

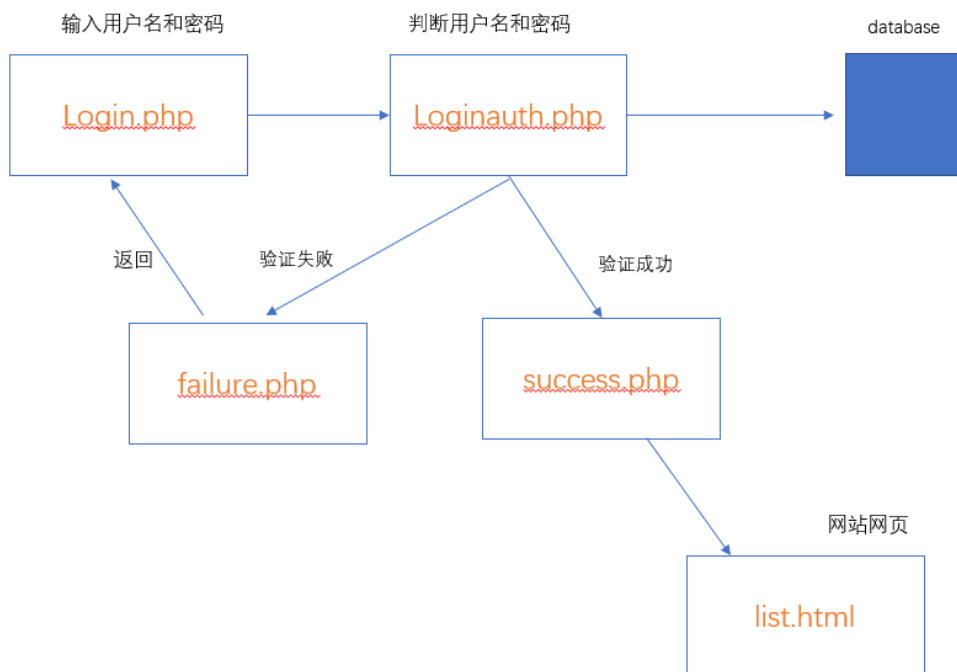
login.php

loginAuth.php:

success.php

list.html

sql注入的密码绕过渗透测试



login.php

login.php

```
1 <html>
2
3 <head>
4 <title>Login Page</title>
5 <meta http-equiv="content-Type" content="text/html; charset=utf-8"/>
6 </head>
7
8 <body>
9 <h1>User Login</h1>
10
11 <form action="loginAuth.php" method="post">
12   Username:<input type="text" name="usernm"/><br />
13   Password:<input type="password" name="passwd"/><br />
14   <input type="submit" value="Submit"/>&nbsp;&nbsp;&nbsp;<input type="reset" value="Reset"/>
15 </form>
16
17 </body>
18 </html>
```

sd4346221正在学习

进入2003这个服务器里面,找到login.php这个网页;

找到提交的变量名:

name = usernm

name = passwd

登录验证的页面:

loginAuth.php:

loginAuth.php

```
1 <?php
2 $username=$_REQUEST['username'];
3 $password=$_REQUEST['passwd'];
4
5 $conn=mssql_connect('127.0.0.1','sa','123');
6 if(!$conn){
7     exit("DB Connect Failure!");
8 }
9 mssql_select_db("users",$conn) or exit("DB Select Failure!");
10
11 $sql="select * from users where username='$username' and password='$password'";
12 $res=mssql_query($sql,$conn) or exit("DB Query Failure!");
13 if (mssql_num_rows($res)!=0){
14     header("location:succcess.php");
15 }else{
16     header("location:failure.php");
17 }
18 ?>
```

首先是连接数据库;

然后判断是否连接的上,如果连接不上,就退出;

loginAuth.php

```
1 <?php
2 $username=$_REQUEST['username'];
3 $password=$_REQUEST['passwd'];
4
5 $conn=mssql_connect('127.0.0.1','sa','123');
6 if(!$conn){
7     exit("DB Connect Failure!");
8 }
9 mssql select db("users",$conn) or exit("DB Select Failure!");
10
11 $sql="select * from users where username='$username' and password='$password'";
12 $res=mssql_query($sql,$conn) or exit("DB Query Failure!");
13 if (mssql_num_rows($res)!=0){
14     header("location:succcess.php");
15 }else{
16     header("location:failure.php");
17 }
18 ?>
```

edu.51cto.com

打开数据库;用or连接,如果前边这一句话执行的不成功,就执行后边这一句话;

下面一句把查询语句复制给了sql变量;查询里面我们输入的变量都放在一对单引号里面,这里极有可能是文本型注入;

下面一句使用mysql_query进行查询;如果执行不成功就执行exit; **mysql_query**这个函数查询的结果是个结果集!

