

XCTF Normal_RSA

原创

[YenKoc](#)



于 2020-04-14 23:33:48 发布



735



收藏 4

分类专栏: [XCTF](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

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

版权



[XCTF 专栏收录该内容](#)

26 篇文章 2 订阅

订阅专栏

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。

输入加密公钥/私钥 (以 "-----BEGIN PUBLIC/PRIVATE KEY-----" 开头 "-----END PUBLIC/PRIVATE KEY-----" 结尾)

```
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauXD2OQ/+5erCQKPGqxsC/bNPXDr
y:gb/+1/vjDdAgMBAAE=
-----END PUBLIC KEY-----
```

↑ 将你电脑文件直接拖入试试 ^-^

解析RSA密钥指数、模数

公钥 对应指数及模数如下:

公钥指数及模数信息:

key长度:	256
模数:	C2636AE5C3D8E43FFB97A809028F1AAC6C08F6CD3D70EBCA281BFFE97FBE30DD
指数:	65537 (0x10001)

<https://blog.csdn.net/yenkoo>

n是十六进制的，拿去抓换成十进制的数字，然后分解p, q

Search Sequences Report results Factor tables Status Downloads

87924348264132406875276140514499937145050893665602592992418171647042491658461 Factorize! (?)

Result:

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

More information <https://blog.csdn.net/yenkoo>

写脚本跑就完事了

```

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=87924348264132406875276140514499937145050893665602592992418171647042491658461
p=275127860351348928173285174381581152299
q=319576316814478949870590164193048041239
d=gmpy2.invert(e,(p-1)*(q-1))
print(d)
m=pow(r,int(d),n)
print(num2str(m))

```