

# 蓝鲸安全CTF打卡题——第一期密码学

转载

[weixin\\_34162629](#) 于 2018-09-21 14:35:00 发布 748 收藏 1

文章标签: [密码学](#) [操作系统](#) [5g](#)

原文链接: <https://yq.aliyun.com/articles/649034>

版权

## 前言

渣渣一枚, 萌新一个, 会划水, 会喊六六

本文首发于先知社区: <https://xz.aliyun.com/t/2778>

再发于i春秋平台: <https://bbs.ichunqiu.com/thread-46119-1-1.html>

个人博客: <https://www.cnblogs.com/lxz-1263030049/>

再过几天就是中秋节了, 我打算尽自己最大的能力把蓝鲸安全平台上面的打卡题目的writeup整理出来。

有什么错误的地方 希望各位大佬指正(谢谢Orz)



## 一: 检查符号

## 检查符号

## 4

截取一段电波，一不小心全变成了泡泡。你能够解密吗？"oooo0.  
 000.ooo.o000.0oooo.0o.0o00.00o.00ooo.o00o.  
 0000o.0oo.0oo.o000.00oo.o000.o0o0.oooo.o0.  
 o000." 答案格式：key{flag}, flag是解密内容

 txt



## 知识点

摩斯密码、替换密码

## 解题思路

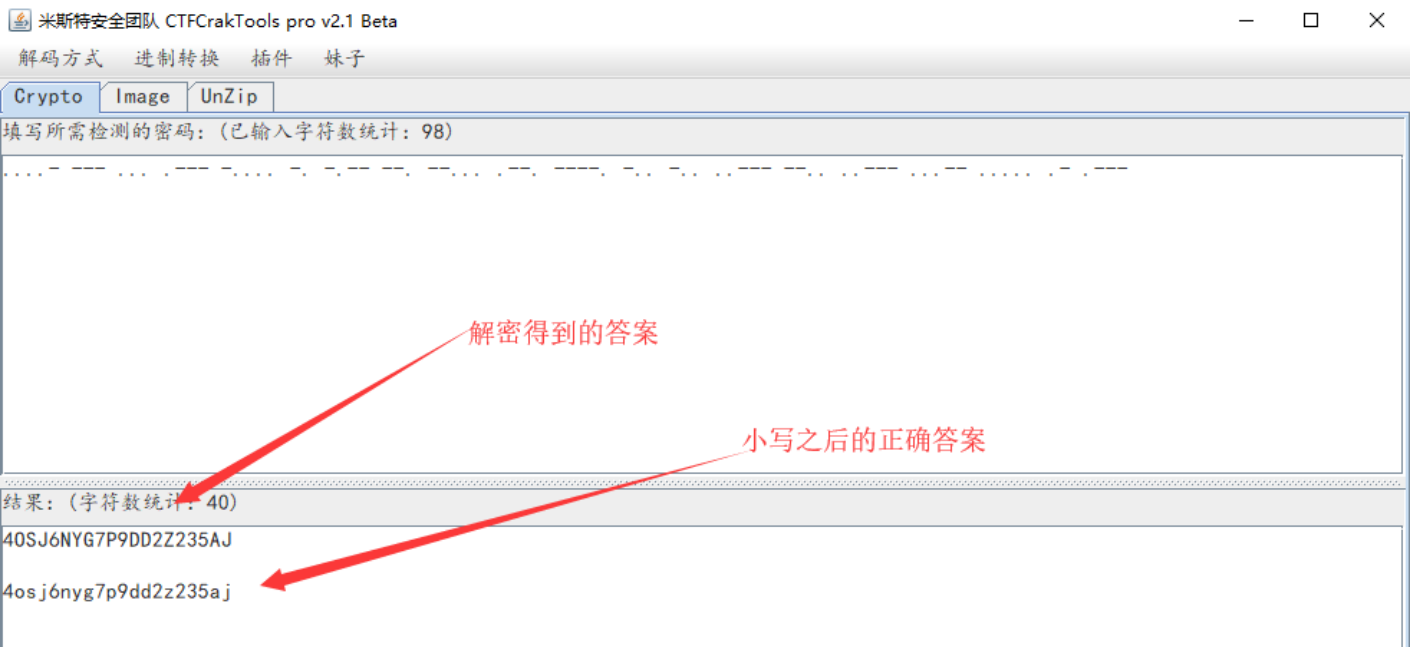
这道题很容易就可以看出是摩斯密码 摩斯密码就是由 '.' 和 '-' 组成的密码 先丢在notepad++里转换下  
 把。换成 空格

把o转换成.

