虎符网络安全赛道 Re Game

原创

Hk_Mayfly 于 2020-04-23 23:08:00 发布 337 收藏

测试文件：https://lanzous.com/ibuflzc

手工还原

这道题没办法，只能自己结合dis.dis手工还原代码。得到：

```
# -*- coding:utf-8 -*-

# 8
arr0 = [249, 91, 149, 113, 16, 91, 53, 41]
# 20
arr1 = [43, 1, 6, 69, 20, 62, 6, 44, 24, 113, 6, 35, 0, 3, 6, 44, 20, 22, 127, 60]
# 12
arr2 = [90, 100, 87, 109, 86, 108, 86, 105, 90, 104, 88, 102]

# 应该是判断输入字符在32~128
def check0():
    pass

def check1(s):
    if len(s) < 100 and (((len(s) * len(s)) % 777) ^ 233 == 513):
        return True
    else:
        return False

def check2(s):
    if (((((ord(s[0]) * 128 + ord(s[1])) * 128 + ord(s[2])) * 128 + ord(s[3])) * 128 + ord(s[4])) * 128 +
ord(
            s[5]) == 3533889469877) and (ord(s[-1]) == 125):
        return True
    else:
        return False


def check3(s):
    arr = map(ord, s)
    a = arr[6:30:3]
    for i in range(len(a)):
        if (a[i] * 17684 + 372511) % 257 != arr0[i]:
            return False
    b = arr[-2:33:-1] * 5
    print (map(chr,b))
    c = map(lambda b: b[0] ^ b[1], zip(b, arr[7:27]))
    print (c)
    if c != arr1:
        return False
    p = 0
    for i in range(28, 34):
        if (((arr[i] + 107) // 16 + 77) != arr2[p]) or ((arr[i] + 117) % 16 + 99) != arr2[p + 1]:
            return False
        p = p + 2
    return True
```

代码分析

我们逐一分析就行。

check1

```
def check1(s):
    if len(s) < 100 and (((len(s) * len(s)) % 777) ^ 233 == 513):
        return True
    else:
        return False
```

通过这一段代码，我们能够知道flag的长度为39

```
from math import *

for i in range(100):
     len_s = sqrt(744+777*i)
     if (len_s%1) == 0 and len_s < 100:
             print(len_s)
```

check2

```
def check2(s):
    if (((((ord(s[0]) * 128 + ord(s[1])) * 128 + ord(s[2])) * 128 + ord(s[3])) * 128 + ord(s[4])) * 128 +
ord(
            s[5]) == 3533889469877) and (ord(s[-1]) == 125):
        return True
    else:
        return False
```

这里使用了flag的前五个字符，且最后一个字符为**'}'**，猜测flag的前四个字符为**'flag{'**，因此很容易求出第五个字符为**'5'**

```
chr(3533889469877 - ((((ord(s[0]) * 128 + ord(s[1])) * 128 + ord(s[2])) * 128 + ord(s[3])) * 128 +
ord(s[4])) * 128)
```

check3

这个函数外面分成三个部分分析

第一部分

```
    arr = map(ord, s)
    a = arr[6:30:3]
    for i in range(len(a)):
        if (a[i] * 17684 + 372511) % 257 != arr0[i]:
            return False
```

这部分，使用了索引6,9,12,15,18,21,24,27的字符。

```
arr0 = [249, 91, 149, 113, 16, 91, 53, 41]

for i in arr0:
    for n in range(10000):
        num = ((i+257*n)-372511) / 17684
        if num%1 == 0 and num <= 256 and num > 0:
            print (chr(int(num)),end="")
```

得到s[6],s[9],s[12],s[15],s[18],s[21],s[24],s[27] = **L5xiV5PK**

第二部分

```
    b = arr[-2:33:-1] * 5
    print (map(chr,b))
    c = map(lambda b: b[0] ^ b[1], zip(b, arr[7:27]))
    print (c)
    if c != arr1:
        return False
```

整个就是两两异或，这部分我们需要使用第一部分的结果，我们知道第一部分解出索引6,9,12,15,18,21,24,27的字符，第二部分计算了37,36,35,34和7,8,9,10,11,12,...,26这两部分的代码，第一部分和第二部分的第二部分有重合，我们可以利用重合部分计算索引37,36,35,34的值。

重合的索引有9,12,15,18,21,24，且我们可以知道，flag[7]和flag[37]异或得到arr1[0]，8和38得到arr[1]，每四个循环一次。

因此，n和4求余为3的和flag[37]异或得到arr1[n-7]；n和4求余为0的和flag[36]异或得到arr1[n-7]；n和4求余为1的和flag[35]异或得到arr1[n-7]... ....

