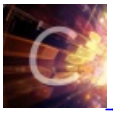
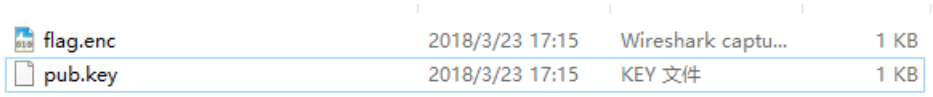
2018.04.21-安恒杯线上赛 RSA writeup

原创

乌鸦安全 于 2018-04-23 10:49:07 发布 2785 收藏 2
分类专栏： ctf 文章标签： 安恒杯 writeup RSA

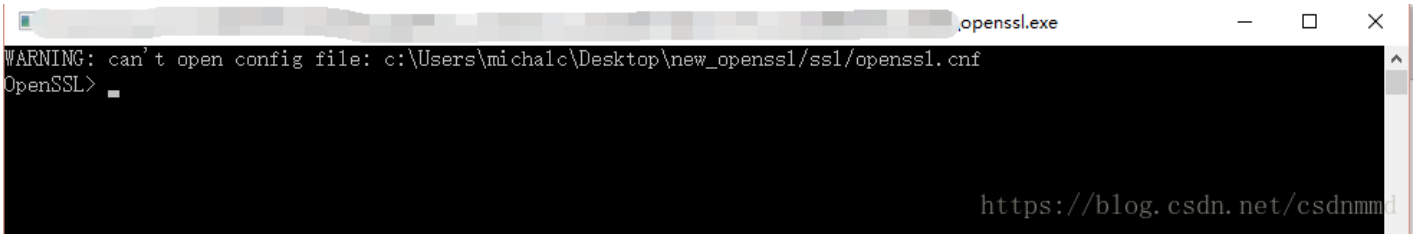




1.pub.key 打开之后就是这样的

-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMAzLFxkrkcYL2wch21CM2kQVFpY9+7+
/AvKr1rzQczdAgMBAAE=
-----END PUBLIC KEY-----

2.首先将两个文件复制到Openssl.exe所在文件目录下，然后打开软件



openssl分析私钥，执行**rsa -pubin -text -modulus -in pub.key** 命令exponent就是e值，modulus是n模数的值。

Exponent: 65537 (0x10001)
Modulus=C0332C5C64AE47182F6C1C876D42336910545A58F7EEFEFC0BCAAF5AF341CCDD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMAzLFxkrkcYL2wch21CM2kQVFpY9+7+
/AvKr1rzQczdAgMBAAE=

-----END PUBLIC KEY-----

3.然后使用msieve进行分解n值

msieve下载地址：https://sourceforge.net/projects/msieve/

>msieve153.exe 0xC0332C5C64AE47182F6C1C876D42336910545A58F7EEFEFC0BCAAF5AF341CCDD -v

记得在前面加上 0x 后面是-v

分解之后的结果如下：

```
attempting to build 36745 cycles
found 36745 cycles in 1 passes
distribution of cycle lengths:
   length 1 : 19412
   length 2 : 17333
largest cycle: 2 relations
matrix is 36372 x 36745 (5.3 MB) with weight 1106386 (30.11/col)
sparse part has weight 1106386 (30.11/col)
filtering completed in 3 passes
matrix is 25255 x 25319 (4.0 MB) with weight 847109 (33.46/col)
sparse part has weight 847109 (33.46/col)
saving the first 48 matrix rows for later
matrix includes 64 packed rows
matrix is 25207 x 25319 (2.5 MB) with weight 600451 (23.72/col)
sparse part has weight 398074 (15.72/col)
commencing Lanczos iteration
memory use: 2.5 MB
lanczos halted after 400 iterations (dim = 25207)
recovered 18 nontrivial dependencies
p39 factor: 285960468890451637935629440372639283459
p39 factor: 304008741604601924494328155975272418463
elapsed time 00:00:06
```


p39 factor: 285960468890451637935629440372639283459
p39 factor: 304008741604601924494328155975272418463

此时既知：p和q的值

p39 factor: 285960468890451637935629440372639283459 //p

p39 factor: 304008741604601924494328155975272418463 //q

4.然后在unbantu中运行以下代码：

```
#!/usr/bin/python

# coding=utf-8

#代码转自实验吧

#通过脚本，根据p，q，e值，生成私钥，貌似该脚本只能在Linux或者cygwin的python下运行。
#我就在windows试试不行，装不了能力有限，试过pip install pycrypto
#果断用Linux吧
import math
import sys
from Crypto.PublicKey import RSA
keypair=RSA.generate(1024)
keypair.p=285960468890451637935629440372639283459
keypair.q=304008741604601924494328155975272418463
keypair.e=65537  //这个值不要忘记了，不一样的
keypair.n=keypair.p*keypair.q
Qn=long((keypair.p-1)*(keypair.q-1))

i=1
while(True):
    x=(Qn*i)+1
    if(x%keypair.e==0):
        keypair.d=x/keypair.e
        break
    i+=1
private=open('private.pem','w')
private.write(keypair.exportKey())
private.close()
```

结束之后就可以得到一个private.pem的文件

```
ubuntu@ubuntu-virtual-machine:~/Desktop$ python temp.py
ubuntu@ubuntu-virtual-machine:~/Desktop$ cat private.pem
-----BEGIN RSA PRIVATE KEY-----
MIGrAgEAAiEAwDMsXGSuRxgvbByHbUIzaRBUWlj37v78C8qvWvNBzN0CAwEAAQIh
ALN4FVhA+yuPvdhp21t+kZlPHs4lbuEXXsLCvTpKeVrRAhEA1yH7pYqizM9lhi0g
n9A5AwIRAOS19DGOkbq76Neo6RaAzJ8CEGaAwKA9kBg58UufEN5UnfUCEDqhzs1Z
kFB/1cxbuA2VzV8CEQCscJ85//NGDC0ypw68xFWE
-----END RSA PRIVATE KEY-----ubuntu@ubuntu-virtual-machine:~/Desktop$
```


5.使用密钥进行解密

还是在openssl中进行操作：

命令： rsautl -decrypt -in flag.enc -inkey private.pem

```
OpenSSL> rsautl -decrypt -in flag.enc -inkey private.pem
Loading 'screen' into random state - done
flag{decrypt_256}
```

flag{decrypt_256}