020000020002000200000 00
200000200222222222022222222222222202222230000000000000000000000000000000'
from PIL import Image
pic = Image.new('RGB',(31,31),(255,255,255))
for i in range(31):
    for j in range(31):
        if(s[i*31+j] == '0'):
            pic.putpixel((j,i),(0,0,0))
        elif(s[i*31+j] == '1'):
            pic.putpixel((j,i),(255,255,0))
        elif(s[i*31+j] == '3'):
            pic.putpixel((j,i),(255,0,255))
pic.show()
pic.save('flag.png')
```



在后面还能看到如果在29,29就输出flag，说明起点是1,1。

手动走一下

```
ssssddssaassddddwwwwddwwddddddwwddddddssddwwddddddddssssaawwaassaassaassddssaassaawwwwwwaaaaaaaaassaassddddwwddss
ddssssaassddssssaaaaaawwddwwaawwwwaassssssssssssddddssddssdddddddddwwaaaaaawwwwddssddwwwwwwwwddssddsssssssssddddss
```

md5一下就是flag(重新走的，flag不确定是不是这个了)

```
VNCTF{801f190737434100e7d2790bd5b0732e}
```

Misc

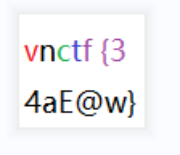
问卷

```
VNCTF{See_You_Next_Time!!!}
```

仔细找找

直接脚本近邻发现间隔不相同，想用蓝帽半决的爆破脚本发现找不到，PS也没成功弄出来，最后本来想着一个个手撸，然后发现其实第24列和15排，就只有黑色和白色，那只提取白色，不就有了横坐标和纵坐标的索引。

```
from PIL import Image
pic = Image.open('flag.png')
w,h = [],[]
for i in range(pic.size[0]):
    if(pic.getpixel((i,15)) == (255,255,255)):
        w.append(i)
for i in range(pic.size[1]):
    if(pic.getpixel((24,i)) == (255,255,255)):
        h.append(i)
img = Image.new('RGB',(len(w),len(h)),(255,255,255))
for i in range(len(w)):
    for j in range(len(h)):
        img.putpixel((i,j),pic.getpixel((w[i],h[j])))
img.show()
```



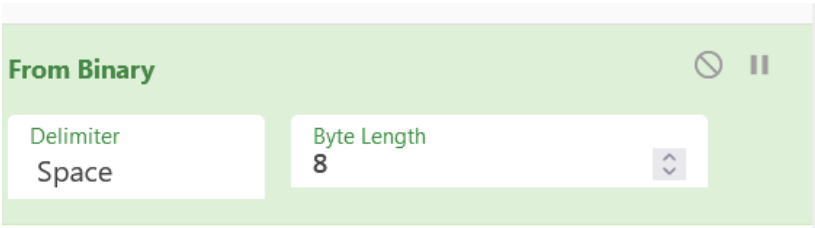

```
vnctf{34aE@w}
```

Strange flag

其实就是用tree里面是否有文件夹来代表0和1，是一种esolang，叫Folders。举例：

```
|-- New\ Folder\ (3)
|   |-- New\ folder
|   |   |-- New\ folder                 0
|   |   |   |-- New\ folder             1
|   |   |   |-- New\ folder(2)
|   |   |   |   `-- New\ folder
|   |   |   |-- New\ folder(3)          1
|   |   |   |   `-- New\ folder
|   |   |   `-- New\ folder(4)
|   |   |       `-- New\ folder         1
|   |   `-- New\ folder(2)
|   |       |-- New\ folder
|   |       |-- New\ folder(2)          0
|   |       |   `-- New\ folder         1
|   |       |-- New\ folder(3)
|   |       |   `-- New\ folder
|   |       `-- New\ folder(4)          1
|   |-- New\ folder(10)                 0
|   |   |-- New\ folder                 ------
|   |   |   |-- New\ folder
|   |   |   |-- New\ folder(2)          0
|   |   |   |-- New\ folder(3)          0
|   |   |   |   `-- New\ folder         1
|   |   |   `-- New\ folder(4)
|   |   |       `-- New\ folder
|   |   `-- New\ folder(2)              1
|   |       |-- New\ folder             1
|   |       |   `-- New\ folder
|   |       |-- New\ folder(2)
|   |       |-- New\ folder(3)          0
|   |       `-- New\ folder(4)          0
|   |           `-- New\ folder
```


然后手撸