必须通过各种方法获得结果集里面的数据;

mysql_num_rows这个方法是用来判断这个结果集里面有几行数据;结果不为0,就证明结果集里面有数据,

head是一个跳转,通过这个语句, 可以把当前网页跳转至另外一个网页上去! (**Location**指明了你要跳转的那个网页上去!)

success.php

success.php

```
1 <html>
2     Login Success!<br />
3     <a href=list.html>Enter the Web Site!</a>
4 </html>
```

这个里面就是一个超链接

list.html

list.html

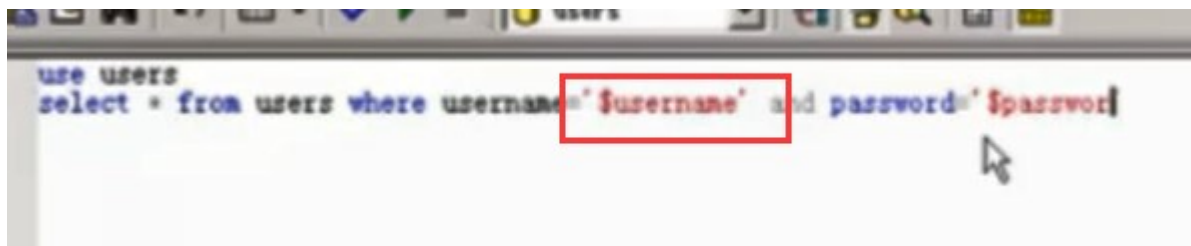
```
1 <html>
2 <head>
3     <title> List </title>
4     <meta http-equiv="content-Type" content="text/html; charset=utf-8">
5 </head>
6
7 <body>
8     <a href='query.html'>Employee Infromation Query</a><br />
9     <a href='MessageBoard.php'>Employee Message Board</a><br />
10    <a href='ShoppingHall.php'>Shopping Hall</a><br />
11    <a href='DisplayDirectory.php'>Display Directory</a><br />
12    <a href='FileSharing.php'>File Sharing</a><br />
13    <a href='DisplayFile.php'>Display Uploaded's File Content</a><br />
14    <br /><br /><br /><a href='index.php'>Go Back to Index</a><br />
15 </body>
16
17 </html>
```

list.html

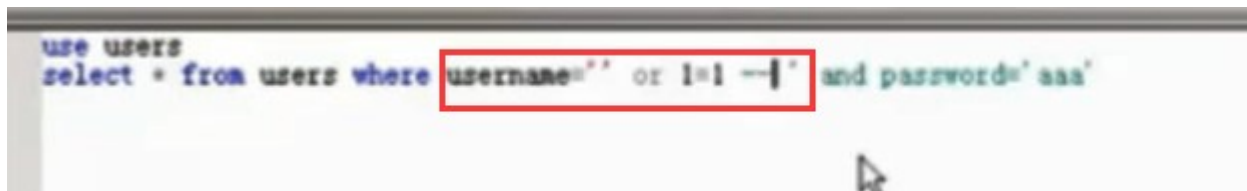
```
1 <html>
2 <head>
3     <title> List </title>
4     <meta http-equiv="content-Type" content="text/html; charset=utf-8">
5 </head>
6
7 <body>
8     <a href='query.html'>Employee Infromation Query</a><br />
9     <a href='MessageBoard.php'>Employee Message Board</a><br />
10    <a href='ShoppingHall.php'>Shopping Hall</a><br />
11    <a href='DisplayDirectory.php'>Display Directory</a><br />
12    <a href='FileSharing.php'>File Sharing</a><br />
13    <a href='DisplayFile.php'>Display Uploaded's File Content</a><br />
14    <br /><br /><br /><a href='index.php'>Go Back to Index</a><br />
15 </body>
16
17 </html>
```