把0转换成-

就会得到： .....- --- ... .--- ..... - . --- .--- .--- ..... .--- .--- .--- .---  
 .. .--- .--- ..... .- .---

放在解密工具里面就会得到：



最后得到答案: **key{4osj6nyg7p9dd2z235aj}**

## 二: 密钥生成

Challenge 139 Solves

# 密钥生成

## 4

在一次RSA密钥对生成中, 假设 $p=473398607161$ ,  $q=4511491$ ,  
 $e=17$ 求解出 $d$ 格式: key{d}

key{125631357777427553} Submit

## 知识点

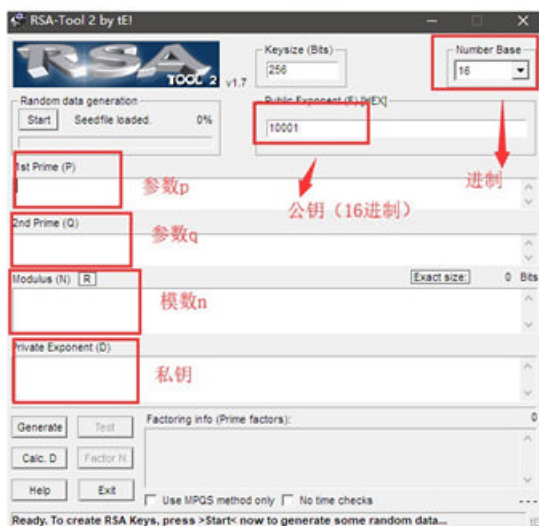
RSA密钥生成 RSA-Tool工具的使用

## 解题思路

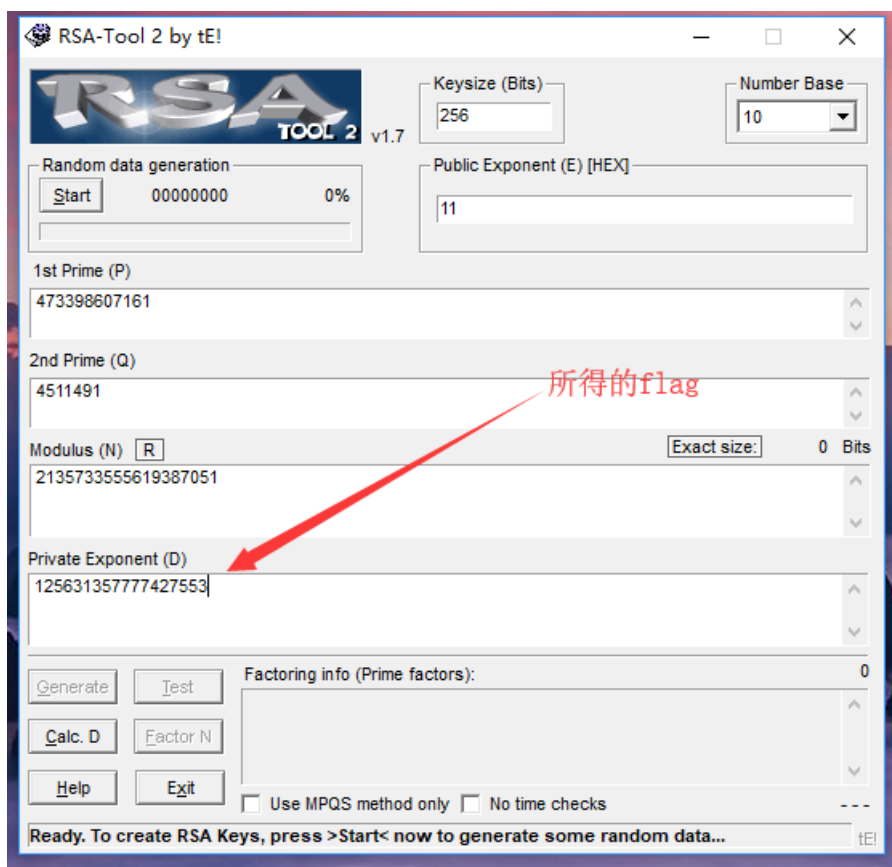
首先放一张关于RSA-Tool工具使用的方法图;