```
01111011
01100100
00110010
00110011
00111001
00110000
00110011
00111000
00110111
00111001
01100100
01100110
00110101
00110111
```

Output

```
{d23903879df57503879bcdf1efc141fe}
```


只能说第一次还少了一个字母

```
vnctf{d23903879df57503879bcdf1efc141fe}
```

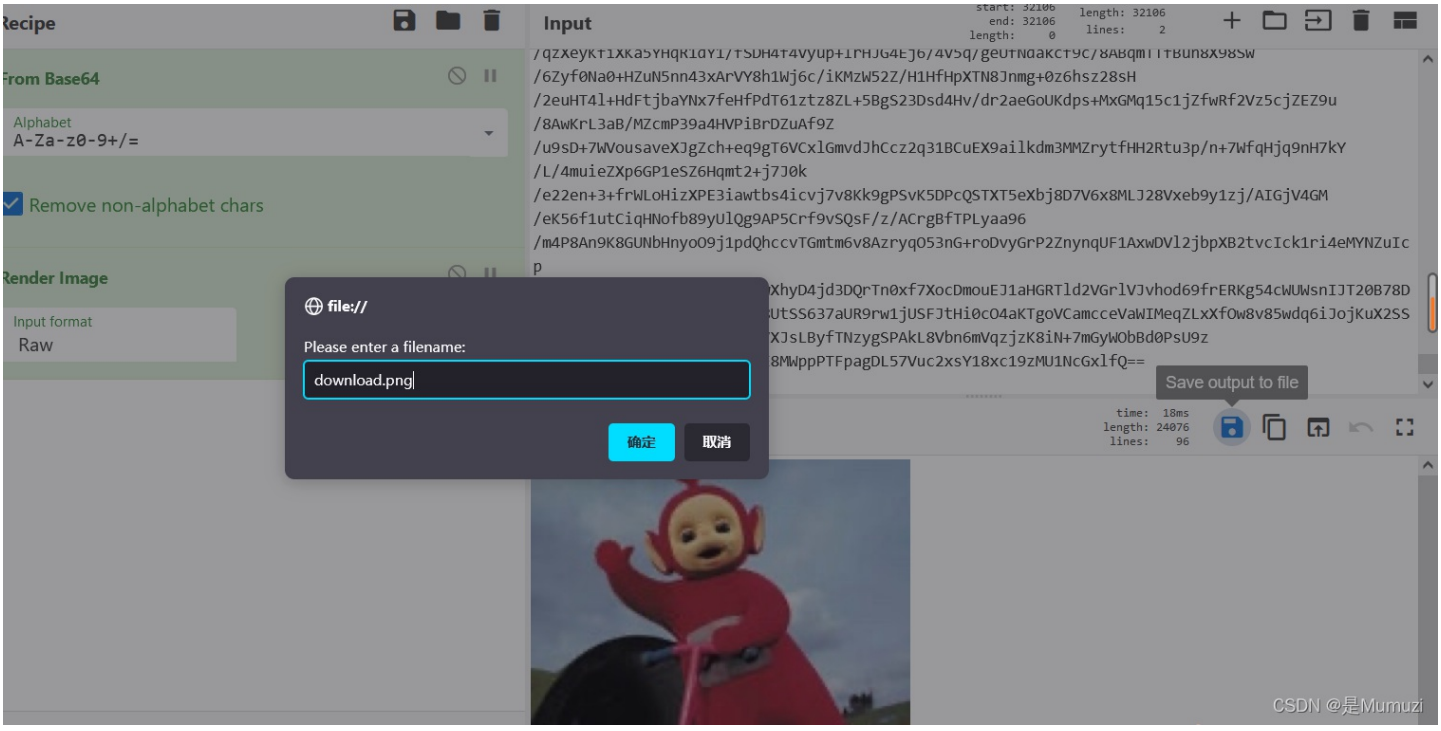
simple macos

不是给了hint说是弱密码吗，一开始我就对着keychain和login-keychain爆破，后者用了rockyou，前者用了常见6000密码，依旧没有爆破出来。搜VNCTF(含大小写)及其16进制和base也没搜出什么名堂来。题目主要是说system，于是就去system文件夹找。除了Preboot以外其他都是空的，直接进去。然后就是一个个翻了（系统文件没看）

接着在/var/db下发现东西。为CryptoUserInfo.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>79FABCCE-3636-4266-A6CF-8E3BB40332B4</key>
 <dict>
  <key>FullName</key>
  <string>Scr1pt</string>
  <key>PasswordHint</key>
  <string>our secret need a password</string>
  <key>PictureData</key>
  <data>
  base串，不方便展示
  </data>
  <key>PictureFormat</key>
  <string>JPEG</string>
  <key>ShortName</key>
  <string>scr1pt</string>
  <key>UserType</key>
  <string>OpenDirectory</string>
 </dict>
</dict>
</plist>
```