主要做了一些超链接;

sql注入的密码绕过渗透测试

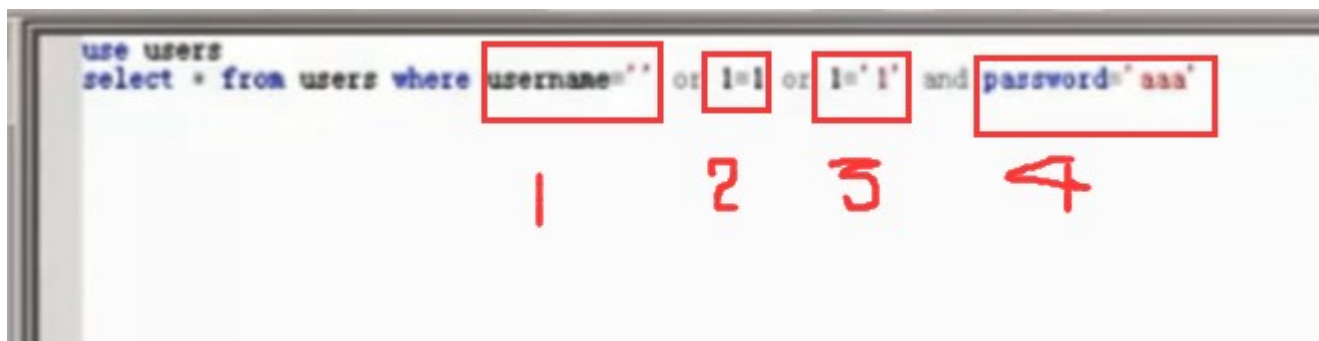


主要是闭合前面的单引号;



`user = ''` 这是一个假值, `or 1=1` 这是一个真值, 然后注释符号 `--` 闭合了单引号, 同事后边的 `and` 语句也没有起到作用;

另外一种方法:



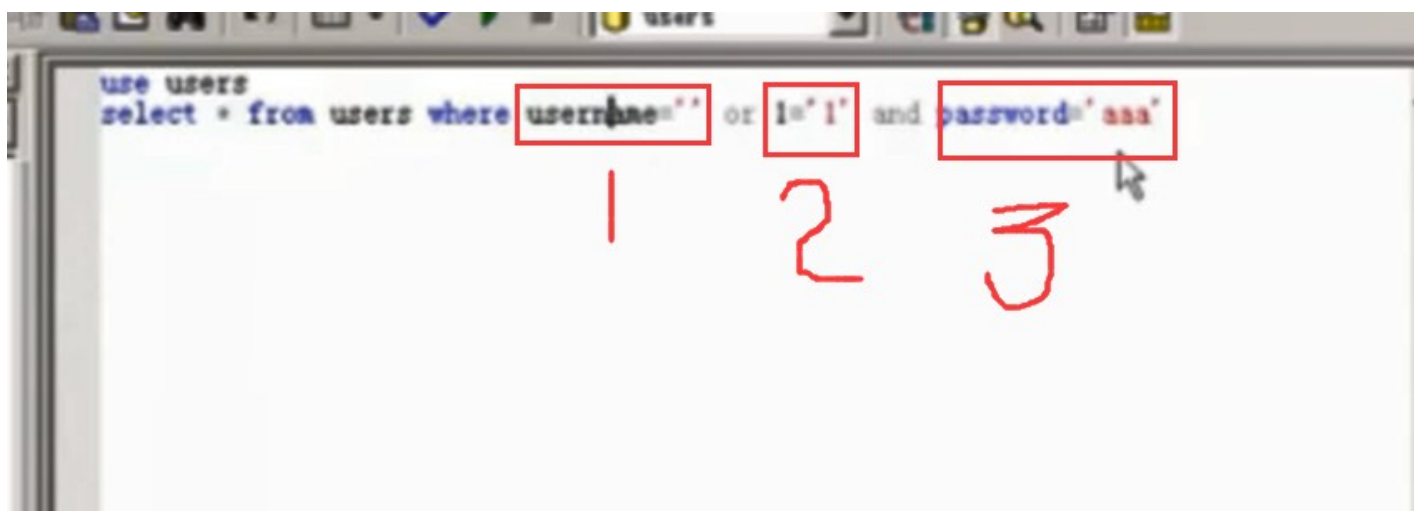
1-4依次为假or真or真and假

最后为一个真值(1=1), 这是密码绕过;

