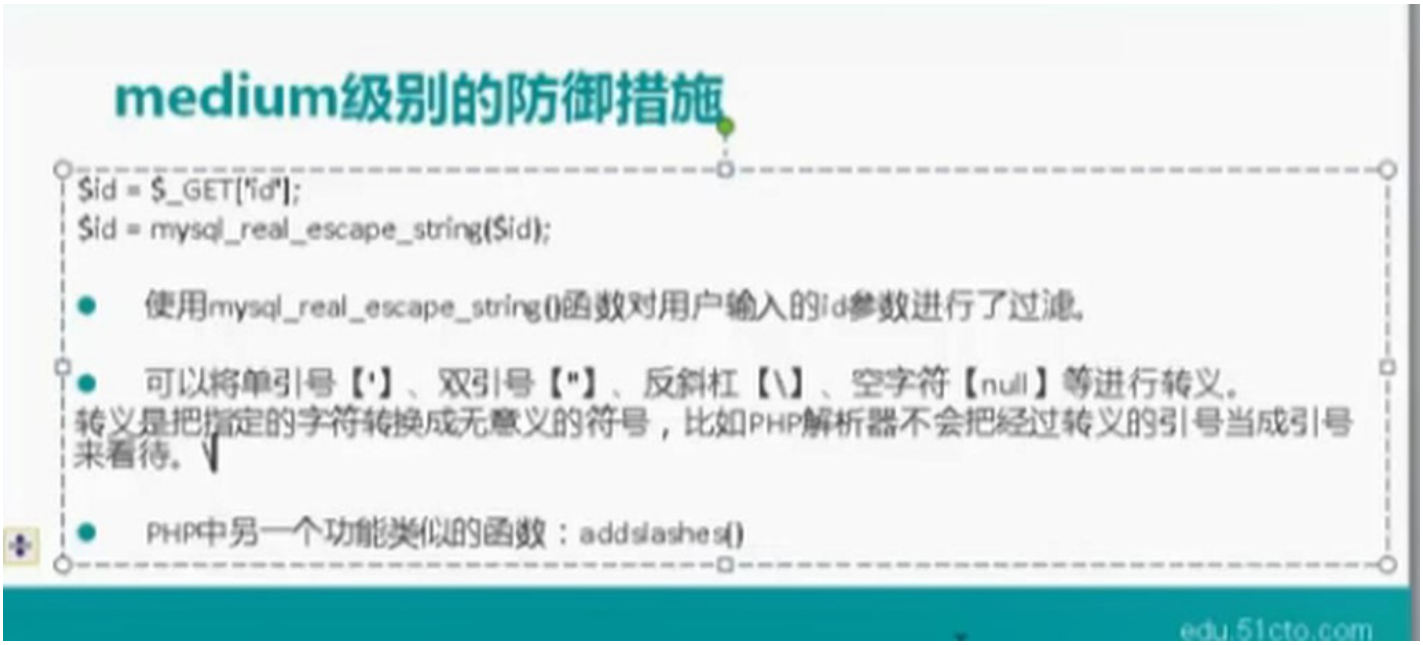