passwordHint oursecret



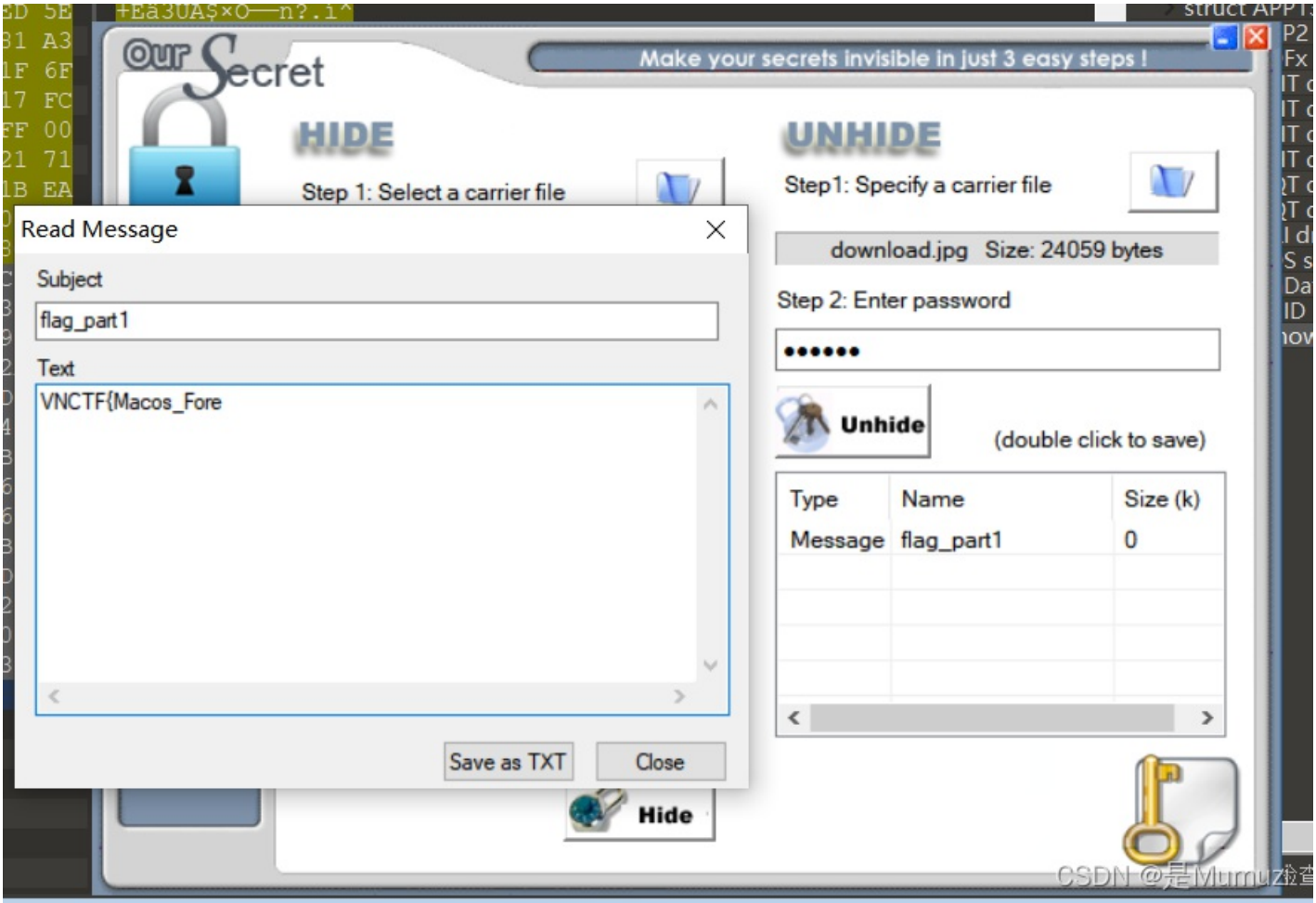
当然，不给提示看文件尾也是很明显是oursecret，因为oursecret靠的这些特征块来识别的

```
I..È...m98ilk;1<
1ji=1ij.ËçµnsIlc
```