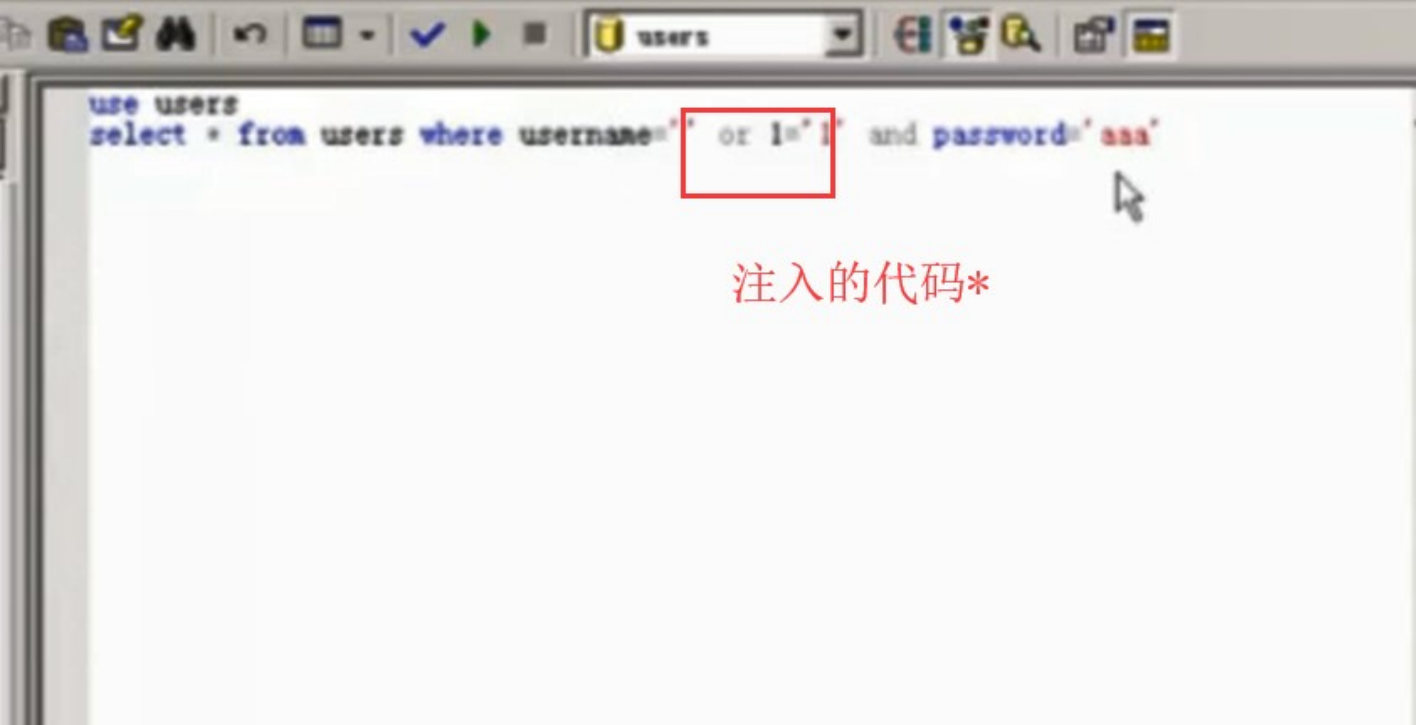
这场用户名为注入代码, passwd为任意的! 这里是用任意的密码去登录!!!

而这里的要求是在passwd这里构造注入代码! (用户名任意)

