

PENTEST

LoRexxar 的渗透之战之二

我怀疑这道题跟下一题的 hint 放反了,这道题我没有看出可以业务逻辑漏洞,而是水平越权;首先申请一个账号 ngc 密码 111111 从源码中进去得到 code,发现可以进行改密码并且在网页中不能改账号,那就在 firebug 中改为 LoRexxar,改完登录。

hctf{y3wu_louj1_l1_h4i_l3}

123

test

<_>alert(1)

111

flag

进击的 Crypto [4]

挺简单的数学题,看懂代码就好,写了个代码直接算出来了

```
k.py - C:\Users\lenovo\Desktop\k.py (2.7.13)
File Edit Format Run Options Window Help

from gmpy2 import invert
import hashlib
import math

p = 1907160274476467928441897336813583793273430611993693413844749025573097696829
q = 930788704028200015275140127068138499329817310955
g = 2202371560627246570864134638319544885657038758393686391531626458242597354756
r = 568752653628483014849549142909331362115254788206
z1 = 427262976273228083221871998313131945010029561209591706262118913937489577133
z2 = 835940898148680488372488685713345793755099380413493862399556052721366535745
s1 = 618159893787048300752592802884467155388759696698
s2 = 659836539307844663175437862395252943516139307036

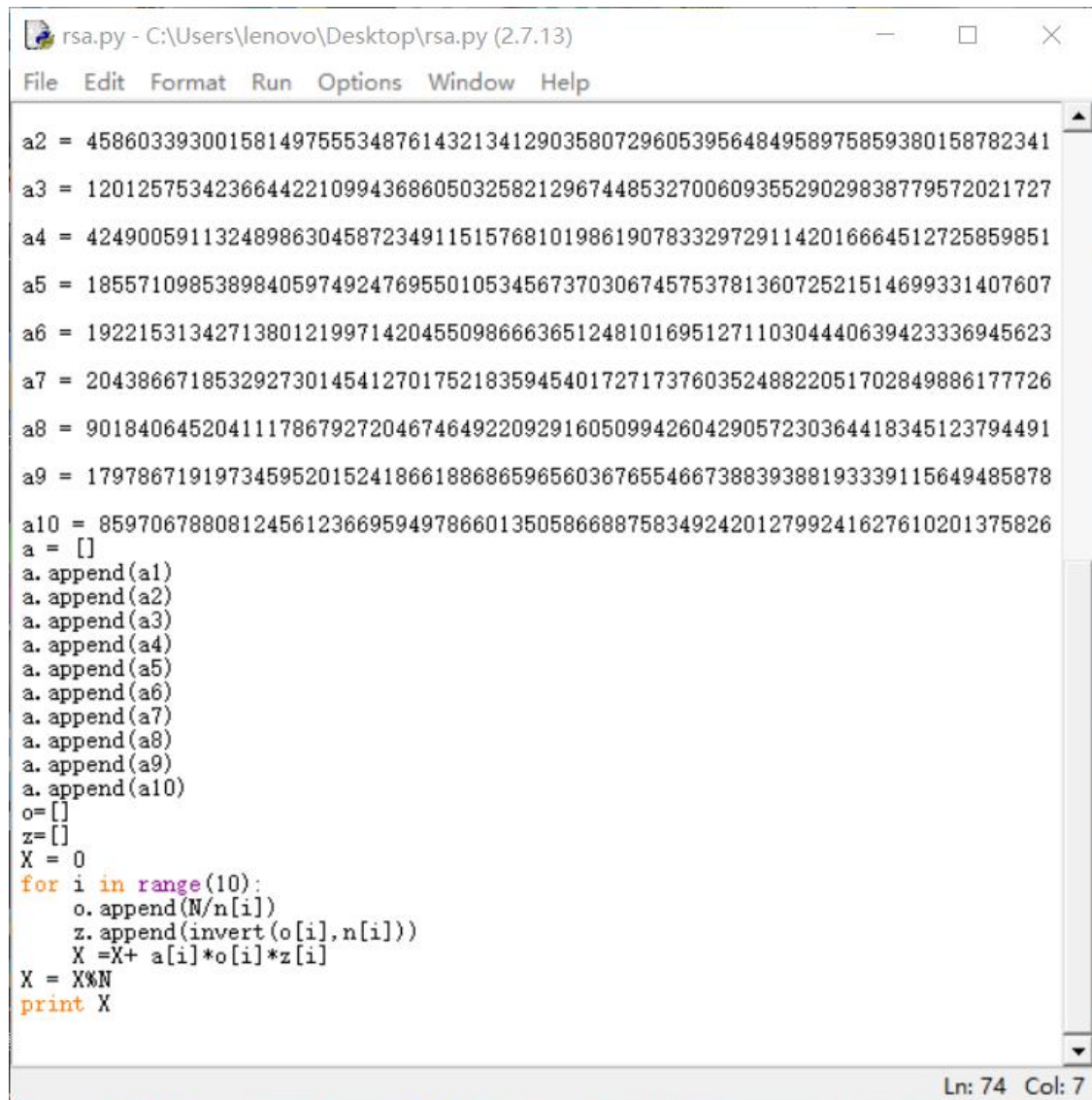
k = invert(((s1 - s2)*invert(z1 - z2, q)) % q, q)
print k
x = ((s1 * k - z1)*invert(r, q)) % q
def getflag(data):
    if data == "getflag":
        (r, s) = encrypt(data, p, q, g, x, k)
        flag = "hctf{" + str(s % r) + "}"
        print flag
def encrypt(data, p, q, g, x, k):
    r = pow(g, k, p) % q
    s = (invert(k, q) * (SHA1(data) + x * r)) % q
    return (r, s)
def SHA1(data):
    return data_to_int(hashlib.sha1(data).hexdigest())
def data_to_int(s):
    return int(s.encode('hex'), 16)
getflag('getflag')
```

```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help

Python 2.7.13 (v2.7.13:a06454b1afaf, Dec 17 2016, 20:53:40) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\lenovo\Desktop\k.py =====
209569583662944399958807472421680653
hctf{88169191231439818447681393510021281730269252095}
>>>
```