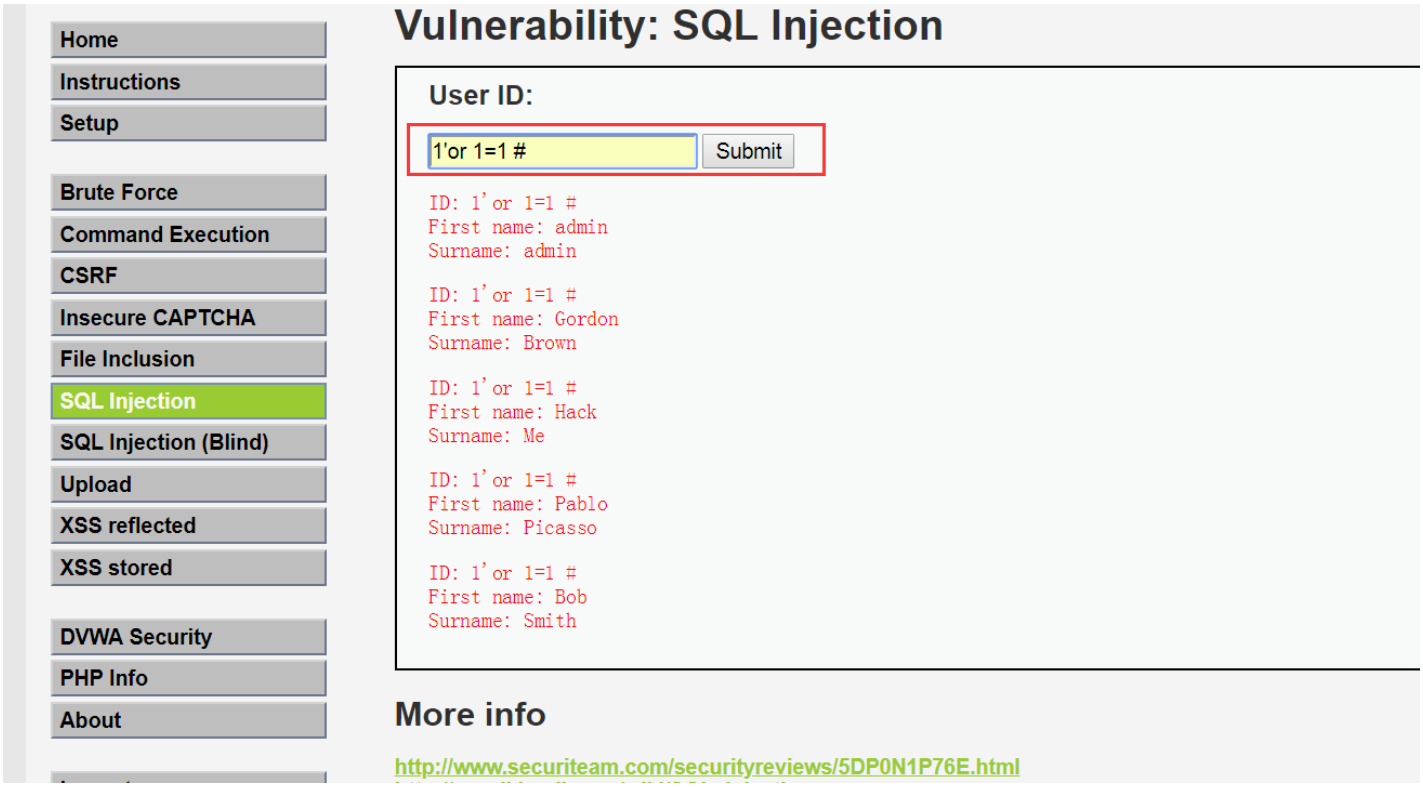