很明显，文件尾现在掺杂了其他数据，且有}，猜测是后半部分的flag。于是复制出来并删掉下面部分

```
I..È...m98ilk;1<
1ji=1ij.ËçµnsIlc
_1s_s1MMple}
```

这时再结合弱口令。试了两个就出了。密码是123456



合起来得到flag

```
VNCTF{Macos_Forensllc_1s_s1MMple}
```

prize wheel

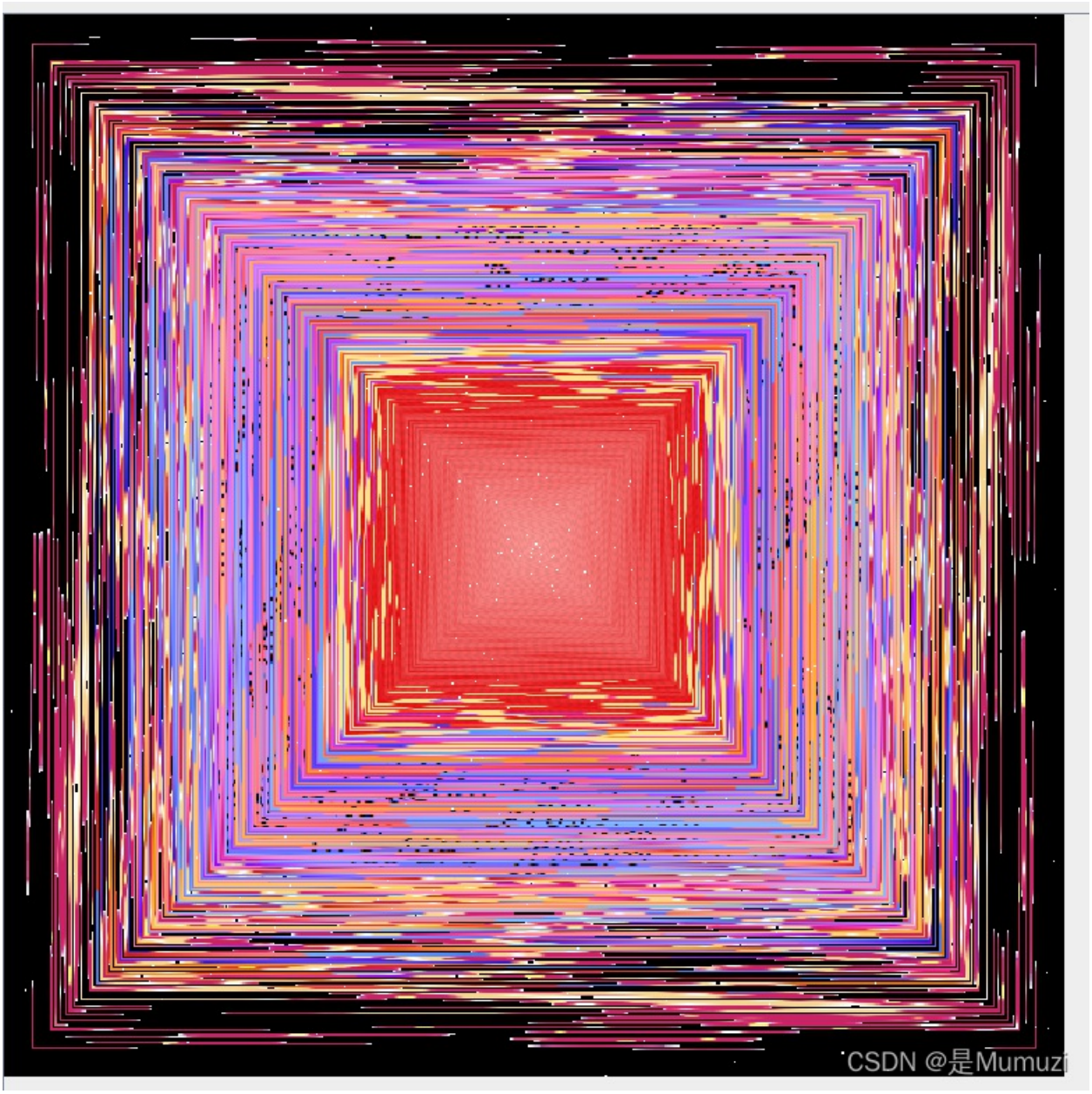
首先是python-exe-unpacker-master逆

然后再用BabyMaze说的方法弄出字节码

能很明显找到password：f6a623a2c577de3b46c079267d4bdd6e

```
 14 (print)
 24 ('Wow,you really get the password of zip!')
  1



 14 (print)
 25 ('the password is f6a623a2c577de3b46c079267d4bdd6e')
  1

226 (to 534)

  2 (a)
 26 (2)
  5 (>=)
  1
346
  2 (a)
 27 (20)
  1 (<=)
  1
346

 14 (print)
 28 ('congratulations!You get the password.')
```

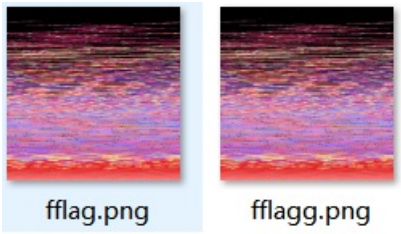
解压出来。



然后成功做了6个多小时

2022/2/12 15:45	PNG 图片文件	268 KB
2022/2/12 15:45	PNG 图片文件	737 KB
2022/2/12 13:13	PNG 图片文件	316 KB
2022/2/12 13:40	PNG 图片文件	257 KB
2022/2/7 12:15	PNG 图片文件	367 KB
2022/2/12 12:56	JetBrains PyCharm	5 KB
2022/2/12 18:43	PNG 图片文件	329 KB
2022/2/12 18:00	PNG 图片文件	1,158 KB