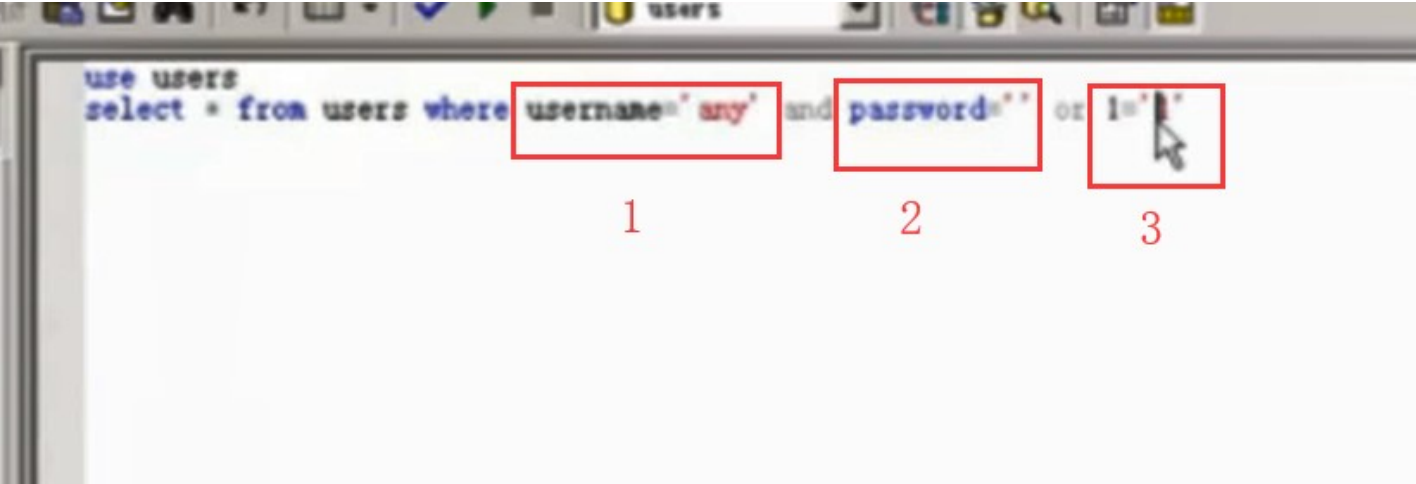
`and` 的优先级 比 `or` 高



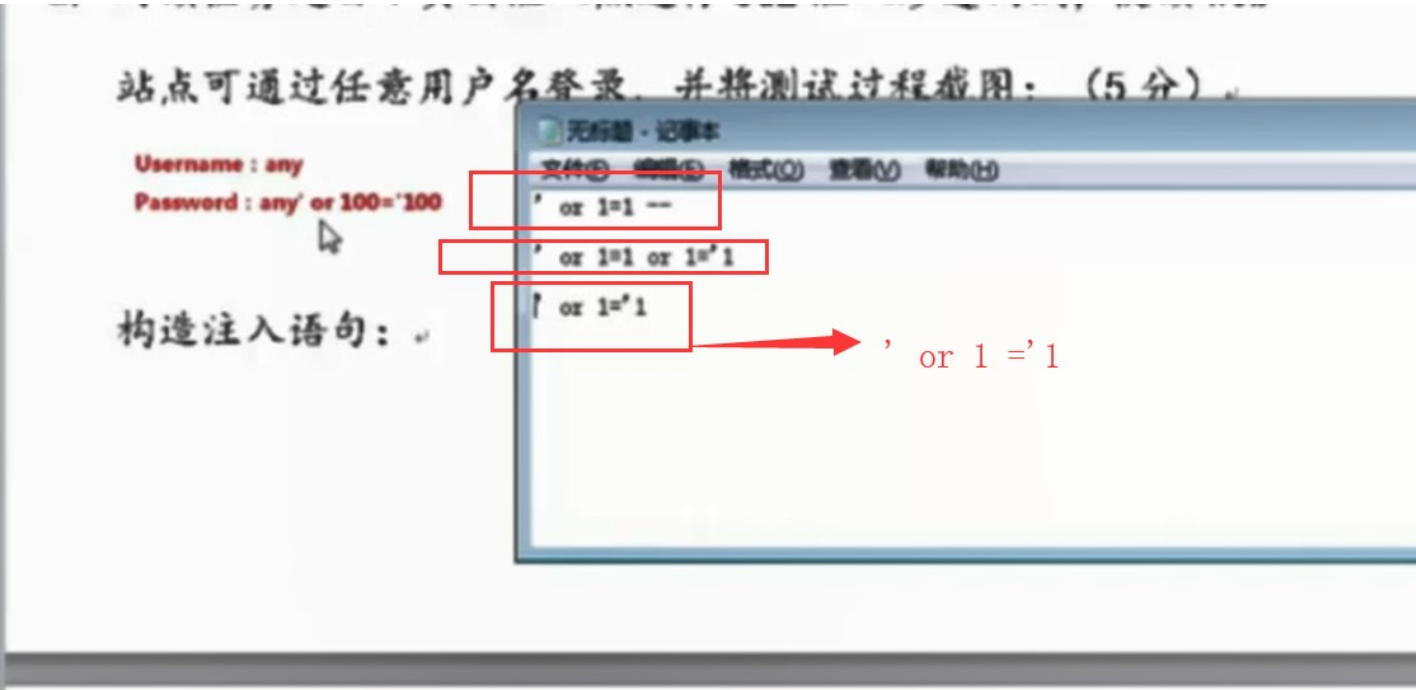
假 or 真 and 假 (先算and, 最后的结果为假),所以这里查不出什么结果;



