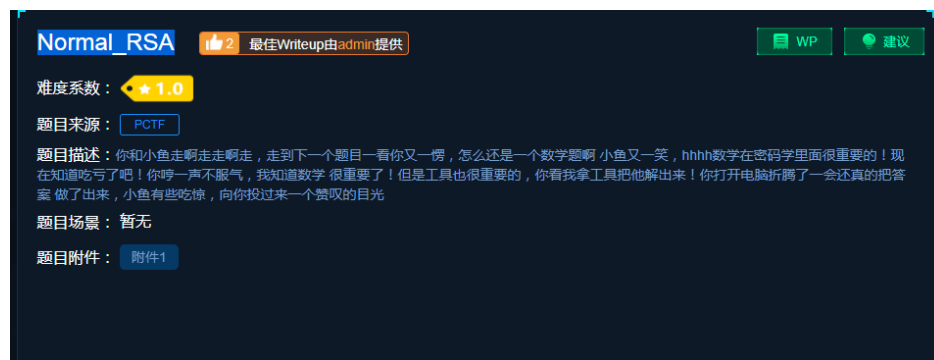
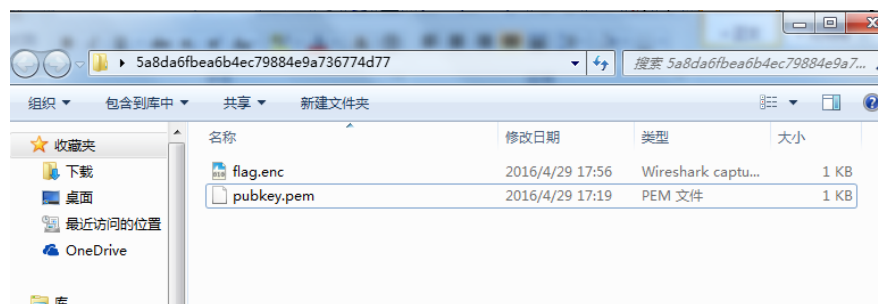


1、打开题目



下载附件看到两个文件



2、OpenSSL 使用 **PEM 文件格式存储证书和密钥**。PEM 实质上是 Base64 编码的二进制内容，再加上开始和结束行，如证书文件的

-----BEGIN CERTIFICATE-----

和

-----END CERTIFICATE-----

在这些标记外面可以有额外的信息，如编码内容的文字表示。文件是 ASCII 的，可以用任何文本编辑程序打开它们。

3、解题思路是：①使用 openssl 解密.pem 中参数 --> ②参数十六进制转换为十进制 --> ③利用 factor 对大整数进行分解，得到 p 和 q --> ④用 rsatool 生成私钥文件: private.pem --> ⑤用 private.pem 解密 flag.enc

接下来解题了

①使用 openssl 解密.pem 中参数。

Openssl 是 linux 自带的一个加密库，可以直接使用。

指令：openssl rsa -pubin -text -modulus -in warmup -in **pubkey.pem**

```

root@kali: ~/Desktop#
root@kali: ~/Desktop# cat private.pem
root@kali: ~/Desktop# openssl rsa -pubin -text -modulus -in warmup -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-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMJjauxD20Q/+5erCQKPGqxsC/bNPXD
yigb/+L/vjDdAgMBAAE=
-----END PUBLIC KEY-----
root@kali: ~/Desktop#

```

②参数十六进制转换为十进制

Python 支持直接将 16 进制转换为 10 进制

Linux 下进入 python 命令行

0x C2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD

```

-----END PUBLIC KEY-----
root@kali: ~/Desktop# python private.pem
Python 2.7.11 (default, Jan 11 2016, 21:04:40)
[GCC 5.3.1 20160101] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 0xC2636AE5C3D8E43FFB97AB09028F1AAC6C0BF6CD3D70EBCA281BFFE97FBE30DD
87924348264132406875276140514499937145050893665602592992418171647042491658461
>>>

```

③利用 factor 对大整数进行分解，得到 p 和 q

在线大整数分解网站: <http://www.factordb.com/>

Search	Sequences	Report Results	Factor Tables	Status	Comments	Login
<input type="text" value="87924348264132406875276140514499937145050893665602592992418171647042491658461"/> <input type="button" value="Factorize!"/> (2)						
Result:						
status	digits	number				
FF	77 show	8792434826...61<77> = 275127860351348928173285174381581152299<39> · 319576316814478949870590164193048041239<39>				
More information						

分解得到 p= 275127860351348928173285174381581152299

q= 319576316814478949870590164193048041239

④用 rsatool 生成私钥文件: private.pem

python rsatool.py -o private.pem -e XXX -q XXX

```
>>>
root@kali: ~/Desktop# python rsatool.py -o private.pem -e 65537 -p 27512786035134
8928173285174381581152299 -q 319576316814478949870590164193048041239
Using (p, q) to initialise RSA instance
serialatool.py dist publickey.pem
n =
c2636ae5c3d8e43ffb97ab09028f1aac6c0bf6cd3d70ebca281bffe97f3e30dd

e = 65537 (0x10001)

d =
1806799bd44ce649122b78b43060c786f8b77fb1593e0842da063ba0d8728bf1
get-pip.py rsatool.egg-info private.pem
p = 275127860351348928173285174381581152299 (0xcefbb2cf7e18a98ebcdc36e3e7c3b02b)
q = 319576316814478949870590164193048041239 (0xf06c28e91c8922b9c236e23560c09717)
Saving PEM as private.pem
root@kali: ~/Desktop#
```

⑤用 private.pem 解密 flag.enc

openssl rsautl -decrypt -in flag.enc -inkey private.pem

```
Saving PEM as private.pem
root@kali: ~/Desktop# openssl rsautl -decrypt -in flag.enc -inkey private.pem
PCTF{256b_i5_m3dium}
root@kali: ~/Desktop#
```