那么我们就可以知道flag[9]和flag[35]异或得到arr1[2]；flag[12]和flag[36]异或得到arr1[5]... ....

逆向异或可以得到索引37,36,35,34的值，有了这些值，我们又可以通过arr1反解出7,8,9,10,11,12,...,26的值

```
arr1 = [43, 1, 6, 69, 20, 62, 6, 44, 24, 113, 6, 35, 0, 3, 6, 44, 20, 22, 127, 60]

s = chr(ord('i')^arr1[15-7]) + chr(ord('x')^arr1[12-7]) + chr(ord('5')^arr1[9-7]) + chr(ord('V')^arr1[18-7])
print (s)
model = [ord(x) for x in s]
flag = ''.join([chr(model[x%4]^arr1[x]) for x in range(len(arr1))])
print (flag)
```

第一部分的值为：**qF3u**，第二部分的值为：**ZG50ex5Yi75VqE5YePLl**

第三部分

```
    p = 0
    for i in range(28, 34):
        if (((arr[i] + 107) // 16 + 77) != arr2[p]) or ((arr[i] + 117) % 16 + 99) != arr2[p + 1]:
            return False
        p = p + 2
```

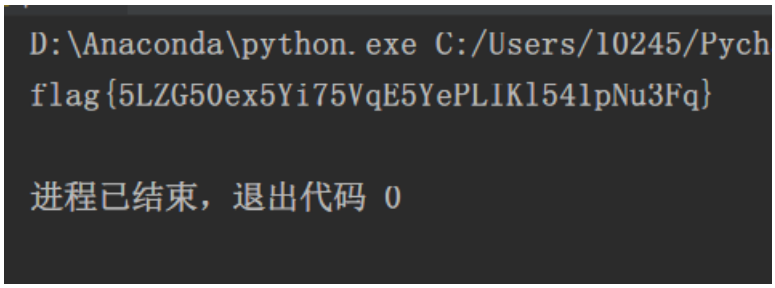
这就是个等式判断，直接爆破反解就行，得到：**l541pN**

```
arr2 = [90, 100, 87, 109, 86, 108, 86, 105, 90, 104, 88, 102]

for i in range(0, len(arr2), 2):
    for ch in range(256):
        if int((ch + 107) // 16) + 77 == arr2[i] and ((ch + 117) % 16) + 99 == arr2[i + 1]:
            print (chr(ch),end="")
```

综合上面的解，我们就能到的flag为：**flag{5LZG50ex5Yi75VqE5YePLlKl541pNu3Fq}**

脚本

```
# -*- coding:utf-8 -*-

flag = [''] * 39

flag[0:5] = 'flag{'
flag[5] = chr(3533889469877 - (
            (((ord(flag[0]) * 128 + ord(flag[1])) * 128 + ord(flag[2])) * 128 + ord(flag[3])) * 128 + ord(
        flag[4])) * 128)
flag[-1] = '}'

arr0 = [249, 91, 149, 113, 16, 91, 53, 41]

tmp1 = 6
for i in arr0:
    for n in range(10000):
        num = ((i + 257 * n) - 372511) / 17684
        if num % 1 == 0 and num <= 256 and num > 0:
            flag[tmp1] = chr(int(num))
            tmp1 = tmp1 + 3

arr1 = [43, 1, 6, 69, 20, 62, 6, 44, 24, 113, 6, 35, 0, 3, 6, 44, 20, 22, 127, 60]

flag[37:33:-1] = chr(ord('i') ^ arr1[15 - 7]) + chr(ord('x') ^ arr1[12 - 7]) + chr(ord('5') ^ arr1[9 - 7]) +
chr(
    ord('V') ^ arr1[18 - 7])

model = [ord(x) for x in flag[37:33:-1]]
flag[7:27] = ''.join([chr(model[x % 4] ^ arr1[x]) for x in range(len(arr1))])

arr2 = [90, 100, 87, 109, 86, 108, 86, 105, 90, 104, 88, 102]

tmp2 = 28
for i in range(0, len(arr2), 2):
    for j in range(256):
        if int((j + 107) // 16) + 77 == arr2[i] and ((j + 117) % 16) + 99 == arr2[i + 1]:
            flag[tmp2] = chr(j)
            tmp2 = tmp2 + 1
print (''.join(flag))
```



get flag！

> flag{5LZG50ex5Yi75VqE5YePLIKl541pNu3Fq}