# 题目解析

- 使用RSAtool帮助我们生成密钥：
  - 先选择进制（根据题目给出数据）  
本题目选择10进制
  - 在E框中填入公钥参数
  - 在P和Q的框中对应填入大素数P和q



我们只需要根据上面的解释以及下面一些关于RSA算法的资料就可以得到：



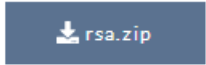
即最后答案：**key{12563135777427553}**

三：RSA解密

# RSA解密

## 4

加密方式好假，rsa直接解密吧~



key{c42bcf773d54cf03}

Submit

### 知识点

zip伪加密、RSA解密，openssl用法

### 解题思路

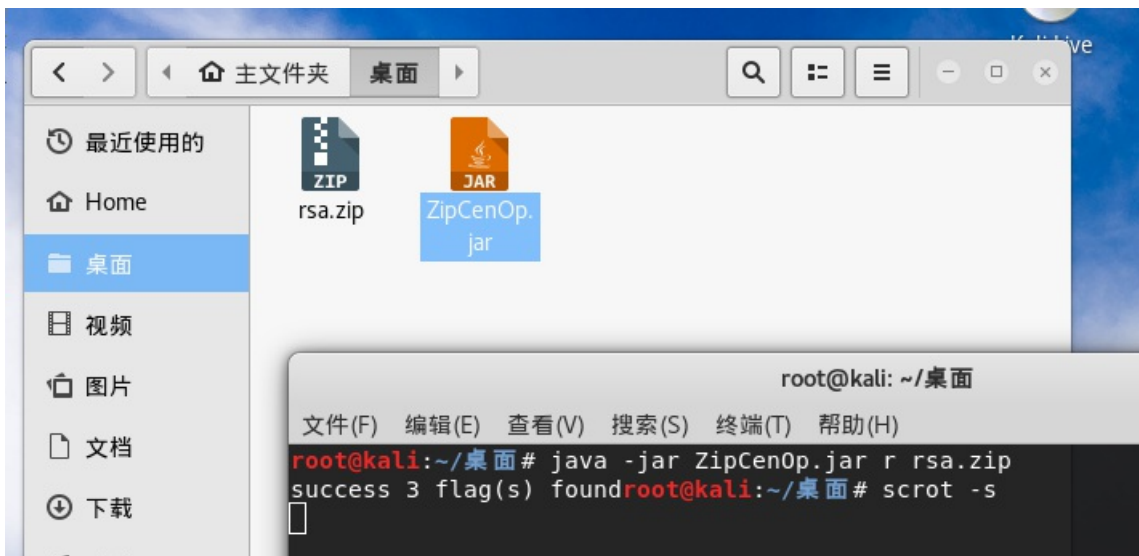
下载下来题目，告诉我们需要解压密码

这里就要涉及到伪加密的知识了

解开伪加密有多种方法，包括用7z，在linux下直接打开，更改伪加密位置处的奇数位为0(偶数)等方法，我选择使用ZipCenOp.jar这个工具来解密

```
# 在window下  
$ java -jar ZipCenOp.jar r rsa.zip
```

使用linux



两种都可以的(我个人更偏向于使用linux),打开经过解密之后的伪加密文件:

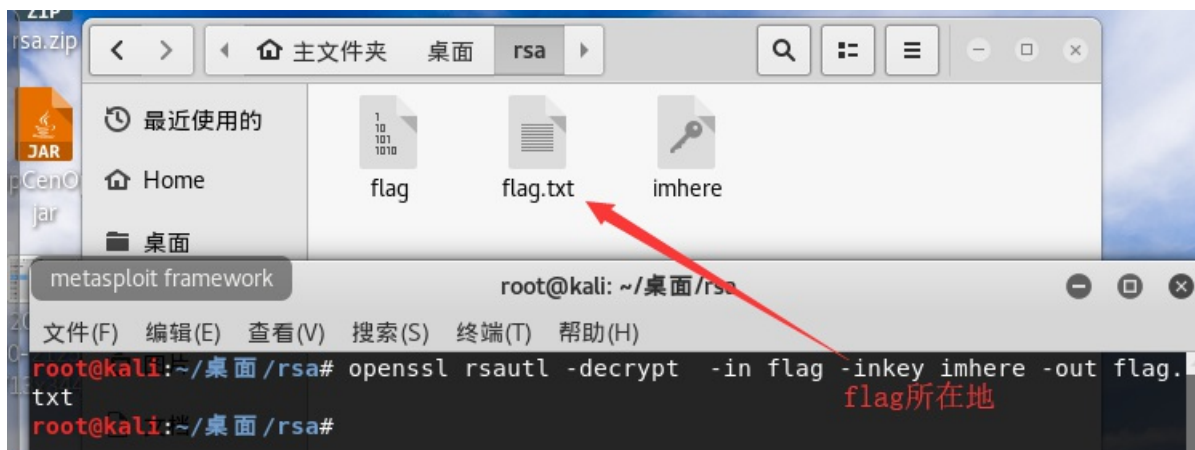
名称	大小	日期	类型	MD5
..			文件夹	
flag	128	2015/7/26 19	文件	C7CBF558
imhere	887	2015/7/13 13	文件	05E6B015

