2017第二届广东省强网杯线上赛Random

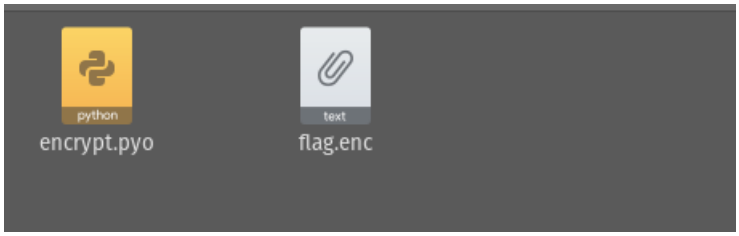
原创

cc啊昂昂 于 2021-02-24 15:31:10 发布 84 收藏

文章标签： CTF

本文链接：https://blog.csdn.net/qq_45735611/article/details/113887588

1. 题目给出2个文件，一个pyo为文件和一个enc。

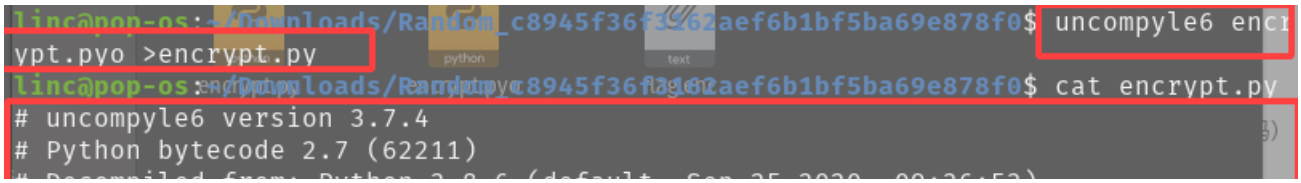


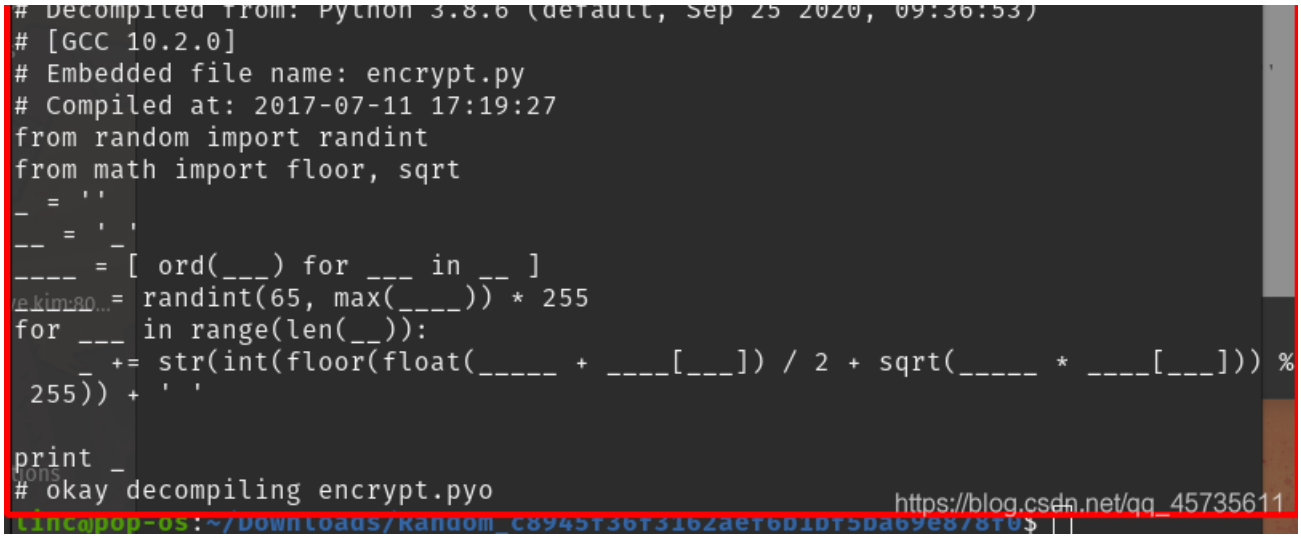
2. 打开pyo文件发现是一个二进制文件

3. 不太清除pyo是什么，去搜索一下，发现如下内容
    1. py：源码文件，由 Python 程序解释。
    2. pyc：源码经编译后生成的二进制字节码（Bytecode）文件。
    3. pyo：优化编译后的程序，也是二进制字节码文件。
4. pyo文件阅读不来，而pyo又由py文件编译而来推测需要反编译得到py文件,搜索如何反编译pyo文件。

```
pip install uncompyle
```