第一想法是一圈圈的读，然后重排，结果是这样的

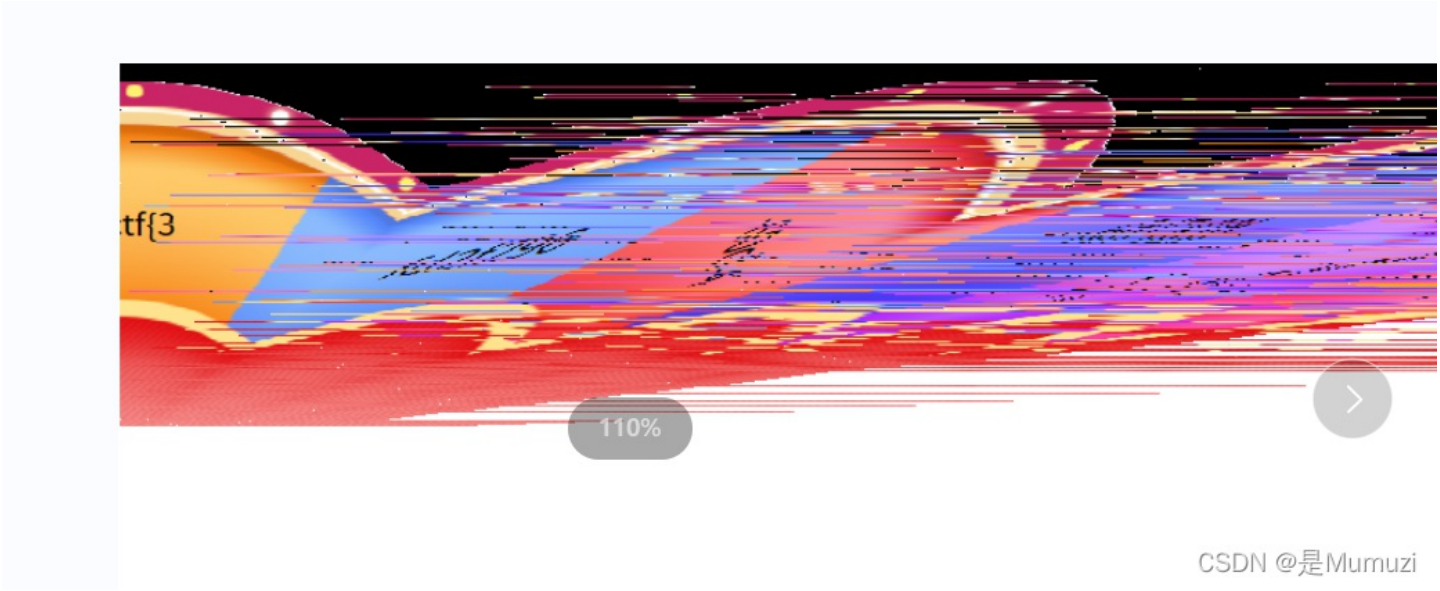


就很奇怪。

给出了索引这个hint，立刻联想到了[HECTF2021]七色彩虹、[b01lers2020]image_adjustments。具体是什么去看wp

于是有了更奇怪的图。这里因为调试覆盖掉了所以不放图。

之后，想到了将上面两者联系起来。即转圈圈读、找索引、平铺写入



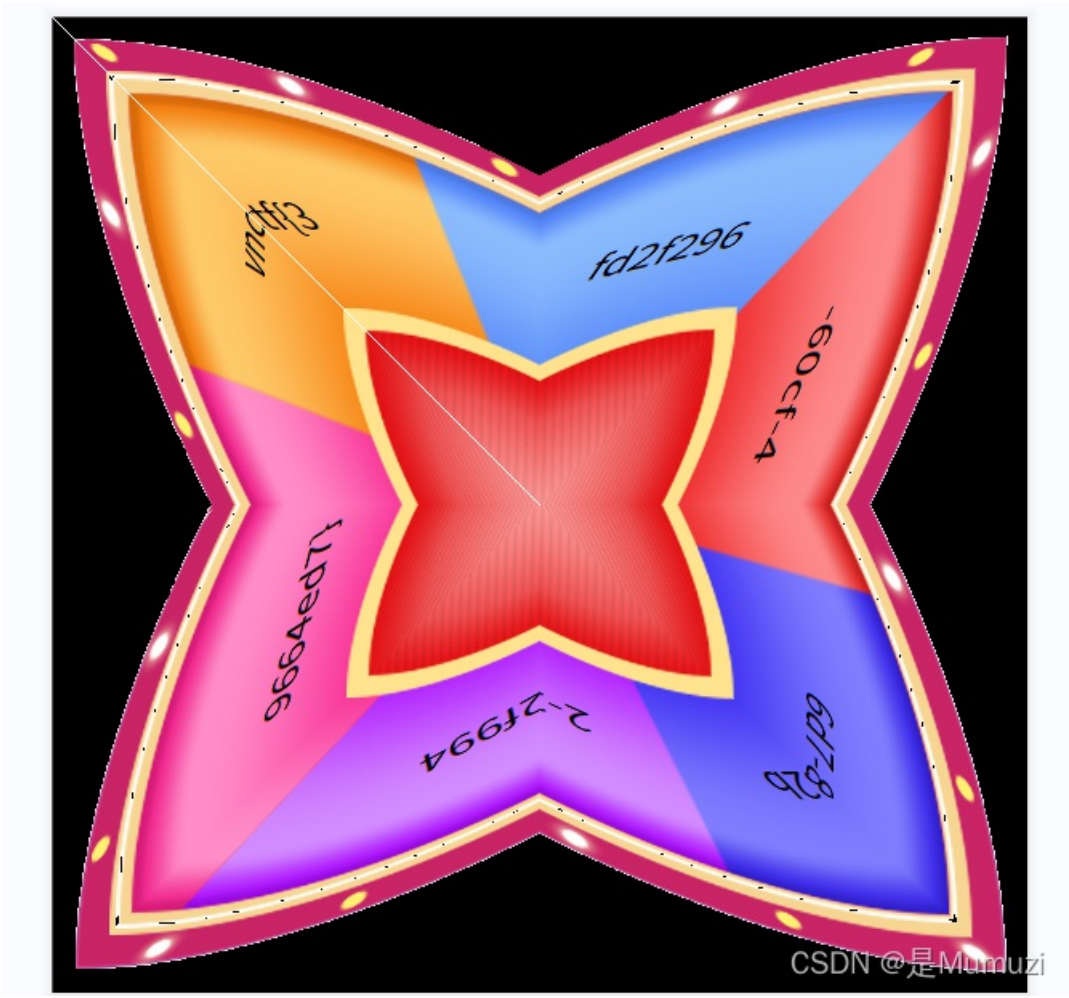
大概得到这个鸟样的图片。

很明显可以看出什么，那就是他们越来越扁。然后结合这是个大转盘，诶想出来了。

应该转圈圈读、找索引、转圈圈写入

之后就是调了快一个小时的脚本，我太菜了。

```python
from PIL import Image
def get_round(w):
    tb = []
    #从左上到右上、从右上到右下、从右下到左下、从左下到左上，分四步
    for i in range(w,609-w):
        tmp = img.getpixel((i,w))
        # print(tmp)
        tb.append(tmp)
    # print(len(tb))
    for i in range(w+1,609-w):
        tmp = img.getpixel((609-1-w,i))
        tb.append(tmp)
    for i in range(w+1,609-w):
        tmp = img.getpixel((609-1-i,609-w-1))
        tb.append(tmp)
    for i in range(w+1,609-w-1):
        tmp = img.getpixel((w,609-1-i))
        tb.append(tmp)
    # print(len(tb)) #2432== 609*2+607*2
    return tb

def put_round(w,img,tb): #用和上面一样的方法，只不过要放值进去，就这样。
    ind = 0
    for i in range(w,609-w):
        tmp = img.putpixel((i,w),tb[ind])
        ind += 1
    # print(len(tb))
    for i in range(w+1,609-w):
        tmp = img.putpixel((609-1-w,i),tb[ind])
        ind += 1
    for i in range(w+1,609-w):
        tmp = img.putpixel((609-1-i,609-w-1),tb[ind])
        ind += 1
    for i in range(w+1,609-w-1):
        tmp = img.putpixel((w,609-1-i),tb[ind])
        ind += 1

if __name__ == '__main__':
    # 除去最中间那个点，一共(609-1)//2
    img = Image.open('flag.png')
    pic = Image.new('RGBA',(609,609),(255,255,255,255))
    for i in range(304):
        table = get_round(i)
        ind = table.index((255,255,255,255))
        # print(ind)
        new_table = table[ind:] + table[:ind] #当时这里我一直是用两个循环去弄的，跟个傻子一样
        put_round(i,pic,new_table)
    pic.show()
```




```
vnctf{3fd2f296-60cf-46d7-82b2-2f9949664ed7}
```