但是如果把注入的代码 * 放入到后面的语句里面,



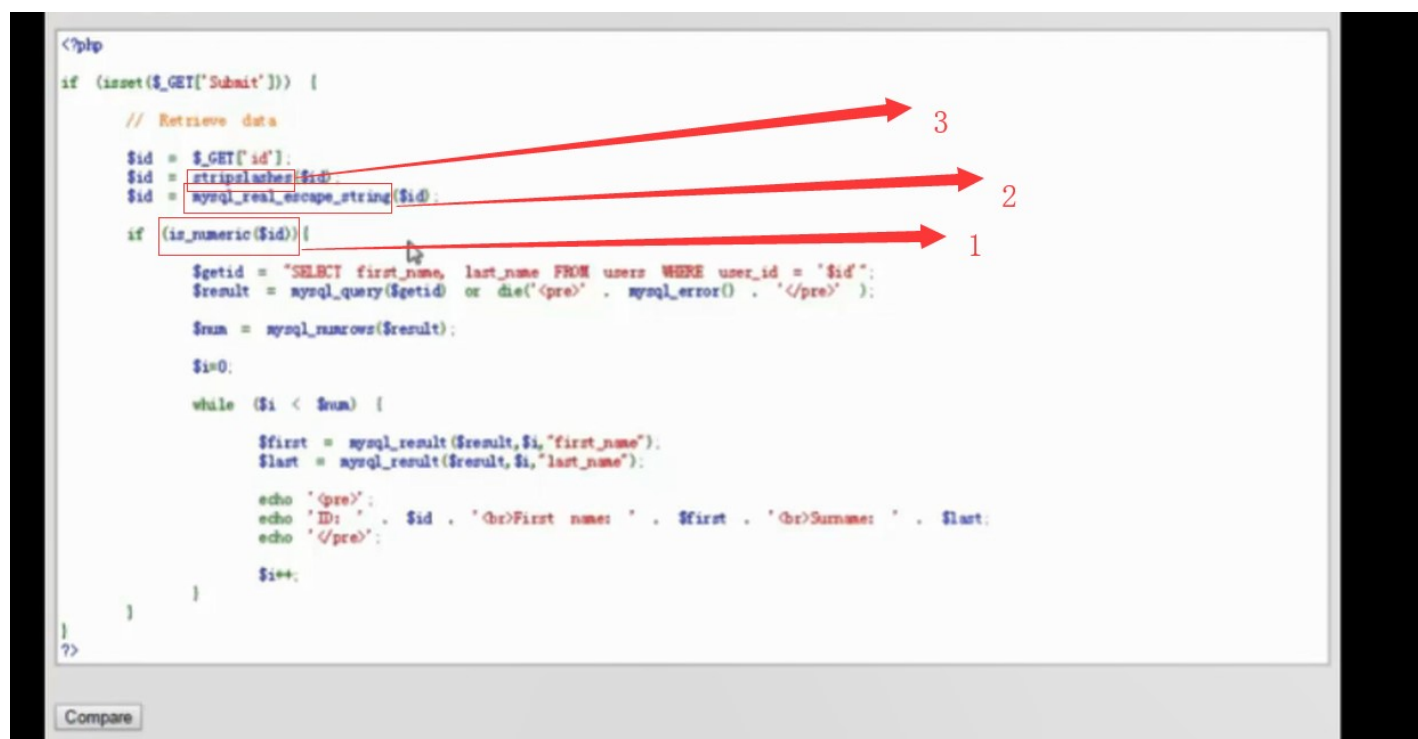
此时为:假 and 假 or 真;结果为真;




```
1 ' or 1=1 --
2 ' or 1=1 or 1 ='1
3 ' or 1 ='1
```

密码框里面的注入代码(上面两个在密码框和用户名里面填写都可以实现绕过,第三个只能填写密码框才能实现绕过)

