XCTF Normal_RSA

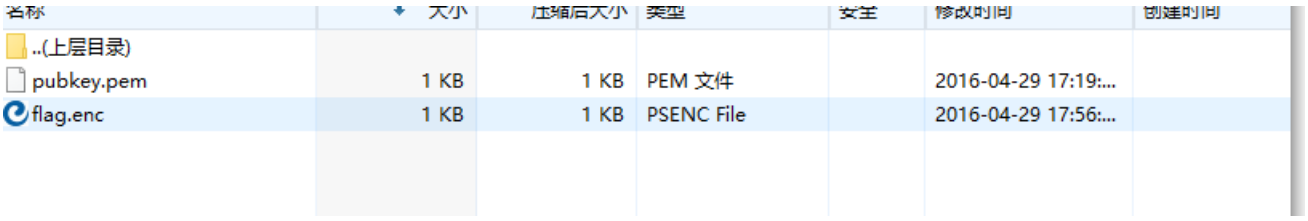
原创

YenKoc 于 2020-04-14 23:33:48 发布

分类专栏： XCTF

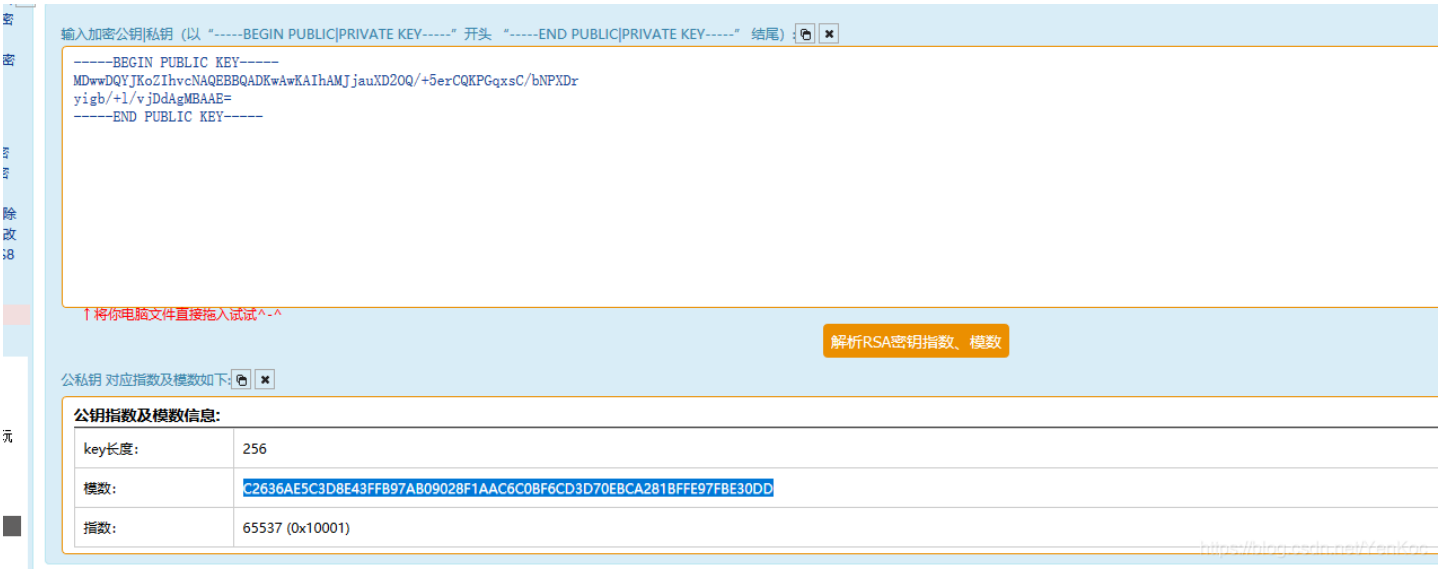
本文链接：https://blog.csdn.net/YenKoc/article/details/105524362

这题本来算是很常规的rsa了，下载附件

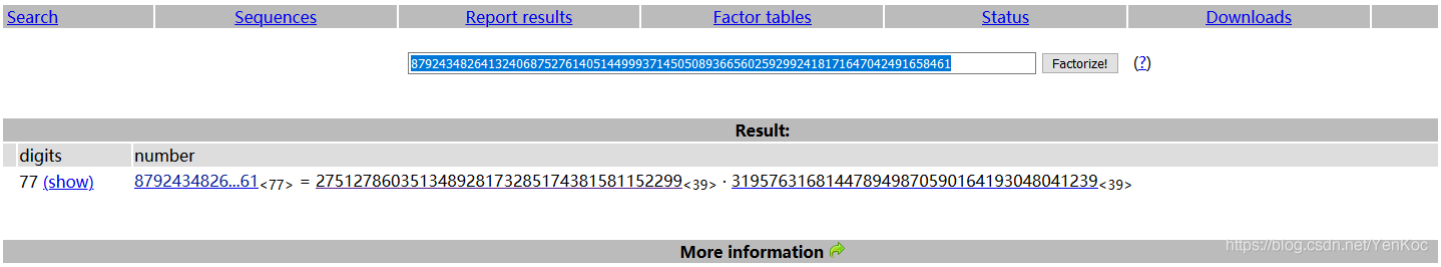


名称	大小	压缩后大小	类型	安全	修改时间	创建时间
..(上层目录)						
pubkey.pem	1 KB	1 KB	PEM 文件		2016-04-29 17:19:...	
flag.enc	1 KB	1 KB	PSENC File		2016-04-29 17:56:...	

发现有个公钥文件，还有一个加密文件，这种题之前有遇到一次，做法和这个类似，上次那个是用rsa的库，直接解的，这次直接用常规的，好像更简单，记录下模板
记事本打开那个公钥文件，放到在线网站上面解开。



n是十六进制的，拿去抓换成十进制的数字，然后分解p，q



写脚本跑就完事了

```python
import rsa
import binascii
import sys
import gmpy2
import re
from Crypto.Util import number
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
#将 10 进制数值按照 ascii 码转为字符串
def num2str(n):
    tmp=str(hex(n))[2:]
    if len(tmp)%2==0:
        pass
    else:
        tmp='0'+tmp
    s=''
    for i in range(0,len(tmp),2):
        temp=tmp[i]+tmp[i+1]
        s+=chr(int(temp,16))
    return s
r=open("flag.enc","rb")
r=r.read()
r=bytes2num(r)
e=65537
n=879243482641324068752761405144999371450508936656025929924181716470424916584 61
p=27512786035134892817328517438158115229 9
q=31957631681447894987059016419304804123 9
d=gmpy2.invert(e,(p-1)*(q-1))
print(d)
m=pow(r,int(d),n)
print(num2str(m))
```