5. 查看得到的py文件，发现还是不能完全''可读

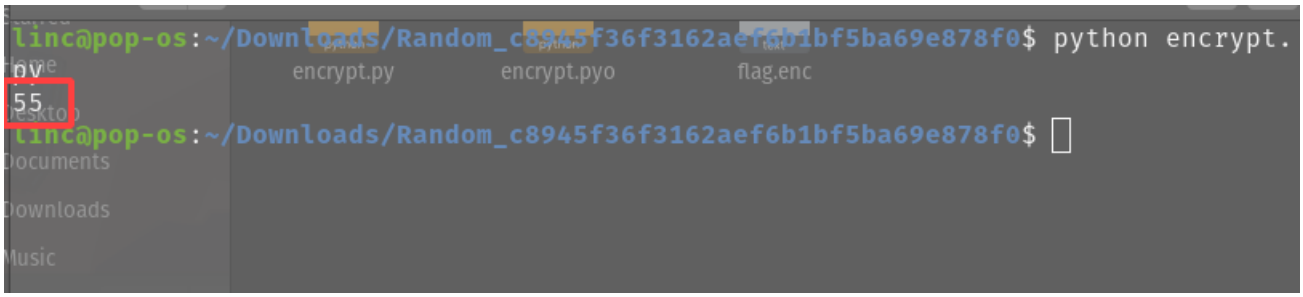




6. 搜索了一些资料后推测是做了加密措施，替换了变量名，那么以此来改写刚刚的py文件

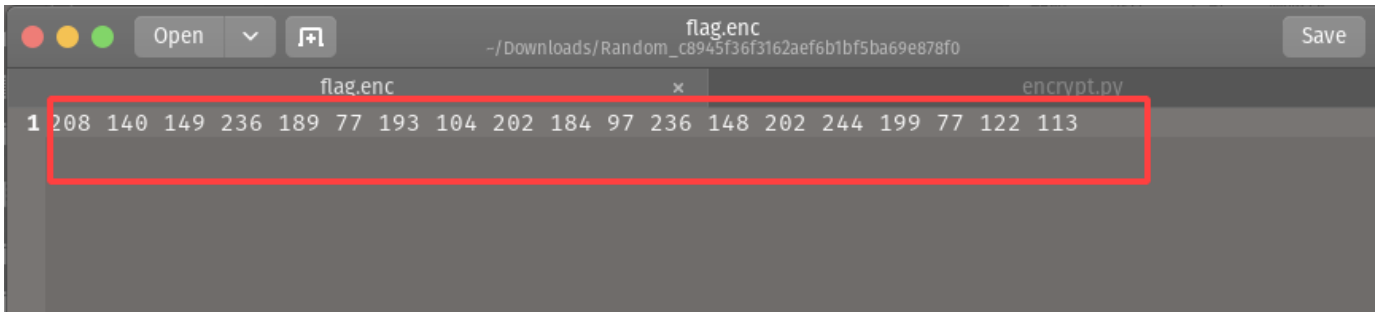
```
# uncompyle6 version 3.7.4
# Python bytecode 2.7 (62211)
# Decompiled from: Python 3.8.6 (default, Sep 25 2020, 09:36:53)
# [GCC 10.2.0]
# Embedded file name: encrypt.py
# Compiled at: 2017-07-11 17:19:27
from random import randint
from math import floor, sqrt
a = ''
aa = '_'
aaaa = [ ord(aaa) for aaa in aa ]
aaaaa = randint(65, max(aaaa)) * 255
for aaa in range(len(aa)):
    a += str(int(floor(float(aaaaa + aaaa[aaa]) / 2 + sqrt(aaaaa * aaaa[aaa])) % 255)) + ' '

print (a)
# okay decompiling encrypt.pyo
```