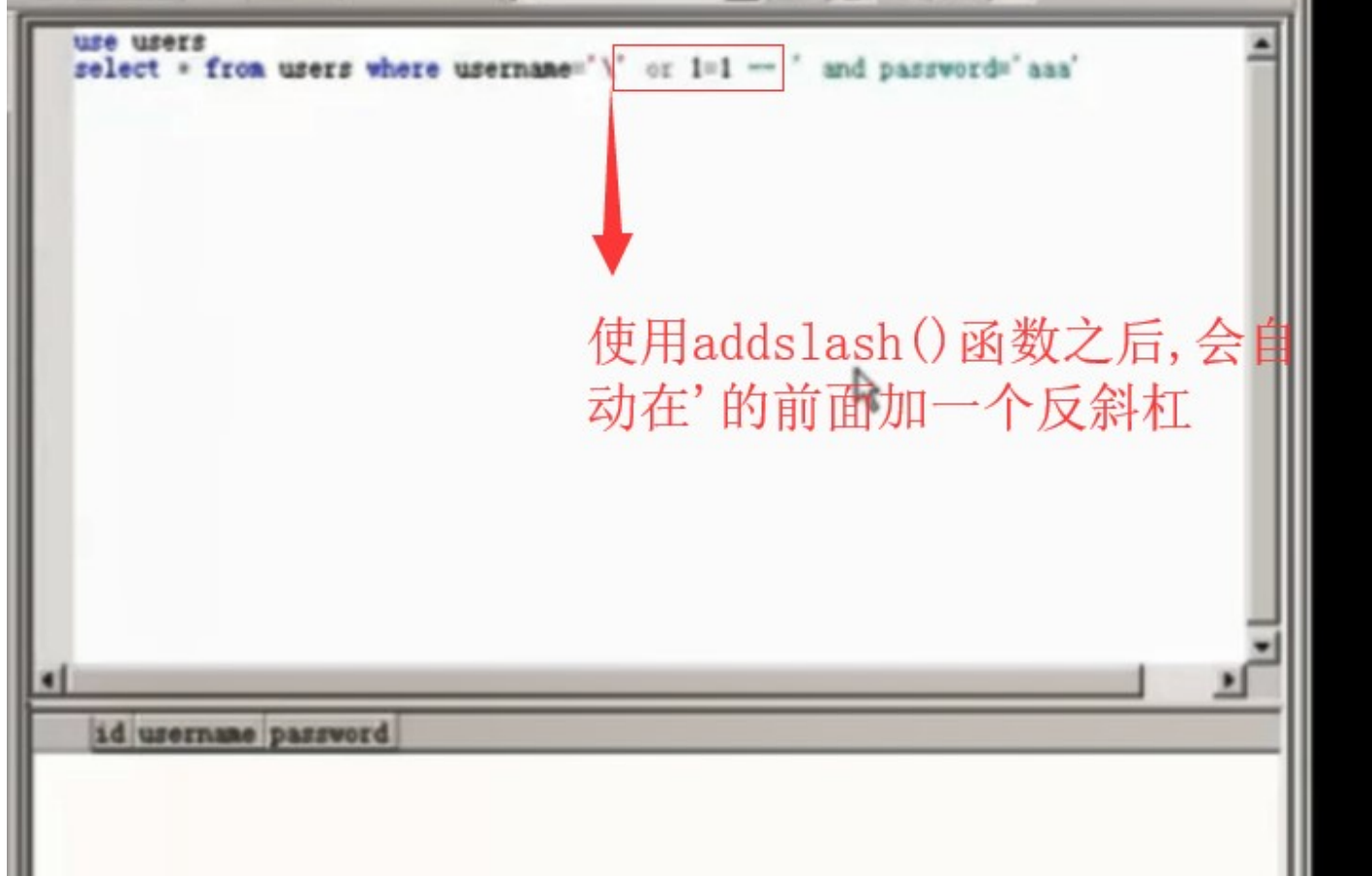
即: 上面两句注入语句不管写在用户名还是密码框里面的时候,它都可以绕过,比如: 如果第一句写在username里面的时候,可以实现密码绕过(密码随便写),第一句写在passwd里面的时候, 可以实现用户名绕过(用户名随便写);其他依次类推;



```
<?php
if (isset($_GET['Submit'])) {
    // Retrieve data
    $id = $_GET['id'];
    $id = addslashes($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>";
            $i++;
        }
    }
}
?>
```

1. `is_numeric` 这个函数用于防御数字 型注入;

