攻防世界——Normal_RSA

原创

Irving- 于 2021-01-17 11:47:18 发布 373 收藏 2

分类专栏： 题解

本文链接：https://blog.csdn.net/weixin_51867782/article/details/112733178

版权

题解 专栏收录该内容

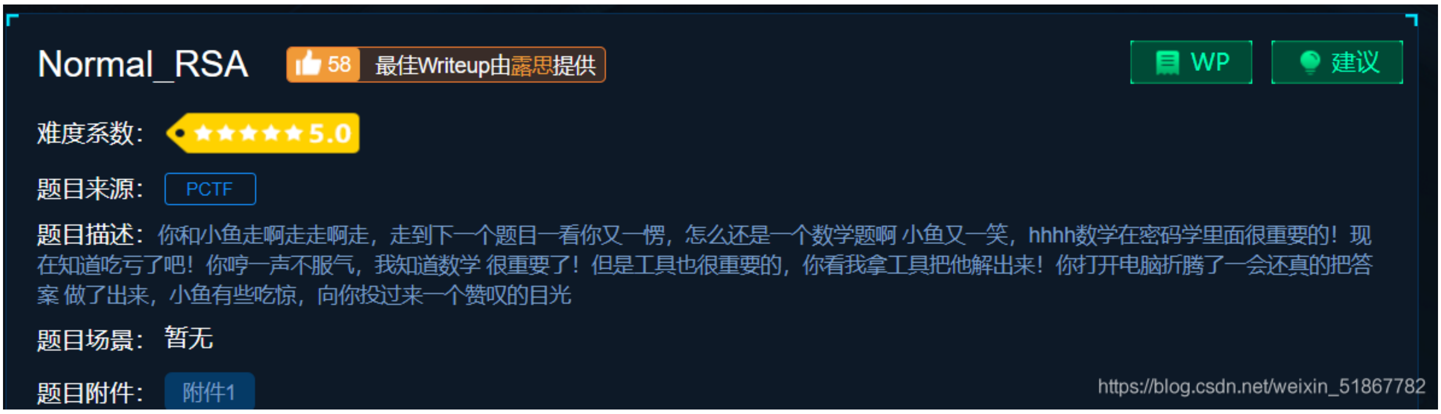
8 篇文章 0 订阅

订阅专栏

攻防世界——Normal_RSA

工具：**qpenssl**

文件：**flag.enc,应该是密文。pubkey.pem，是公钥。**



Normal_RSA 58 最佳Writeup由露思提供

难度系数： ★★★★★5.0

题目来源： PCTF

题目描述： 你和小鱼走啊走走啊走，走到下一个题目一看你又一愣，怎么还是一个数学题啊 小鱼又一笑，hhhh数学在密码学里面很重要的！现在知道吃亏了吧！你哼一声不服气，我知道数学 很重要了！但是工具也很重要的，你看我拿工具把他解出来！你打开电脑折腾了一会还真的把答案 做了出来，小鱼有些吃惊，向你投过来一个赞叹的目光