用notepad++打开imhere文件打开看到

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCw44GKtDqB1D2hTQVm9oMyp7w3C8i4tCE0zswTWtV0gVaofyF9
idBeZR3AX/crKn1X1QC6YD/pUJJRPzoKY+bk0YFT64mca7oW2fP719LFiQReirGM
fs4n9mVIYeBx1TtHfFwWBDeIHCjP33eR1hue69Dq5tZYR12e3SrUJXvRFwIDAQAB
AoGAMUQcF1KdHOv5wkweXg/4eIpJHJe2nuLkgL26P5FD9D/1r9ZAsKNkmo/VGnhf
8fDRfQoBOueFwJwJAZ8qRUSwCT+/0ZDs0xCKrQ7Yux02p9HV1sMQF74D2TVcoFZ4d
P5sjTvs0MutaoTdU0YDN0/ssqk3We2e11tr6ii3HmHtquwCQQQDezreU0IjwV30n
ndoGwYf9LsXoEZVMSI6vw/Sqiq0vagN3mufApNfj+JrZ6Lv0hHbYfaVKEUvqMyq
BKAQapNFAKEAyZ2RlugQ20fVzUkzDCSF8ByWjK4GgAqQ/qioXJ9tSPcsgV1yUem4
WM7rTBDWaoHT3N+vhAcsszQ2VJZy6vKqwJBAL2liH7CLD79Uwswgg70FfM8J7oj
1UfMDp+vFIdA4JiDjRX2JUNFTHm/9tZ6Eb+rQgXQ+Z10poUtkZ85tqCihl0CQCQI
R16MyChIRRR/LMizVPer6dkJJWff97LebFL150cxwzcwPQtet2svTDIRLiJ3BMWG
QWsq6hudCk3tNrRQOb8CQCCTcs0uWBe6k1DKWLCPEYxuTqB9xksQTmlqvdwfdCZF
BWaxHtc/ByfAisj9cfq2CY/fEoeGqLagZ5tG5G81V9VZ
-----END RSA PRIVATE KEY-----
```

看到是RSA私钥文件，说明我们要用这个私钥去解密我们的flag文件 这里就需要用到openssl文件了 kali里自带了openssl工具

使用命令：`openssl rsautl -decrypt -in flag -inkey imhere -out flag.txt`



打开flag.txt文件就会得到答案：**key{c42bcf773d54cf03}**

#### 四：公平交易

# 公平交易

## 4

vv公司称，他们给出了最为公平的游戏规则，你能猜到是什么吗？规则：FMGKYBXTSFBNCQDSPT，附件：ZKLIPOAGSUMDWFHCBVTRYENXQ. 答案的格式是key{xxxxx}，xxx为解密内容大写，所以答案是

### 知识点

playfair加密 pycipher库使用

### 解题思路

第一种方法：

使用pycipher库，就可以得到答案

由于pycipher库是python中的第三方库，所以使用的时候需要安装即：`pip install pycipher`



## 题目解析

- 题目给出两个有用内容：
- 规则：FMGKYBXTSFBNCQDSPT
- 附件：ZKLIPOAGSUMDWFHCBVTRYENXQ
  
- 附件长度为25，所以应该是密码表
- 而规则是待解密的密文

使用pycipher就可以得到答案：

```
C:\Users\fn>python
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from pycipher import Playfair
>>> Playfair('ZKLIPOAGSUMDWFHCBVTRYENXQ')
<pycipher.playfair.Playfair object at 0x0000000002D13358>
>>> Playfair('ZKLIPOAGSUMDWFHCBVTRYENXQ').decipher('FMGKYBXTSFBNCQDSPT')
'WHALECTFISVERYFAIR'
>>> _
```

密文

写入需要解密的密文

最后得到答案

第二种方法：

使用在线解密工具：<http://www.practicalcryptography.com/ciphers/classical-era/playfair/>



# JavaScript Example of the Playfair Cipher

Plaintext

