2020——网鼎杯部分writeup

原创

逃课的小学生 于 2020-05-13 17:41:45 发布 3120 收藏 1
分类专栏： ichunqiu crypto 网鼎杯

1.you raise me up

题目源码如下

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
from Crypto.Util.number import *
import random

n = 2 ** 512
m = random.randint(2, n-1) | 1
c = pow(m, bytes_to_long(flag), n)
print 'm = ' + str(m)
print 'c = ' + str(c)

# m = 391190709124527428959489662565274039318305952172936859403855079581402770986890308469084735451207885380
# c = 666585139420321424585678945072365863252081679162179677590976689523300023402364287878602564495379799530
```

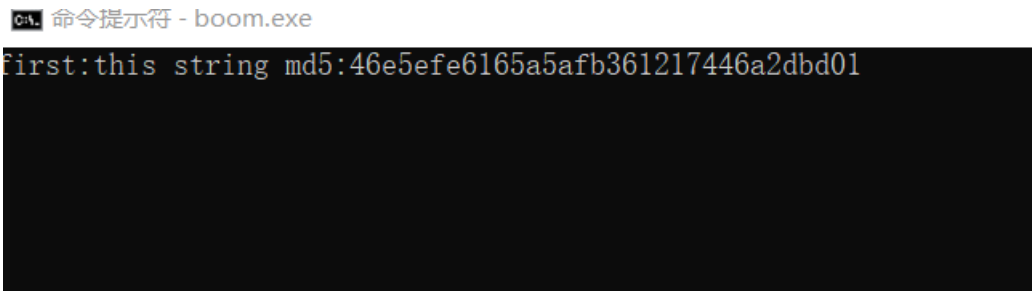
这是一个求解离散对数的问题——经过查询在sage下有discrete_log函数可以直接解决问题

```
e=discrete_log(Mod(c,n),Mod(m,n))
```

2.boom

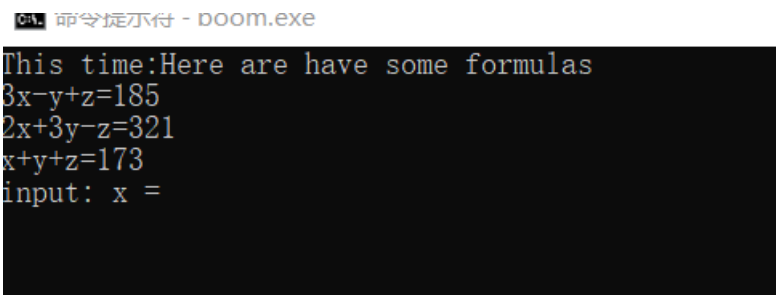
题目给出一个exe，我们运行exe解出每一道题即可有结果

第一关



去md5查询得到en5oy

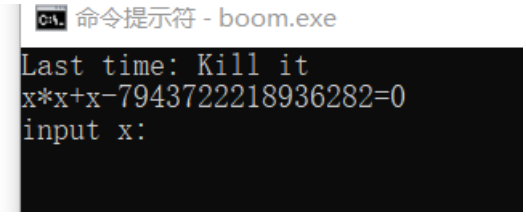
第二关



使用sympy解方程

```
from sympy import *
x = Symbol('x')
y = Symbol('y')
z = Symbol('z')
print solve([3*x-y+z-185,2*x+3*y-z-321,x+y+z-173],[x, y,z])
```

第三关



同上解方程

```
from sympy import *
x = Symbol('x')
print solve([x*x+x-7943722218936282],[x])
```

3.easy_ya

首先爆破获得之后的数据

```
from pwn import *
import gmpy2
def md5(a):
 for i in xrange(128):
  for j in xrange(128):
   for k in xrange(128):
    for o in xrange(128):
     jie=chr(i)+chr(j)+chr(k)+chr(o)
     ll=hashlib.md5(jie).hexdigest()[:20]
     if ll==a:
      return jie

def proof(a,b):
 if b=="md5":
  return md5(a)
 else:
  return ""
io=remote("39.96.90.217",17497)
io.recvuntil("= ")
hashh=io.recvuntil("\n")[:-1]
io.recvuntil("function openssl_")
hashfunction=io.recvuntil(">")[:-1]
print hashh.encode("hex")
print hashfunction
xx=proof(hashh,hashfunction)
io.sendline(xx)
io.interactive()
```

得到RSA加密的密文和异或加密的密文，根据RSA加密的密文获得key，由于两个RSA中p是相同的，我们可以使用求两个RSA的n的最大公约数求解p，分解n，从而解密密文获得key

```
import gmpy2
n1=78930649587473787462025164455888300508645133789007626345909338991391978449655109566054671744689716859129
c1=31085877631483456310809166284488223209916183059372170111112001737975007514570590449588025547483620296503
n2=89450975505510816143913290650301940314670407392775320805682956190022561907119991146818723844739113292931
c2=52897238924731258537862966369639269305568637517927802963866719096469205278307301780755193933426100099616

e=0x10001
p=gmpy2.gcd(n1,n2)
q1=n1/p
q2=n2/p
d1=gmpy2.invert(e,(p-1)*(q1-1))
d2=gmpy2.invert(e,(p-1)*(q2-1))
print hex(pow(c1,d1,n1))
print hex(pow(c2,d2,n2))
```

然后我们回到题目，发现在最后的结果中z的前三位和pads的后三位异或无法得到原来的值，这里我们采用爆破，首先保证pad是可以被32整除的且整除结果在(0x10000000,0xffffffff)之间，然后将结果是可见字符的部分留下(注意flag的结尾有填充\x00)

```
import gmpy2
limit = lambda n: n & 0xffffffff
key="8891898088b197a0bfa78199b28195bfae89".decode("hex")
Key  = [ord(i) for i in key]
a = limit((Key[0] << 24) | (Key[1] << 16) | (Key[2] << 8) | Key[3])
b = limit((Key[4] << 24) | (Key[5] << 16) | (Key[6] << 8) | Key[7])
c = limit((Key[8] << 24) | (Key[9] << 16) | (Key[10] << 8) | Key[11])
d = limit((Key[12] << 24) | (Key[13] << 16) | (Key[14] << 8) | Key[15])
y=int("939660b2b1d42d5f47fb"[:7],16)
pp=int("939660b2b1d42d5f47fb"[7:12]+"000",16)
z=int("939660b2b1d42d5f47fb"[12:20],16)
ll=[]
for j in xrange(0,34):
 pps=pp+j*0x100000000
 for k in xrange(0,0xfff):
  if (pps+k)%32==0:
   if (pps+k)/32>0x10000000 and (pps+k)/32<0xffffffff:
    ll.append(pps+k)
for pps in ll:
 pad=pps/32
 y=int("939660b2b1d42d5f47fb"[:7],16)
 z=int("939660b2b1d42d5f47fb"[12:20],16)
 sign=True
 z=((int(hex(pps)[-4:-1],16))<<20)^z
 for k in range(32):
  padss=limit(pps-k*pad)
  z=z-limit((y*16 + c) ^ (y + padss) ^ ((y>>5) + d))
  if z<0:
   z=z+0x100000000
  y=y-limit((z*16 + a) ^ (z + padss) ^ ((z>>5) + b))
  if y<0:
   y=y+0x100000000

 s=hex((y<<32)+z)[2:-1]
 s1=["","0"][len(s)%2]+s

 for i in xrange(len(s1)/2):
  if (int(s1[i*2:i*2+2],16)>128):
   sign=False
   break
 if sign:
  print s1.decode("hex")
```