首先第一部分，fun随机数产生函数，程序会使用fun(1，255，a)根据a产生32个1-255范围大小的随机数然后放在列表b中，a不同得到的列表b中的数据也不同，很关键的地方就是a被&0xff操作控制在了0-0xff范围内了。

然后第二部分呢，得到的列表b和输入input字符串在两个for循环中进行了运算，然后再比较，这一部分首先要想清楚flag的长度是72 – 32 + 1 = 41。然后这一部分不太好逆，我们使用z3来求解。用z3求解有个问题就是我们不知道随机数列表b的32个数据，但是这个列表b只有0xff种可能（即a的可能），所以我们选择爆破随机数列表b的数据，然后传给z3函数来解第二部分。

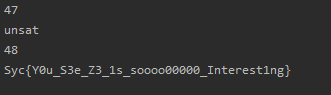
源码：

# -\*- coding:utf-8 -\*-  
import struct,time  
def fun( start,end,s):  
 a=32310901  
 b=1729  
 c=s #将种子seed赋值给rOld  
 m=end-start #得到m 模数  
 while True:  
 d=int(( a\*c+b )%m) #开始产生随机数  
 yield d #遇到yield关键字暂时挂起后面的代码，等带next(r)的调用并返回 rNew  
 c = d  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 arr = [77, 263, 394, 442, 463, 512, 667, 641, 804, 752, 885, 815, 1075, 1059, 1166, 1082, 1429, 1583, 1696, 1380,  
 1987, 2263, 2128, 2277, 2387, 2670, 2692, 3255, 3116, 3306, 3132, 3659, 3139, 3422, 3600, 3584, 3343, 3546,  
 3299, 3633, 3281, 3146, 2990, 2617, 2780, 2893, 2573, 2584, 2424, 2715, 2513, 2324, 2080, 2293, 2245, 2309,  
 2036, 1944, 1931, 1817, 1483, 1372, 1087, 1221, 893, 785, 697, 586, 547, 324, 177, 184]  
 flag = raw\_input("plz input your flag:")  
 length = len(flag)  
  
 #a是一个在0-0xff之间的值  
 a = struct.unpack("<I",flag[length-4:].encode())[0] & 0xff  
  
 #根据a得到一个有32个随机数的列表b  
 b = []  
 c = fun(1, 255, a)  
 for i in range(32):  
 b.append(next(c))  
  
 d = [0 for i in range(72)]  
 for i in range(length):  
 for j in range(32):  
 a = ord(flag[i]) ^ b[j]  
 d[i + j] += a  
  
 for i in range(len(d)):  
 if(d[i] != arr[i]):  
 print "fail"  
 time.sleep(5)  
 exit(0)  
 print "success"  
 time.sleep(5)  
 exit(0)

脚本：

#-\*- coding:utf-8 -\*-  
from z3 import \*  
import time  
  
def myrandint( start,end,seed):  
 a=32310901  
 b=1729  
 rOld=seed #将种子seed赋值给rOld  
 m=end-start #得到m 模数  
 while True:  
 rNew=int(( a\*rOld+b )%m) #开始产生随机数  
 yield rNew #遇到yield关键字暂时挂起后面的代码，等带next(r)的调用并返回 rNew  
 rOld = rNew  
  
def Z3(xor\_data):  
 s = Solver()  
 flag = [BitVec(('x%d' % i),8) for i in range(41) ]  
 cmp\_data = [77, 263, 394, 442, 463, 512, 667, 641, 804, 752, 885, 815, 1075, 1059, 1166, 1082, 1429, 1583, 1696, 1380,  
 1987, 2263, 2128, 2277, 2387, 2670, 2692, 3255, 3116, 3306, 3132, 3659, 3139, 3422, 3600, 3584, 3343, 3546,  
 3299, 3633, 3281, 3146, 2990, 2617, 2780, 2893, 2573, 2584, 2424, 2715, 2513, 2324, 2080, 2293, 2245, 2309,  
 2036, 1944, 1931, 1817, 1483, 1372, 1087, 1221, 893, 785, 697, 586, 547, 324, 177, 184]  
  
 xor\_result = [0 for i in range(72)]  
 for i in range(41):  
 for j in range(32):  
 a = flag[i] ^ xor\_data[j]  
 xor\_result[i + j] += a  
  
 for i in range(0,72):  
 s.add(xor\_result[i] == cmp\_data[i])  
  
 if s.check() == sat:  
 model = s.model()  
 str = [chr(model[flag[i]].as\_long().real) for i in range(41)]  
 print "".join(str)  
 time.sleep(5)  
 exit()  
 else:  
 print "unsat"  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 for seed in range(0xff):  
 xor\_data = []  
 r = myrandint(1, 255, seed)  
 for i in range(32):  
 xor\_data.append(next(r))  
 Z3(xor\_data)  
 print seed

当seed爆破到49时候就会出flag。



Flag：Syc{Y0u\_S3e\_Z3\_1s\_soooo00000\_Interest1ng}