```
WHALECTFISVERYFAIR
```

keysquare =

Ciphertext

```
FMGKYBXTSFBNCQDSPT
```

最后得到答案：**key{WHALECTFISVERYFAIR}**

五：填空题

Challenge

66 Solves



## 填空题

### 4

拿到一个填空题，到底该填什么？答案的格式是key{xxxxx}，所以答案是

 flag\_is\_here\_...

知识点

utf-9编码，替换密码，进制ascii码转换

## 解题思路

下载得到一个叫做flag\_is\_here\_rfc4042的文件 看到rfc4042, 知道了应该是utf-9编码了文件用python2来解, 我们需要先安装VCForPython27.msi

接下来就是需要安装UTF-9文件

```
C:\Users\Administrator>cd D:\pythonutf9-master
C:\Users\Administrator>D:
D:\pythonutf9-master>python setup.py install
running install
running bdist_egg
running egg_info
writing requirements to utf9.egg-info\requires.txt
writing utf9.egg-info\PKG-INFO
writing top-level names to utf9.egg-info\top_level.txt
writing dependency links to utf9.egg-info\dependency_links.txt
reading manifest file 'utf9.egg-info\SOURCE.txt'
writing manifest file 'utf9.egg-info\SOURCE.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\utf9
copying build\lib\utf9\__init__.py -> build\bdist.win-amd64\egg\utf9
byte-compiling build\bdist.win-amd64\egg\utf9\__init__.py to __init__.pyc
creating build\bdist.win-amd64\egg\EGG-INFO
copying utf9.egg-info\PKG-INFO -> build\bdist.win-amd64\egg\EGG-INFO
copying utf9.egg-info\SOURCE.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying utf9.egg-info\dependency_links.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying utf9.egg-info\requires.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying utf9.egg-info\top_level.txt -> build\bdist.win-amd64\egg\EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating 'dist\utf9-0.3.1-py2.7.egg' and adding 'build\bdist.win-amd64\egg' to it
removing 'build\bdist.win-amd64\egg' (and everything under it)
Processing utf9-0.3.1-py2.7.egg
Removing c:\python27\lib\site-packages\utf9-0.3.1-py2.7.egg
Copying utf9-0.3.1-py2.7.egg to c:\python27\lib\site-packages
utf9 0.3.1 is already the active version in easy-install.pth

Installed c:\python27\lib\site-packages\utf9-0.3.1-py2.7.egg
Processing dependencies for utf9==0.3.1
Searching for bitarray
Reading https://pypi.python.org/simple/bitarray/
downloading https://files.pythonhosted.org/packages/e2/1e/b93636ae36d08d0ee3aec40b08731cc97217c69db9422c0afef7ee32abd2/bitarray-0.8.3.tar.gz#sha256=050cd30b810db3aa941e7ddfbe0d8065e793012d0a88cb5739ec23624b9895e
Best match: bitarray 0.8.3
Processing bitarray-0.8.3.tar.gz
Writing c:\users\admini~1\appdata\local\temp\easy_install-ncg4f4\bitarray-0.8.3\setup.cfg
Running bitarray-0.8.3\setup.py -q bdist_egg --dist-dir c:\users\admini~1\appdata\local\temp\easy_install-ncg4f4\bitarray-0.8.3\egg-dist-temp-rqhex1
bitarray.c
bitarray.obj : warning LNK4197: export 'init_bitarray' specified multiple times; using first specification
Creating library build\temp.win-amd64-2.7\Release\bitarray_bitarray.lib and object build\temp.win-amd64-2.7\Release\bitarray_bitarray.exp
zip_safe flag not set; analyzing archive contents...
bitarray.test_bitarray: module references __file__
creating c:\python27\lib\site-packages\bitarray-0.8.3-py2.7-win-amd64.egg
Extracting bitarray-0.8.3-py2.7-win-amd64.egg to c:\python27\lib\site-packages
Adding bitarray 0.8.3 to easy-install.pth file

Installed c:\python27\lib\site-packages\bitarray-0.8.3-py2.7-win-amd64.egg
Finished processing dependencies for utf9==0.3.1

