

BUUCTF---RSA

原创

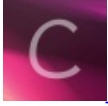
Bigotry77 于 2021-10-17 21:08:28 发布 71 收藏

分类专栏: [ctf python学习](#) 文章标签: [python](#)

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/Bigotry77/article/details/120816187>

版权



ctf同时被 2 个专栏收录

22 篇文章 1 订阅

订阅专栏



python学习

2 篇文章 0 订阅

订阅专栏

题目 解题快手榜

RSA

1

RSA 注意: 得到的 flag 请包上 flag{} 提交

0eaf8d6c-3f...

Flag 提交

名称	类型	压缩大小	密码保护	大小	比率	修改日期
flag.enc	Wireshark capture file	1 KB	否	1 KB	0%	2018/3/23 17:15
pub.key	KEY 文件	1 KB	否	1 KB	15%	2018/3/23 17:15

CSDN @Bigotry77

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櫛YJ^

柝x粘?y[菝?旭?縱泚

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：

```
(kali@kali)~[~/Desktop]
└─$ openssl rsa -pubin -text -modulus -in pub.key
RSA Public-Key: (256 bit)
Modulus:
  00:c0:33:2c:5c:64:ae:47:18:2f:6c:1c:87:6d:42:
  33:69:10:54:5a:58:f7:ee:fe:fc:0b:ca:af:5a:f3:
  41:cc:dd
Exponent: 65537 (0x10001)
Modulus=C0332C5C64AE47182F6C1C876D42336910545A58F7EEFEC0BCAAF5AF341CCDD
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhAMAzLFxkrkcYL2wch21CM2kQVFpY9+7+
/AvKr1rzQczdAgMBAAE=
-----END PUBLIC KEY-----
```

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：
86934482296048119190666062003494800588905656017203025617216654058378322103517

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 = 86934482296048119190666062003494800588905656017203025617216654058378322103517

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点像，flag文档打开之后都是乱码，但是用Python脚本都可以解。



[创作打卡挑战赛](#) >

[赢取流量/现金/CSDN周边激励大奖](#)