题目场景： 暂无

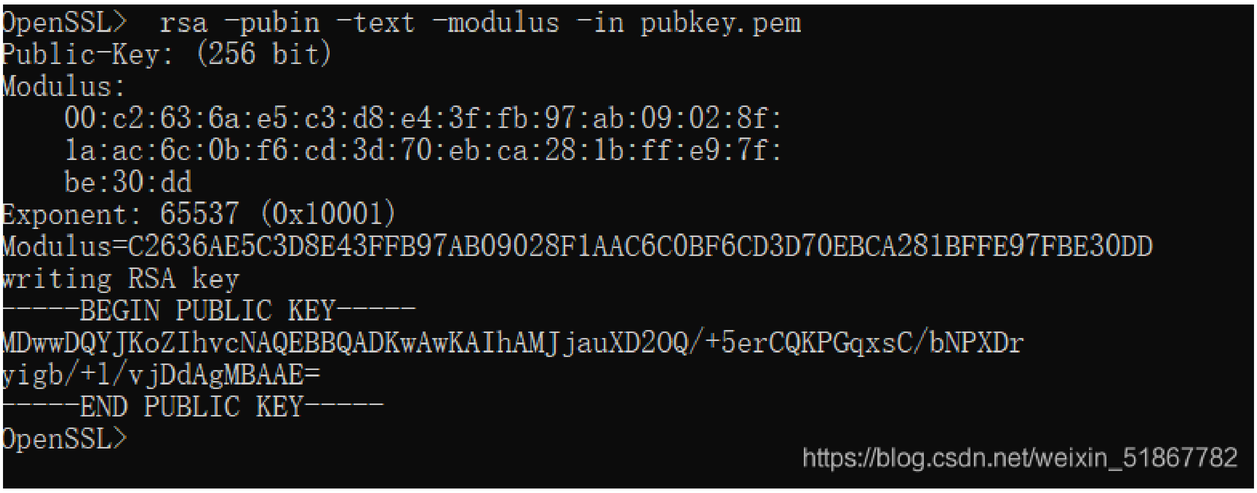
题目附件： 附件1

https://blog.csdn.net/weixin_51867782

1、打开后，拿到两个文件。enc文件和pem文件。pem文件需要用openssl打开。enc文件在哦脚本中打开（如果是kali系统，直接用openssl解密enc文件即可，这里不做详细阐述）

2、先下载openssl，之后打开openssl-bin-openssl.exe,在命令框输入

rsa -pubin -text -modulus -in pubkey.pem

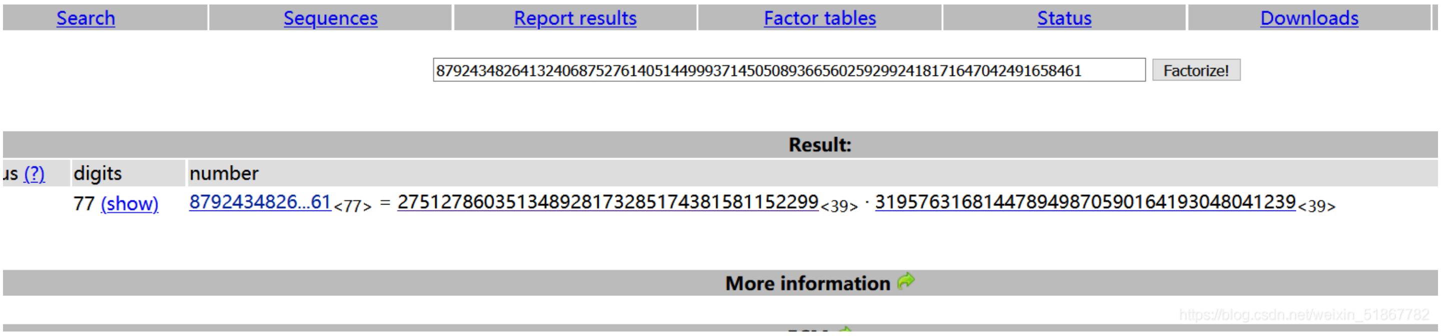


```
OpenSSL>  rsa -pubin -text -modulus -in pubkey.pem
Public-Key: (256 bit)
Modulus:
    00:c2:63:6a:e5:c3:d8:e4:3f:fb:97:ab:09:02:8f:
    1a:ac:6c:0b:f6:cd:3d:70:eb:ca:28:1b:ff:e9:7f:
    be:30:dd
Exponent: 65537 (0x10001)
Modulus=C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
yigb/+1/vjDdAgMBAAE=
-----END PUBLIC KEY-----
OpenSSL>
```

https://blog.csdn.net/weixin_51867782

Modulus代表的是N，E开头的代表e。

得到N后，先由16进制转为10进制，再进行整数分解。(转为十进制要用脚本，因为位数太多，一般的在线转换无法进行）



8792434826413240687527614051449993714505089366560259299241817164704249165846 1 Factorize!