怎样防护SQL注入,用户的一切输入都是有害的----》过滤:

低等级下:



调整为中级之后 就不行了:

查看源代码:

view source:

再次点击 compare:

低级:

```
1 <?php
2
3 if(isset($_GET['Submit'])){
```

```
    // Retrieve data

    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i = 0;

    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

中级:

```
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = mysql_real_escape_string($id);

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";

    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

    $num = mysql_numrows($result);

    $i=0;
```

```
    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
?>
```

高级:

```
<?php

if (isset($_GET['Submit'])) {

    // Retrieve data

    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($id)){

        $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
        $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );

        $num = mysql_numrows($result);

        $i=0;

        while ($i < $num) {

            $first = mysql_result($result,$i,"first_name");
            $last = mysql_result($result,$i,"last_name");

            echo '<pre>';
            echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
            echo '</pre>';
```

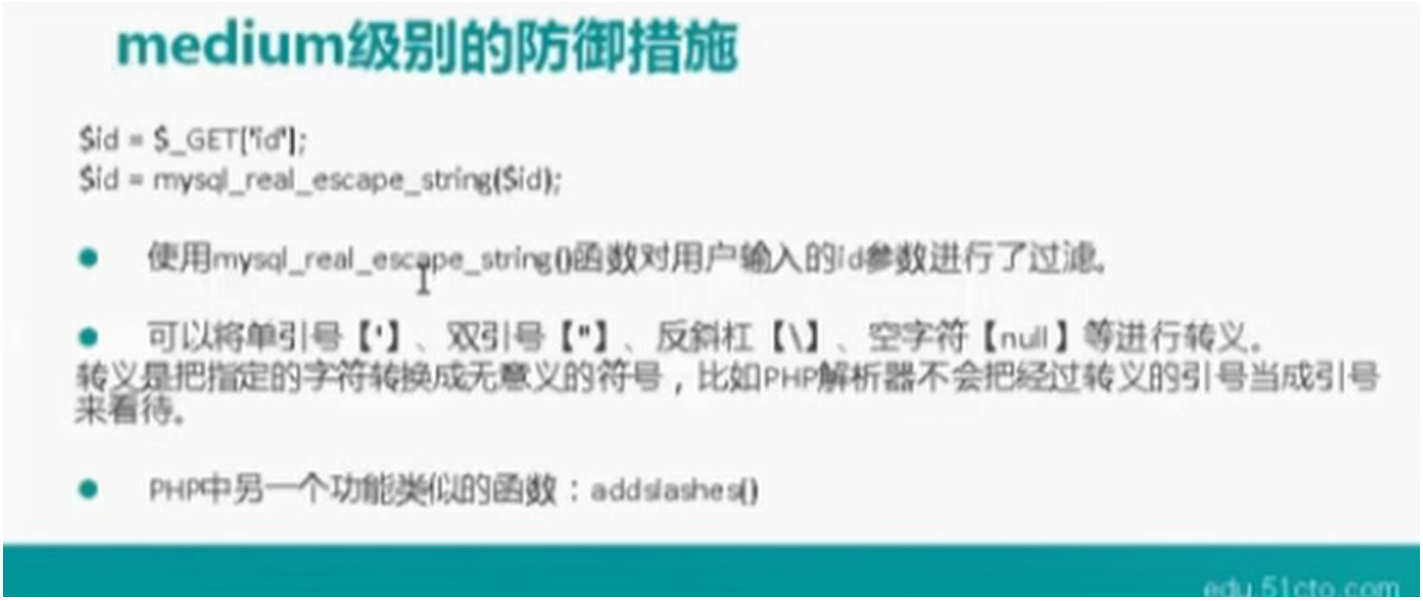
```
28
29              $i++;
30          }
31      }
32  }
33  ?>
```

对比低级和中级的,发现其 区别:
中级里面多了下面一句:

```
1  $id = mysql_real_escape_string($id);
2
```

低级里面直接获取id,而后可以构造查询语句,赋给 $getid变量,下面主要讲上面一句:

```
1  $id = mysql_real_escape_string($id);
2
```



`mysql_real_escape_string` 主要对用户输入的数据进行转义，（一般有单引号,双引号,【\】；【null】），一般转义的话就是在这些符号前面加一个【\】符号;举个例子:
原来有个用户名字就叫