D:\pythonutf9-master>
```

UTF-9文件所在地

UTF-9文件

安装成功的提示

关于UTF-9中的问题, 我们只需要写一段python代码就可以解决了:

```
import utf9
f1 = open('flag_is_here_rfc4042', 'r')
f2 = open('flag.txt', 'w')
str1 = f1.read()
print utf9.utf9decode(str1)
f2.write(utf9.utf9decode(str1))
```

我们采用的是直接读取flag\_is\_here\_rfc4042文件的方法是为了避免复制粘贴时格式出现问题, 下面就是具体的操作方法:

```
C:\Users\Administrator>cd D:\python\工具脚本
C:\Users\Administrator>D:
D:\python\工具脚本>python utf9de.py
*((//+ + - % )**((% (-) )+ + (% + + + % + - ( // ( % ) ) ) ) ) ) ) ) +
*((// / )+ % + - ( // ) ** ( * ( + )+ + + % )+ + * (( // +
% )+ ( - ) ** ( ( + )+ - ( // ) )+ * (( + - ( // - % ) ) **
( + + )+ * ( + - ( // - % ) ) ** ( - + )+ ( + ) ** ( %
% + + )+ ( - ) * ( // - % )+ * ( - // + % )+ (
+ ( % ) * + ) ** + * (( % ) * + - ( // ) ** )+ (
/ ) * (( - + ) * ( + ) ** + * ( ( + - ) ** + -
% ) * ( - + ) / + ( % ) ** + ( // ) * (( % % + + ) % )+ - ) ** +
* ( // ( % )+ ) * ( ( % ) * + + // + + /
```

使用python代码解决utf-9问题

我们先来分析一下关于解密出来的内容:



Challenge

76 Solves

X

# RSA破解

## 4

得到了公钥，怎么才能解密呢？tip分解n，答案格式whtalectf{flag}

↓ RSA.zip

whtalectf{256\_n\_get}

Submit

### 知识点

RSA模数分解，RSA解密 openssl使用方法

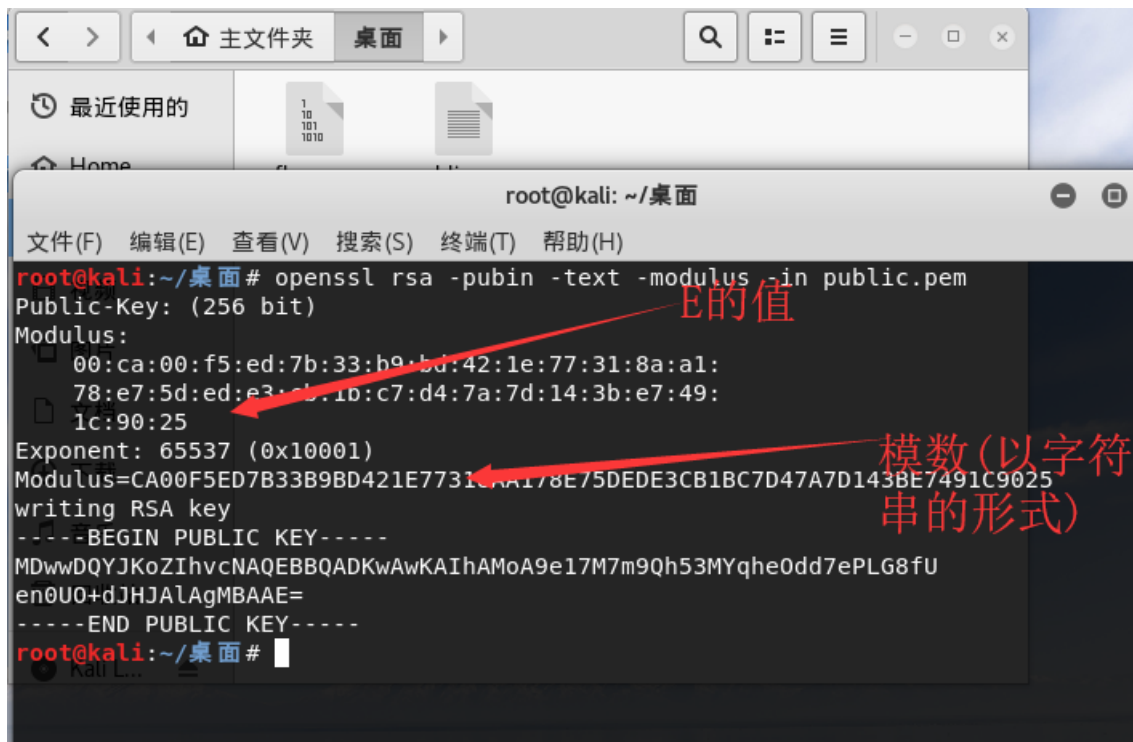
### 解题思路

下载并打开压缩包文件就会发现：

这是一个加密的flag文件和公钥文件 我们先通过openssl来分析一下公钥是否可以被攻击

使用linux，其中的命令是：