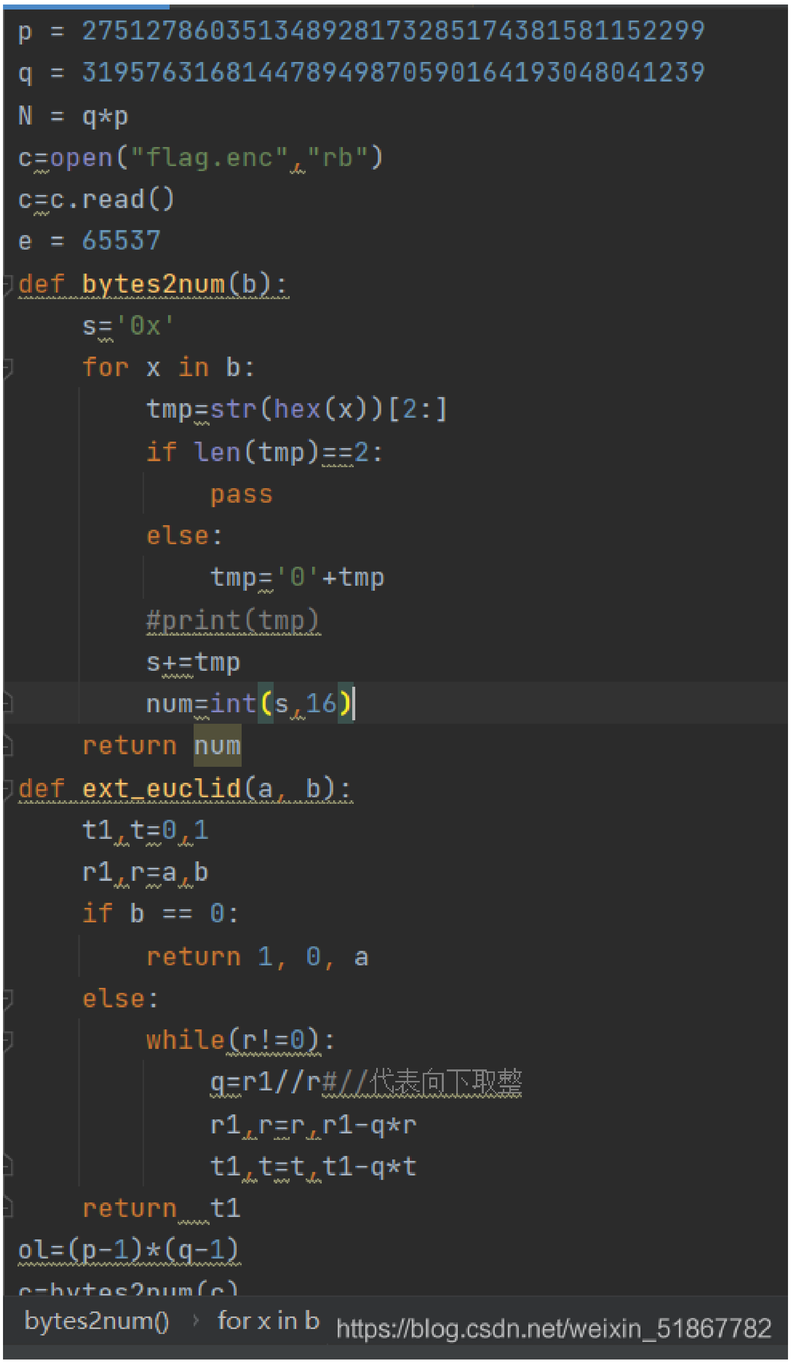
Result:

us (?)	digits	number
	77 (show)	8792434826...61<77> = 275127860351348928173285174381581152299<39> · 319576316814478949870590164193048041239<39>

