

CTF-I春秋第二届春秋欢乐赛-CRYPTO-RSA256

原创

he110_Z 于 2020-06-01 22:13:19 发布 367 收藏

分类专栏: [CTF python](#) 文章标签: [安全 python git](#)

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/weixin_43880435/article/details/105894458

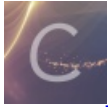
版权



CTF 同时被 2 个专栏收录

10 篇文章 0 订阅

订阅专栏



python

8 篇文章 0 订阅

订阅专栏

#CTF

试题 地址: <https://www.ichunqiu.com/battalion?t=1&r=61107>

i春秋 第二届春秋欢乐赛



分值: 100分 类型: Crypto 题目名称: rsa256

未解答

题目内容: [Download](#)

Flag:

提交

解题排名: icq7b64725h pcat cccmc

[查看writeup](#)

https://blog.csdn.net/weixin_43880435

2下载后有四个文件

encrypted.message1	2017/6/7 12:48	MESSAGE1 文件	1 KB
encrypted.message2	2017/6/7 12:48	MESSAGE2 文件	1 KB
encrypted.message3	2017/6/7 12:48	MESSAGE3 文件	1 KB
public.key	2017/6/7 11:08	注册表项	1 KB

3、text打开public.key

```
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhANmeISKWptlg38JQsrpUw5RC1gp7npMK
/0UceOxV1VXrAgMBAAE=
-----END PUBLIC KEY-----
```

4、分析密钥

方法一：key的格式是ASN.1 在线解析网站<http://lapo.it/asn1js/>

The screenshot shows the 'ASN.1 JavaScript decoder' interface. The URL in the browser is lapo.it/asn1js/#MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhANmeISKWptlg38JQsrpUw5RC1gp7npMK_0UceOxV1VXrAgMBAAE. The main content area displays the decoded ASN.1 structure:

```
SEQUENCE (2 elem)
  SEQUENCE (2 elem)
    OBJECT IDENTIFIER 1.2.840.113549.1.1.1 rsaEncryption (PKCS #1)
    NULL
  BIT STRING (1 elem)
    SEQUENCE (2 elem)
      INTEGER (256 bit) 9843207927151313098126791905614916163189282270716717785883184169952177...
      INTEGER 65537
```

Below the structure, the original key is displayed: `MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhANmeISKWptlg38JQsrpUw5RC1gp7npMK/0UceOxV1VXrAgMBAAE=`. At the bottom, there are buttons for 'with hex dump', 'decode', 'clear', and 'example', along with a '浏览...' button.