```
1  adm`in
```

在输入用户名的时候不能直接输入

```
```

```
adm`in
```

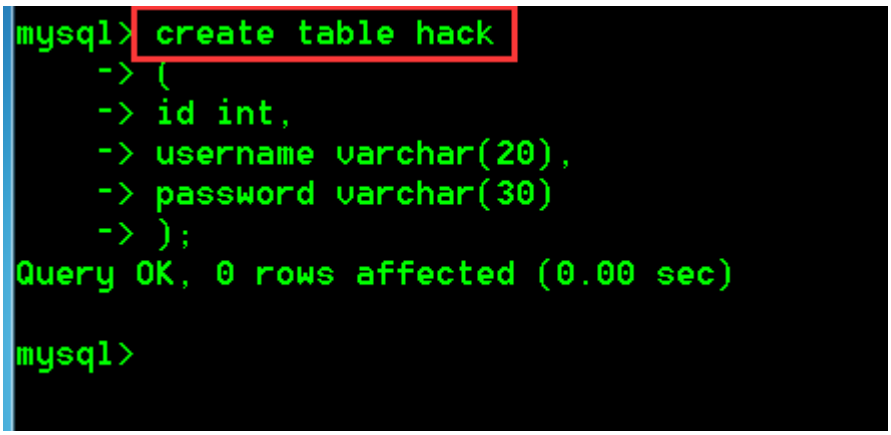
因为他会把这里的单引号和查询语句里面的单引号构成一对闭合了,所以要在这个单引号之前加一个反斜杠符号【\】进行转义,转义就是你这个单引号不要和其他的单引号组成一对进行闭合,它就是被看成一个字符功能,它就是一个单引号!

举例:

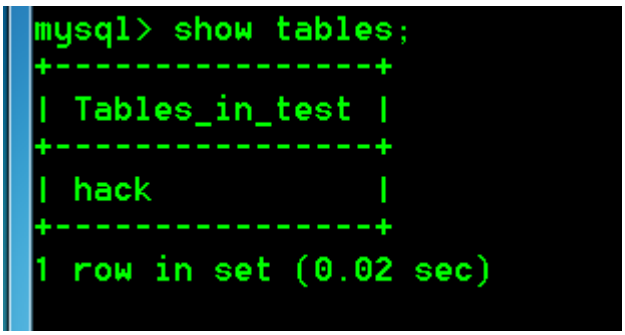
在mysql里面输入:

1.show databases;

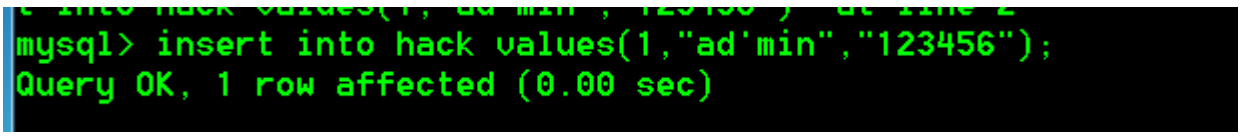
2.create table hack



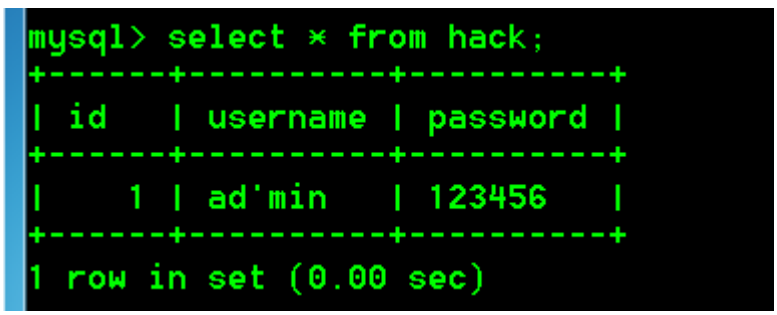
3.show tables;



4_. insert into hack values(1,"adm'in","123");



5.select * from hack;



而后输入:

select * from hack where username ='admin';

这样是不行的!

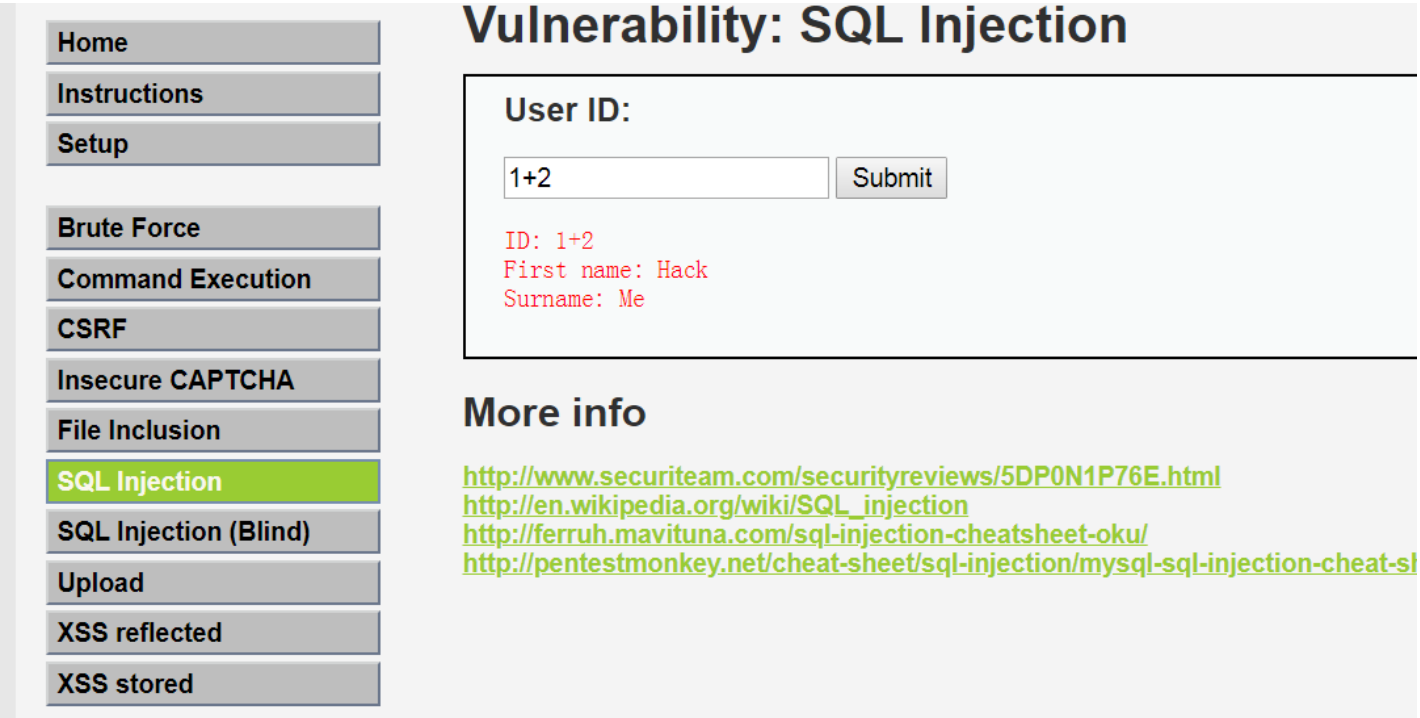
```
mysql> select * from hack where uaername='adm'in';