More information

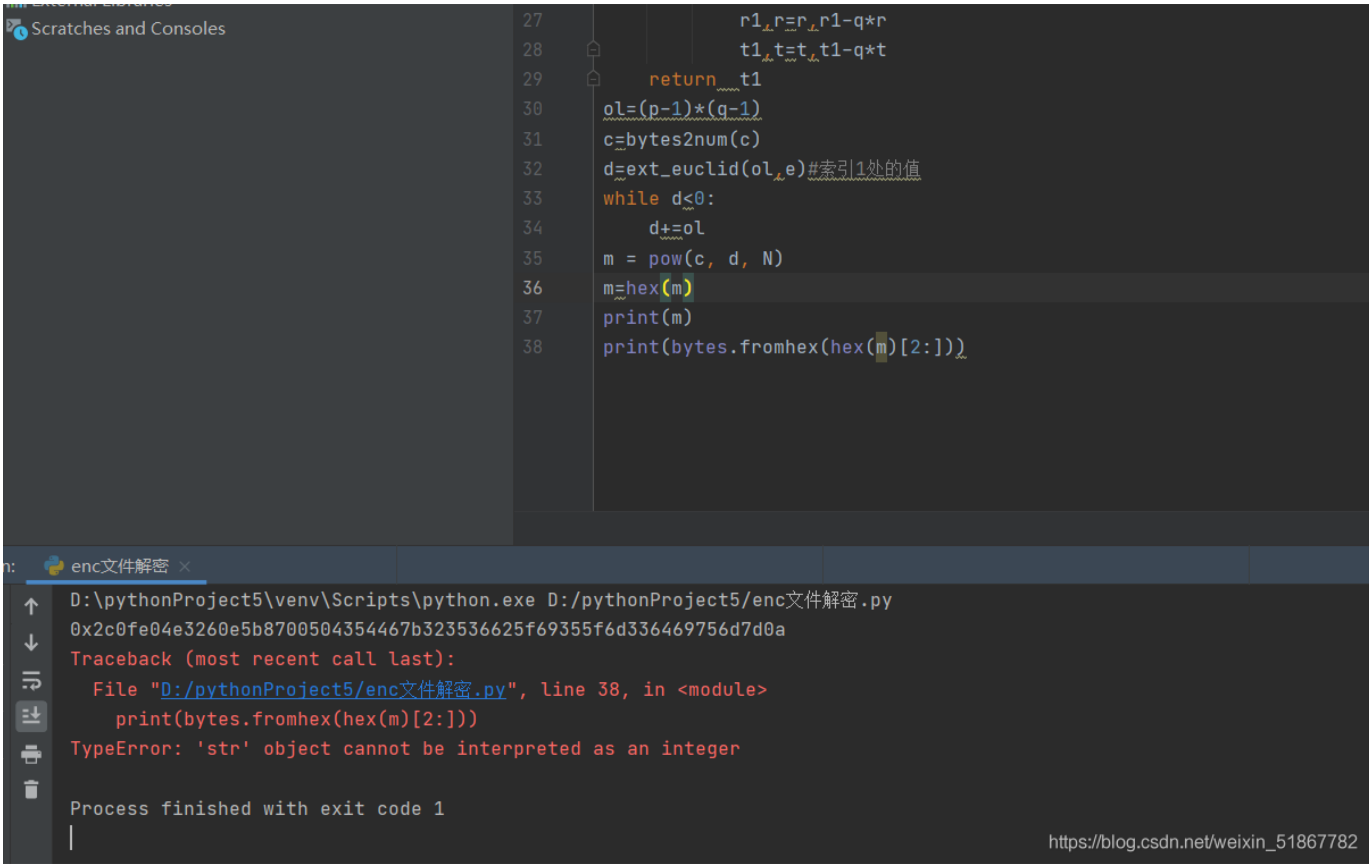
由此得到p,q.