这里为了更好看一点，小修一下。

（后面部分属于赛后复现）

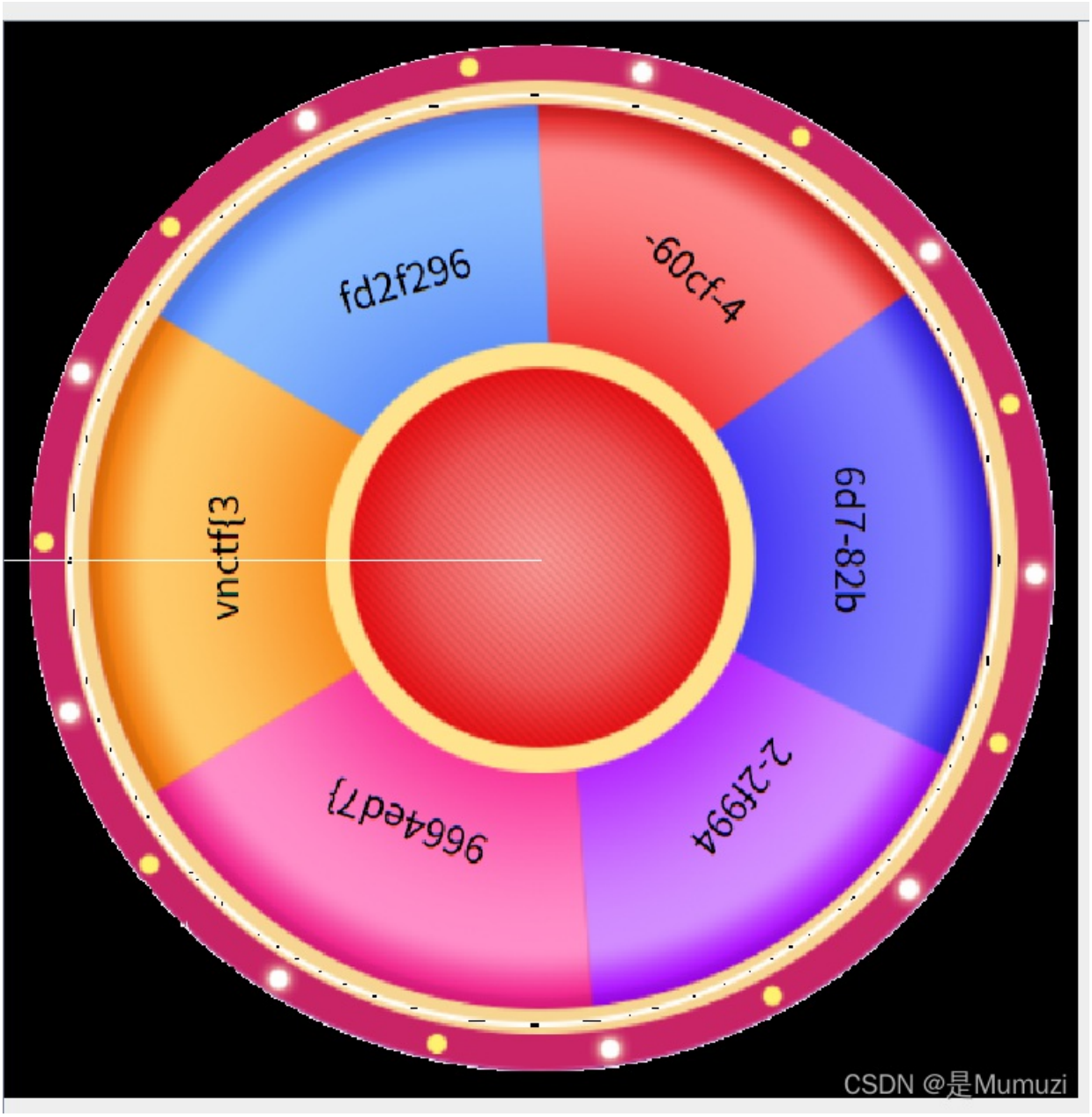
因为这里可以发现，都是从(i,i)点作为起点，所以那根白色的线，是斜着的。

那么想办法让他横着就行。

但是捏，依然要从(i,i)点作为起点。这样的话，只需要让白色像素在横着的水平面或者竖着的水平面上就可以了

只需要把main改成这样：

```python
if __name__ == '__main__':
    # 除去最中间那个点，一共(609-1)//2
    img = Image.open('flag.png')
    pic = Image.new('RGBA',(609,609),(255,255,255,255))
    for i in range(304):
        table = get_round(i)
        ind = table.index((255,255,255,255))
        # print(ind)
        new_table = table[ind:] + table[:ind] #当时这里我一直是用两个循环去弄的，跟个傻子一样
        new_table = new_table[len(new_table)//8:] + new_table[:len(new_table)//8]
        put_round(i,pic,new_table)
    # pic.show()
    pic.save('real_flag.png')
```



我超！太好看了叭
你可能以为一开始代码就这么简洁
错了
一开始的代码边调试边猜边写一共200多行，乱的一批。
甚至一个写出了一个高叟操作花了十五分钟才在不报错只是一片白的图里面知道原因

```
                    for k in range(f,len(table)):
                        img.putpixel((i_2, j_2),table[f])
```