方法二：kali linux

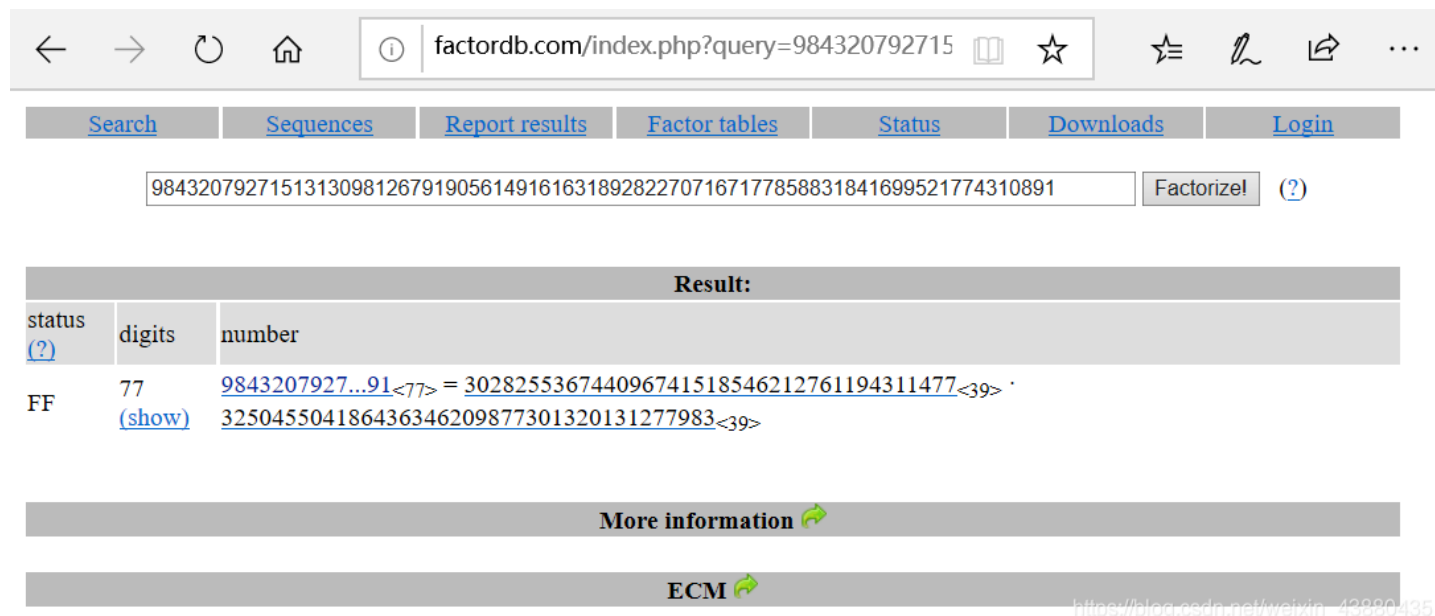
```
root@kali:/mnt/hgfs/CTF/Crypto/fujian_C74892BD664A142AF1F7F85F55754BCF/fujian# openssl rsa -pubin -text -modulus
-in ./public.key
RSA Public-Key: (256 bit)
Modulus:
 00:d9:9e:95:22:96:a6:d9:60:df:c2:50:4a:ba:54:
 5b:94:42:d6:0a:7b:9e:93:0a:ff:45:1c:78:ec:55:
 d5:55:eb
Exponent: 65537 (0x10001)
Modulus=D99E952296A6D960DFC2504ABA545B9442D60A7B9E930AFF451C78EC55D555EB
writing RSA key
-----BEGIN PUBLIC KEY-----
MDwwDQYJKoZIhvcNAQEBBQADKwAwKAIhANmeISKWptlg38JQsrpUw5RC1gp7npMK
/0UceOxV1VXrAgMBAAE=
-----END PUBLIC KEY-----
```

获取到数据如下：

(十制) $N=98432079271513130981267919056149161631892822707167177858831841699521774310891$ $E=65537$ 要想解密数据需要私钥 D ,要计算私钥 D ,则需要分解 N ,得到素数 P , Q 。

5、在线分解大数地址

<http://factordb.com/>



The screenshot shows the factordb.com website interface. At the top, there is a navigation bar with links for Search, Sequences, Report results, Factor tables, Status, Downloads, and Login. Below this is a search bar containing the number 98432079271513130981267919056149161631892822707167177858831841699521774310891 and a 'Factorize!' button. The result is displayed in a table with columns for status, digits, and number. The status is 'FF', the number of digits is 77, and the number is shown as a product of two large prime factors. Below the result, there are links for 'More information' and 'ECM'. A URL is visible in the bottom right corner: https://blog.csdn.net/weixin_43830435

Result:		
status	digits	number
FF	77	$9843207927...91_{<77>} = 302825536744096741518546212761194311477_{<39>} \cdot 325045504186436346209877301320131277983_{<39>}$

分解之后: $P=302825536744096741518546212761194311477$

$Q=325045504186436346209877301320131277983$

6、使用python解密得

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-
#python3.7
import gmpy2
import rsa
p = 302825536744096741518546212761194311477
q = 325045504186436346209877301320131277983
n = 98432079271513130981267919056149161631892822707167177858831841699521774310891
e = 65537
d = int(gmpy2.invert(e, (p-1) * (q-1)))
key = rsa.PrivateKey(n, e, d, p, q)
with open("encrypted.message1", "rb") as f1:
    str1=rsa.decrypt(f1.read(), key).decode()
with open("encrypted.message2", "rb") as f2:
    str2=rsa.decrypt(f2.read(), key).decode()
with open("encrypted.message3", "rb") as f3:
    str3=rsa.decrypt(f3.read(), key).decode()
print(str1+str2+str3)
```

得到flag{3b6d3806-4b2b-11e7-95a0-000c29d7e93d}