```
openssl rsa -pubin -text -modulus -in public.pem
```



我们接着使用msieve就可以了

```
C:\Users\Administrator>cd H:\题目\蓝鲸安全打卡\密码学\工具\msieve153_win64\msieve153
C:\Users\Administrator>H:
H:\题目\蓝鲸安全打卡\密码学\工具\msieve153_win64\msieve153>msieve153.exe 0xCA00F5ED7B33B9BD421E77318AA178E75DEDE3CB1BC7D47A7D143BE7491C9025 -v

Msieve v. 1.53 (SVN 1005)
Fri Sep 21 11:44:43 2018
random seeds: e7065bb0 6e00cf74
factoring 91368892744156824784111061767736072407911145707607105701466887321431798747173 (17 digits)
searching for 15-digit factors
commencing quadratic sieve (77-digit input)
using multiplier of 13
using generic 32kb sieve core
sieve interval: 12 blocks of size 32768
processing polynomials in batches of 17
using a sieve bound of 918109 (36471 primes)
using large prime bound of 91810900 (26 bits)
using trial factoring cutoff of 26 bits
polynomial 'A' values have 10 factors

sieving in progress (press Ctrl-C to pause)
36673 relations (19440 full + 17233 combined from 190873 partial), need 36567
36673 relations (19440 full + 17233 combined from 190873 partial), need 36567
sieving complete, commencing postprocessing
begin with 210313 relations
reduce to 51839 relations in 2 passes
attempting to read 51839 relations
recovered 51839 relations
recovered 40290 polynomials
attempting to build 36673 cycles
found 36673 cycles in 1 passes
distribution of cycle lengths:
length 1 : 19440
length 2 : 17233
largest cycle: 2 relations
matrix is 36471 x 36673 (5.3 MB) with weight 1096867 (29.91/col)
sparse part has weight 1096867 (29.91/col)
filtering completed in 3 passes
matrix is 25290 x 25354 (4.0 MB) with weight 847611 (33.43/col)
sparse part has weight 847611 (33.43/col)
saving the first 48 matrix rows for later
matrix includes 64 packed rows
matrix is 25242 x 25354 (2.5 MB) with weight 605823 (23.89/col)
sparse part has weight 414867 (16.36/col)
commencing Lanczos iteration
memory use: 2.6 MB
lanczos halted after 401 iterations (dim = 25240)
recovered 16 nontrivial dependencies
p39 factor: 290579950064240059571837821251441436997
p39 factor: 314436328879392457343835667929324128609
elapsed time 00:01:45

H:\题目\蓝鲸安全打卡\密码学\工具\msieve153_win64\msieve153>
```

打开msieve153.exe所在文件夹

加上刚刚解析出来的模数(以十六进制表示)

解出来P Q

就会得到相应的P Q

我们需要使用脚本生成使用文件

```

import math

import sys

from Crypto.PublicKey import RSA
keypair = RSA.generate(1024)

keypair.p = 290579950064240059571837821251441436997

keypair.q = 314436328879392457343835667929324128609

keypair.e = 65537

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()

```

使用linux运行脚本很简单 进入文件直接使用命令：