进击的 Crypto [5]

E 特别小，可以用哪个中国剩余定理求密文



```
rsa.py - C:\Users\lenovo\Desktop\rsa.py (2.7.13)
File Edit Format Run Options Window Help

a2 = 458603393001581497555348761432134129035807296053956484958975859380158782341
a3 = 120125753423664422109943686050325821296744853270060935529029838779572021727
a4 = 424900591132489863045872349115157681019861907833297291142016664512725859851
a5 = 185571098538984059749247695501053456737030674575378136072521514699331407607
a6 = 192215313427138012199714204550986663651248101695127110304440639423336945623
a7 = 204386671853292730145412701752183594540172717376035248822051702849886177726
a8 = 901840645204111786792720467464922092916050994260429057230364418345123794491
a9 = 179786719197345952015241866188686596560367655466738839388193339115649485878
a10 = 85970678808124561236695949786601350586688758349242012799241627610201375826
a = []
a.append(a1)
a.append(a2)
a.append(a3)
a.append(a4)
a.append(a5)
a.append(a6)
a.append(a7)
a.append(a8)
a.append(a9)
a.append(a10)
o = []
z = []
X = 0
for i in range(10):
    o.append(N/n[i])
    z.append(invert(o[i],n[i]))
    X = X + a[i]*o[i]*z[i]
X = X%N
print X

Ln: 74 Col: 7
```

n 为 n 的列表，a 为 c 的列表，N 为 n 的乘积，

最后得出十进制做 $1/10$ 次方，转十六进制转字符串

16进制转字符

字符转16进制

清空结果

When e are small and same,it can be Hastad's broadcast attack.Maybe we v
{Hastad's_broadcast_attack_is_interesting}

得到 flag。