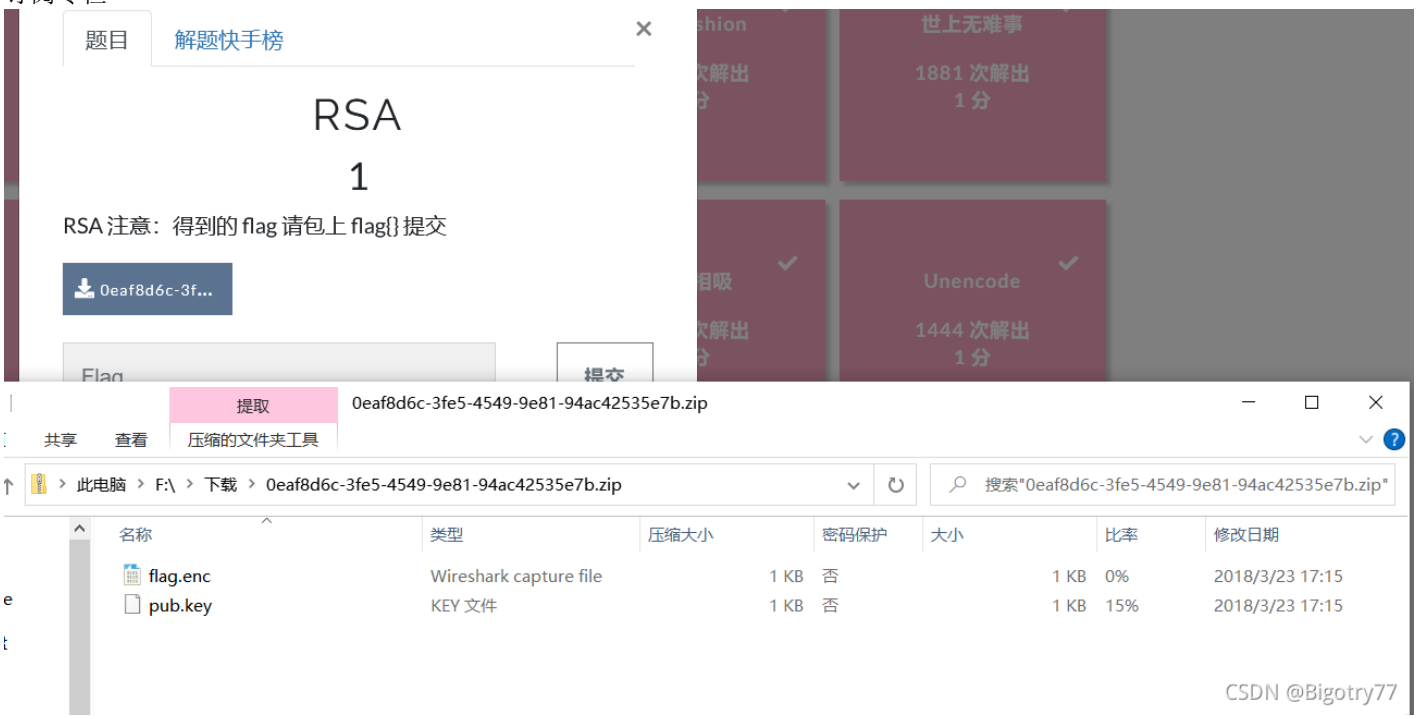
BUUCTF---RSA

原创

Bigotry77 于 2021-10-17 21:08:28 发布 71 收藏
分类专栏： ctf python学习 文章标签： python



如图，这两个文件直接打不开，可以改成.txt结尾的记事本形式，后面flag.txt内容为：

A桝YJ^

柙x秸?y[□薮?旭?縌沚

Pub.txt内容为：

-----BEGIN PUBLIC KEY-----

MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMAzLFxkrkcYL2wch21CM2kQVFpY9+7+

/AvKr1rzQczdAgMBAAE=

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

然后我去网上看思路，不知道咋得到256bit的，后来把这两个文件放到kali里面，再在终端使用openssl的命令打开之后可以得到：

这里面基本包含了解题的所有信息：其中Modulus再进行进制转换，转化成十进制之后得：
869344822960481191906660620034948005889056560172030256172166540583783221035 17

这一串数字就是我们平时用的n，再将n进行分解，可以得到两个互质的数p，q：

285960468890451637935629440372639283459

304008741604601924494328155975272418463

这两个互质的数可以通过网站解，也可以通过脚本解，都比较方便

网站：factordb.com

后面就是脚本的事情了

利用Python脚本可以得到flag：

```
import gmpy2

import rsa

p = 285960468890451637935629440372639283459
q = 304008741604601924494328155975272418463
e = 65537
n = 869344822960481191906660620034948005889056560172030256172166540583783221035 17

d = gmpy2.invert(e,(q-1)*(p-1))
print(d)

d = 81176168860169991027846870170527607562179635470395365333547868786951080991441

key = rsa.PrivateKey(n, e, d, p, q)
print(key)

with open("flag.enc","rb") as f:
 print(rsa.decrypt(f.read(), key).decode())
```

flag.enc的路径我是跟这个脚本放在了同一个路径下，所以直接open就可以了

这个跟buuctf异或加密的脚本有一点像，flag文档打开之后都是乱码，但是用Python脚本都可以解。