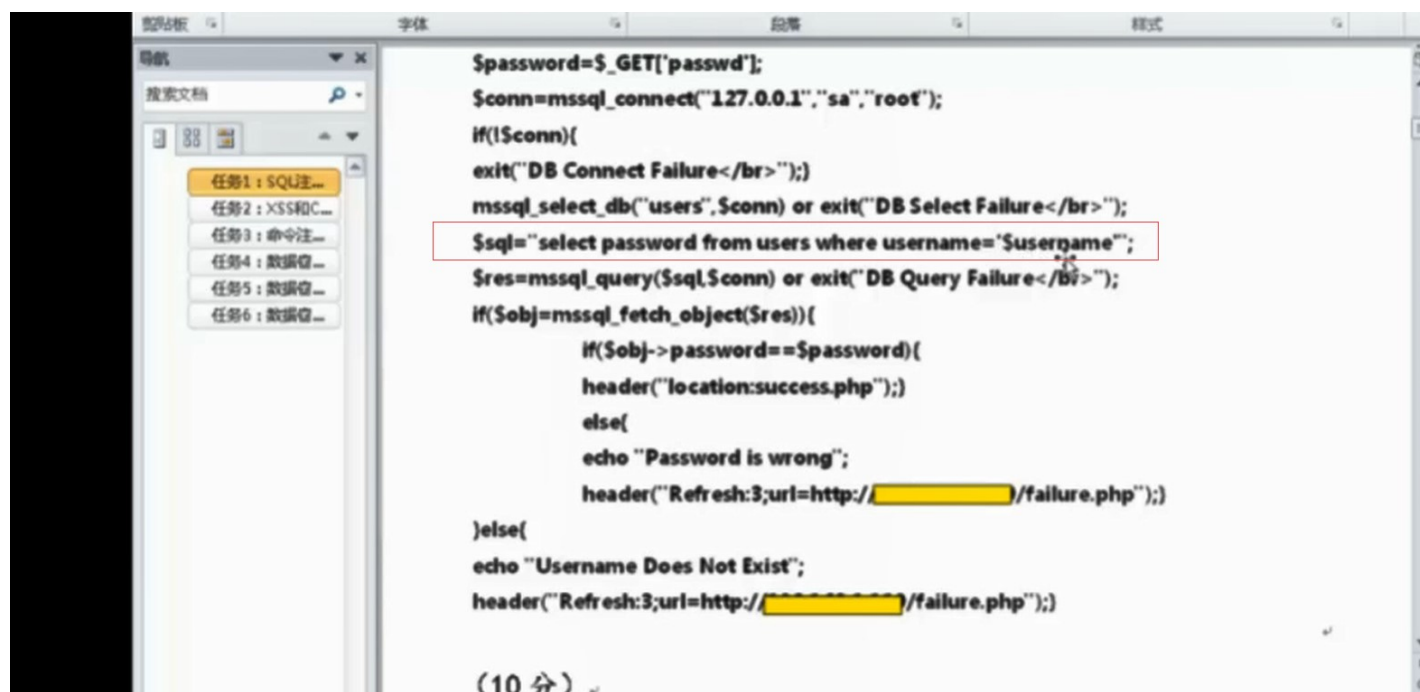
2.2和3这两个函数一般用于防御字符型注入,3这个函数只能用于mysql这种数据库,可以用`addslashes`这个函数;



addslashes()的作用是自动添加反斜杠;

(这里还可以(就是还可以被sql注入了)是因为在sqlserver里面转义符号不是反斜杠), 这里认为你输入的用户名就是反斜杠,这跟你输入或者输入any一样效果;

后面不能注入不是你把它单引号 转义了,而是那个反斜杠引起了语法错误;



他这里的答案不能绕过了;

也有些错误;header是不允许有echo这个语句的,否则会报错! 将这两者顺序改一改;(其实这里只是防止绕过,还有其他漏洞!!)

试题分析

第3题

mysql_real_escape_string()函数只适用于Mysql，sqlserver中没有相应的函数。
addslashes()函数适用于sqlserver，但sqlserver中的转义字符是'，而不是\。

利用addslashes()转义之后，可过滤注入语句any' or '1'='1，但无法阻止过滤注入语句any' or 1=1 --

试题分析

第3题

个人观点。

利用str_replace()函数过滤单引号'

```
$username=str_replace("'", "", $username);  
$password=str_replace("'", "", $password);
```

试题分析

第3题

个人观点。

利用str_replace()函数过滤单引号'

```
$username=str_replace("'", "", $username);  
$password=str_replace("'", "", $password);
```

str_replace("'", "", \$username);

str_replace("'", "", \$username); 这里是把username里面有没有字符 ' ,有的话就换成空的字符;(就是把敏感的字符替换掉);