7. 随手运行一下，发现得到一个数字



8. 这里通过py代码推测加密了aa变量。

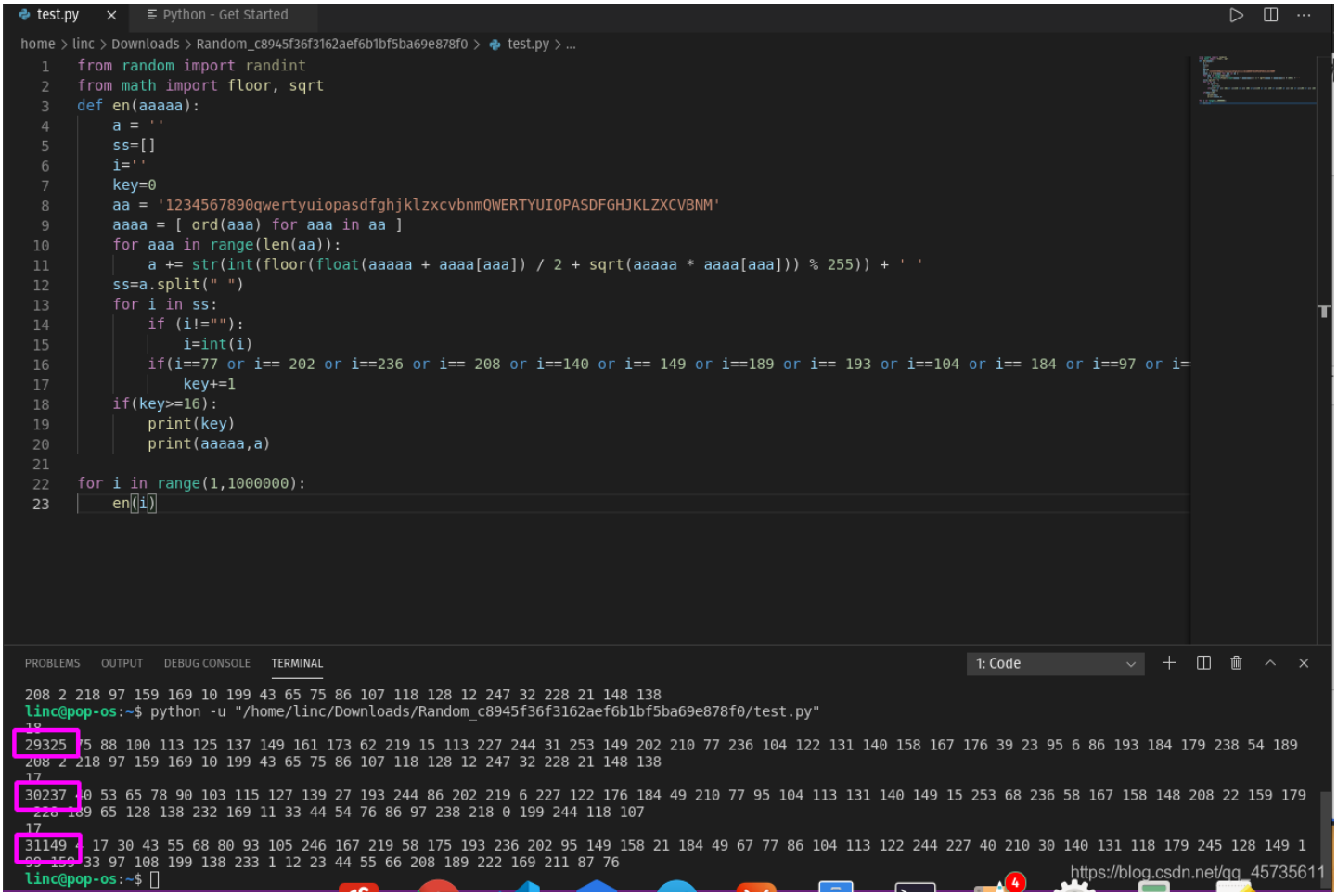
9. 打开另外一个文件，发现一串数字，推测是由第一个py文件加密得到的数字



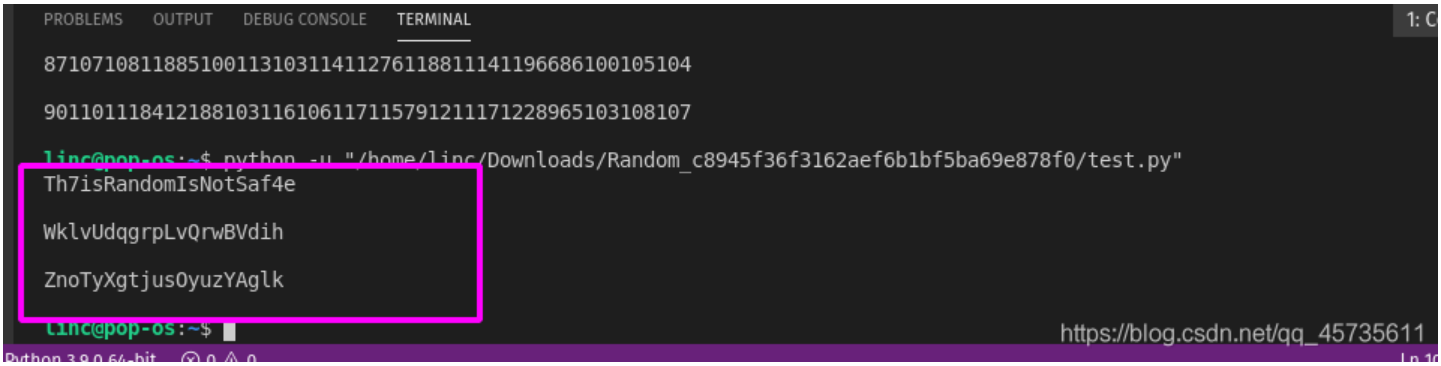
10. 想要一个字符一个字符去加密后比对，但加密过程中有随机数参与运算，此时需要求出次随机数
11. 再次观察那串数字，发现3对数字重复

12. 改写代码，采用爆破的方法获得3个可能的随机数

```python
from random import randint
from math import floor, sqrt
def en(aaaaa):
 a = ''
 ss=[]
 i=''
 key=0
 aa = '1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'
 aaaa = [ ord(aaa) for aaa in aa ]
 for aaa in range(len(aa)):
     a += str(int(floor(float(aaaaa + aaaa[aaa]) / 2 + sqrt(aaaaa * aaaa[aaa])) % 255)) + ' '
 ss=a.split(" ")
 for i in ss:
  if (i!=""):
   i=int(i)
  if(i==77 or i== 202 or i==236 or i== 208 or i==140 or i== 149 or i==189 or i== 193 or i==104 or i== 184 or i==
97 or i==148  or i==244 or i== 199 or i==122 or i== 113 ):
   key+=1
 if(key>=16):
  print(key)
  print(aaaaa,a)

for i in range(1,1000000):
 en(i)
```



13. 此时利用3个随机数反推加密数字,得到3个flag



