2020-虎符网络安全赛道 reverse game

原创

Y0ng. 于 2020-04-29 21:14:48 发布 353 收藏

分类专栏： reverse 文章标签： 编程语言

版权声明：本文为博主原创文章，遵循 CC 4.0 BY 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/YANG12345_6/article/details/105848655
版权

reverse 专栏收录该内容
8 篇文章 0 订阅
订阅专栏

新手一枚，如有错误，请指正

题目给了我们一个.txt文档，打开看一下是一段类Python代码。
一段一段的分析：

前三块代码声明了三个列表：

```
arr0[8]=[249,91,149,113,16,91,53,41]

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

arr2[12]=[90,100,87,109,86,108,86,105,90,104,88,102]
```

由分析可以看出，这一个Python文件是由 check0 check1 check2 check3四个函数构成的

首先看check0

```
def check0:  #此处调用了genexpr函数
      ord(x) in range (32,128)  #由此可知flag的ascii码在(32, 128)之间
```

再看check1：

```
def check1:    #513^233=744  再爆破求s 也就是flag的长度
      if(len(s)<100  and (len(s)*len(s)%777)^233==513):
           return true
```

之后check2：check2 表明了flag的前五位和最后一位

```


def check2:
      (((((ord(s[0])*128+ord(s[1]))*128+ord(s[2]))*128+ord(s [3]))*128+ord(s[4]))*128+ord(s[5]))*128 =353388894
69877
      s[-1]='}'
```

由这个式子可以看出是以128进制计算的，所以第一种方法可以使用128进制的计算方法来计算flag的前五位。
因为我们知道flag的前四位是“flag{”，所以将前四位的ascii码值代入算式里面，可求出第五位是‘5’’。

再看check3函数，此函数分为三部分：

第一部分：

```
    #flag的 [6:30:3] 个

    map(ord(s),arr)  #此处将arr内所有的字符换成ascii码形式。多理解几下  arr也就是flag
    a[]=arr[6:30:3]  #a的长度是（29-6）/3=8，a[0,1,2,3...8]=arr(flag)[6,9,12...27]

    for i in range len(a):
         for n in range (32,128):
              if  (n*17684+372511)%257==arr0[i]:
                   flag[6+i*3]=j
```

第二部分

```
      #flag的 [7:27]和  [34:38]


      b = arr[-2:33:-1] * 5

      #根据Python的汇编代码理解的是这样的，但是  *5 这里有点懵，arr的-2到33也就是 37到34，总共有4个： flag[37],flag[36]
,flag[35],flag[34] 乘以5可能是因为个数少，进行循环操作。再看下面

      c = map(lambda b:b[0]^b[1] , zip(b,flag[7,27]))
      c[i] == arr1[i]
```

分析一下zip和map两个函数： zip: 例：zip(x[1,2,3] , y[3,4]) =[(1,3),(2,4)]

所以，这里的zip(b , arr[7,27]) 是把 arr[-2:33:-1]和arr[7:27]搞成以上20行，2列的形式，又因为arr[7:27]总共有20个数，所以上面的*5也就是循环的意思

下面再看map函数：例：map(lambda a: a*a , a[1,2,3]) = [1,4,9]
这里的map函数的第一个参数是一个函数，第二个参数是一个列表，第二个参数执行第一个参数的操作得到后面的结果。

所以这里的c = [b[]^arr[], b[]^arr[]...]

通过第一部分的操作，得到了flag[6,9,12,15,18,21,24,27]的值。再根据一一对应的关系，可以得出：
flag[35]= flag[9]^arr1[2],
flag[36]=flag[12]^arr1[5],
flag[37] = flag[15]^arr1[8],
flag[34]=flag[18]^arr1[11]

```
      #在这里可以逆向一下。求得flag[34:37]之后再算出flag[7:27]的值：
      x[0] = flag[37]
      x[1] = flag[36]
      x[2] = flag[35]
      x[3] = flag[34]
      y=x*5
      for i in range 20:
           flag[i+7] = arr1[i] ^ y[i]
```

第三部分：

```
    # flag的[28,34]

    p = 0
    for i in range(28,34):
        for j in range(32,128):
           if   (j+107)//16 + 77 == arr2[p] and (j+117)%16+99 == arr2[p+1]:
            flag[i] = j
            p+=2
            break
```

以上就是分析的题目给的全部的内容
下面根据分析写脚本：
exp：

```
arr0=[249,91,149,113,16,91,53,41]
arr1=[43,1,6,69,20,62,6,44,24,113,6,35,0,3,6,44,20,22,127,60]
arr2=[90,100,87,109,86,108,86,105,90,104,88,102]
flag=[65 for i in range(39)]
flag[0]=ord('f')
flag[1]=ord('l')
flag[2]=ord('a')
flag[3]=ord('g')
flag[4]=ord('{')
flag[5]=ord('5')
flag[38]=125
for i in range(8):
    for j in range(32,128):
        if(j*17684+372511) %257 == arr0[i]:
            flag[6+i*3]=j
            break
x=[0,0,0,0]
x[0]=arr1[8]^flag[15]
x[1]=arr1[17]^flag[24]
x[2]=arr1[14]^flag[21]
x[3]=arr1[11]^flag[18]
flag[37]=x[0]
flag[36]=x[1]
flag[35]=x[2]
flag[34]=x[3]
y=x*5
for i in range(20):
    flag[i+7]=arr1[i]^y[i]
p=0
for i in range (28,34):
    for j in range(32,128):
        if (j+107)//16+77 == arr2[p]  and (j+117)%16+99 == arr2[p+1]:
            flag[i]=j
            p+=2
            break
print(len(flag))
for i in range(39):
    print(chr(flag[i]),end='')

# 得出flag
```