```
python prikeygen.py
```

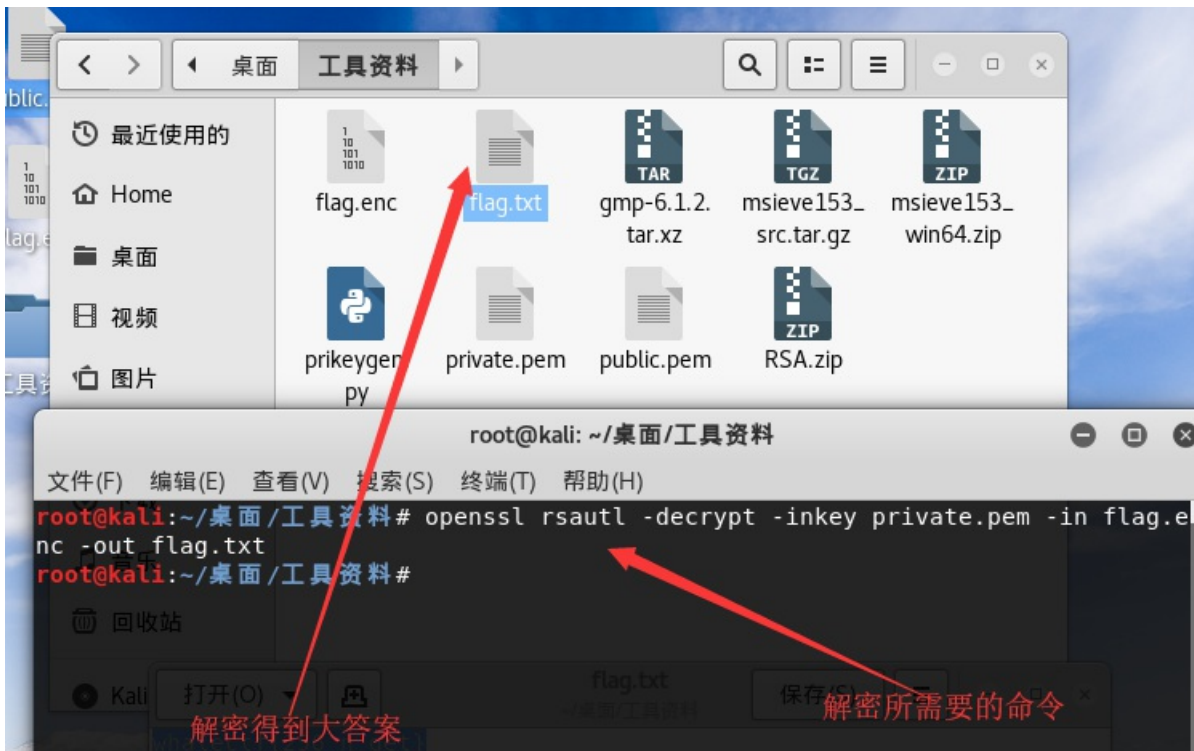
就会生成相应的私钥

```

-----BEGIN RSA PRIVATE KEY-----
MIGpAgEAAiEAYgD17Xszub1CHncxiqF4513t48sbx9R6fRQ750kckCUCAwEAAQIg
XbBsX6TQrMj2raKiu6WAqRtv86ps61bgXsftak7iqgECEQDam6mnJeZJM1o5QLZY
XK1FAhEA7I478Iv1HnDUJ/xGsCkJYQIQFvqv7bMNLvqn7Ebt3qH25QIQFwZ1KS3G
Rxc+X0H782ubIQIQL2kzJ1i81uHD5eQ1bYjXgg==
-----END RSA PRIVATE KEY-----

```

接下来就是需要使用openssl命令了



就会得到答案：`whalectf{256_n_get}`

参考资料：

RSA算法原理（一）：[http://www.ruanyifeng.com/blog/2013/06/rsa\\_algorithm\\_part\\_one.html](http://www.ruanyifeng.com/blog/2013/06/rsa_algorithm_part_one.html)

RSA算法原理（二）：[http://www.ruanyifeng.com/blog/2013/07/rsa\\_algorithm\\_part\\_two.html](http://www.ruanyifeng.com/blog/2013/07/rsa_algorithm_part_two.html)

CTF中那些脑洞大开的编码和加密：<https://www.tuicool.com/articles/2E3INm>

UTF-9: <https://github.com/enricobacis/utf9>

CTF中RSA的常见攻击方法：<https://www.anquanke.com/post/id/84632>

您可以考虑给博主来个小小的打赏以资鼓励，您的肯定将是我最大的动力。



作者：[落花四月](#)

出处：<https://www.cnblogs.com/lxz-1263030049/>

关于作者：潜心于网络安全学习。如有问题或建议，请多多赐教！

版权声明：本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接。

特此声明：所有评论和私信都会在第一时间回复。也欢迎园子的大大们指正错误，共同进步。或者直接私信我

声援博主：如果您觉得文章对您有帮助，可以点击文章右下角【推荐】一下。您的鼓励是作者坚持原创和持续写作的最大动力！