```python
from random import randint
from math import floor, sqrt
def en(aaaaa,flag):
 a = ''
 aa = '1234567890qwertyuiopasdfghjklzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM'
 aaaa = [ ord(aaa) for aaa in aa ]
 for aaa in range(len(aa)):
  a=str(int(floor(float(aaaaa + aaaa[aaa]) / 2 + sqrt(aaaaa * aaaa[aaa])) % 255))
  if (a == flag):
   print(chr(aaaa[aaa]),end="")

randint=[29325,30237,31149]
flags="208 140 149 236 189 77 193 104 202 184 97 236 148 202 244 199 77 122 113".split(" ")
for i in randint:
 for j in flags:
  en(i,j)
 print("\n")
```

依次提交答案，发现都不正确，猜测是因为有1个或者2个字符的加密数字是一样的造成,仔细观察后发现第一个随机数可以有如下2对字符的加密数字一样

1 --G

e --4

原始得到：Th7isRandomIsNotSaf4e

1：Th7isRandomIsNotSaf44

2：Th7isRandomIsNotSafee

3:Th7isRandomIsNotSafe4

提交后发现还是不对，此时发现3组答案中的第一组好像是个句子。

```
linc@pop-os:~$ python -u "/home/linc/Downloads/Random_c8945f36f3162aef6b1bf5ba69e878f0/test.py"
Th7isRandomIsNotSaf4e
```

将其修改为可读的正确句子后提交，发现答案正确

4：ThisRandomIsNotSafe