    '>
```

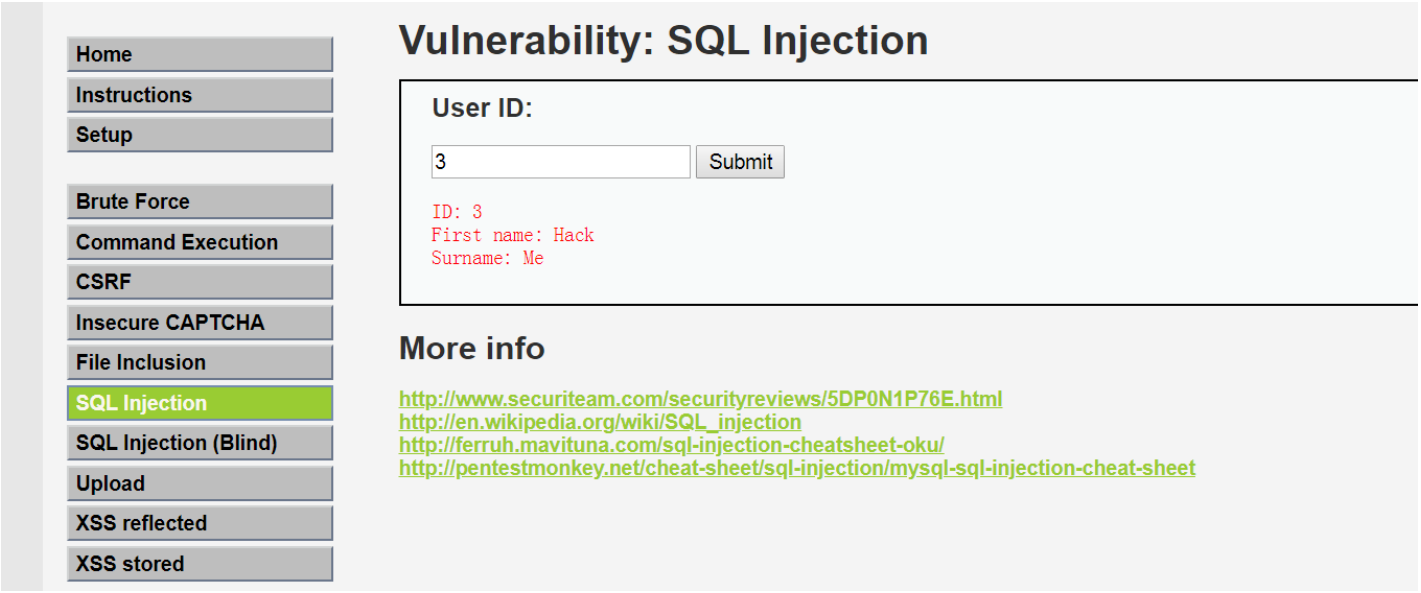
select * from hack where username ='ad\'min';

```
mysql> select * from hack where username='ad\'min';
+------+----------+----------+
| id   | username | password |
+------+----------+----------+
|    1 | ad'min   | 123456   |
+------+----------+----------+
1 row in set (0.00 sec)
```

只要你输入的语句里面有这些四个符号的话(（一般有单引号,双引号,【\】； 【null】),只要在其前面加一个【\】,就可以转义了!还有一个函数的功能和mysql_real_escape_string差不多！是addslashes()【slash就是反斜杠的意思】

还有一个区别:





看中级里面这句话:

```
$getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";
```

这里有一个数字型注入，`$id` 没有放在单引号里面，这里就不用考虑单引号问题，如下图