3、在rsa解密脚本中打开文件，跑一遍脚本。



```
p = 275127860351348928173285174381581152299
q = 319576316814478949870590164193048041239
N = q*p
c=open("flag.enc","rb")
c=c.read()
e = 65537
def bytes2num(b):
    s='0x'
    for x in b:
        tmp=str(hex(x))[2:]
        if len(tmp)==2:
            pass
        else:
            tmp='0'+tmp
        #print(tmp)
        s+=tmp
        num=int(s,16)
    return num
def ext_euclid(a, b):
    t1,t=0,1
    r1,r=a,b
    if b == 0:
        return 1, 0, a
    else:
        while(r!=0):
            q=r1//r#//代表向下取整
            r1,r=r,r1-q*r
            t1,t=t,t1-q*t
    return  t1
ol=(p-1)*(q-1)
c=bytes2num(c)
```

```
q=r1//r#//代表向下取整
```

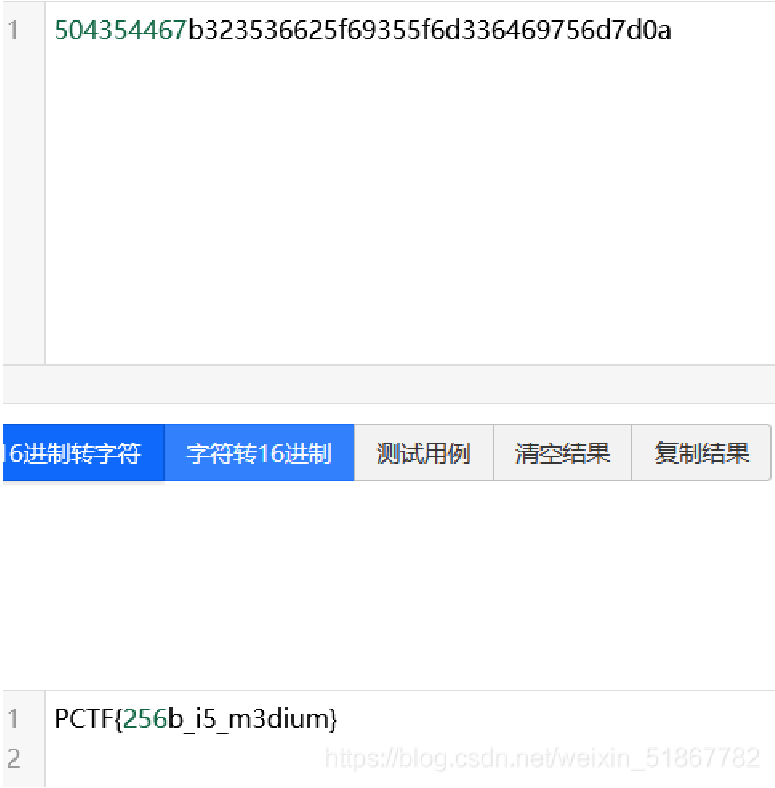


```
            r1,r=r,r1-q*r
            t1,t=t,t1-q*t
    return  t1
ol=(p-1)*(q-1)
c=bytes2num(c)
d=ext_euclid(ol,e)#索引1处的值
while d<0:
    d+=ol
m = pow(c, d, N)
m=hex(m)
print(m)
print(bytes.fromhex(hex(m)[2:]))
```

```
D:\pythonProject5\venv\Scripts\python.exe D:/pythonProject5/enc文件解密.py
0x2c0fe04e3260e5b8700504354467b323536625f69355f6d336469756d7d0a
Traceback (most recent call last):
  File "D:/pythonProject5/enc文件解密.py", line 38, in <module>
    print(bytes.fromhex(hex(m)[2:]))
TypeError: 'str' object cannot be interpreted as an integer

Process finished with exit code 1
```

发现报错，应该是打开文件时出了问题，提示在61的位置是错误的16进制。但看到输出的16进制明文后，发现7b,是'{'的十六进制。探索后发现去掉前面的几位，从5043开始便是flag.



```
504354467b323536625f69355f6d336469756d7d0a
```

```
PCTF{256b_i5_m3dium}
```