

medium级别的防御措施

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

- 使用mysql_real_escape_string()函数对用户输入的id参数进行了过滤。
- 可以将单引号【'】、双引号【"】、反斜杠【\】、空字符【null】等进行转义。转义是把指定的字符转换成无意义的符号，比如PHP解析器不会把经过转义的引号当成引号来看待。
- PHP中另一个功能类似的函数：addslashes()

edu.51cto.com

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:

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Vulnerability: SQL Injection

User ID:

1'or 1=1 #

Submit

ID: 1'or 1=1 #
First name: admin
Surname: admin

ID: 1'or 1=1 #
First name: Gordon
Surname: Brown

ID: 1'or 1=1 #
First name: Hack
Surname: Me

ID: 1'or 1=1 #
First name: Pablo
Surname: Picasso

ID: 1'or 1=1 #
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

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'])){
```

```

4
5 // Retrieve data
6
7 $id = $_GET['id'];
8
9 $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
10 $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
11
12 $num = mysql_numrows($result);
13
14 $i = 0;
15
16 while ($i < $num) {
17
18     $first = mysql_result($result,$i,"first_name");
19     $last = mysql_result($result,$i,"last_name");
20
21     echo '<pre>';
22     echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
23     echo '</pre>';
24
25     $i++;
26 }
27 }
28 ?>

```

中级:

```

1 <?php
2
3 if (isset($_GET['Submit'])) {
4
5     // Retrieve data
6
7     $id = $_GET['id'];
8     $id = mysql_real_escape_string($id);
9
10    $getid = "SELECT first_name, last_name FROM users WHERE user_id = $id";
11
12    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
13
14    $num = mysql_numrows($result);
15
16    $i=0;

```

```

17
18     while ($i < $num) {
19
20         $first = mysql_result($result,$i,"first_name");
21         $last = mysql_result($result,$i,"last_name");
22
23         echo '<pre>';
24         echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
25         echo '</pre>';
26
27         $i++;
28     }
29 }
30 ?>

```

高级:

```

1 <?php
2
3 if (isset($_GET['Submit'])) {
4
5     // Retrieve data
6
7     $id = $_GET['id'];
8     $id = stripslashes($id);
9     $id = mysql_real_escape_string($id);
10
11     if (is_numeric($id)){
12
13         $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
14         $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>' );
15
16         $num = mysql_numrows($result);
17
18         $i=0;
19
20         while ($i < $num) {
21
22             $first = mysql_result($result,$i,"first_name");
23             $last = mysql_result($result,$i,"last_name");
24
25             echo '<pre>';
26             echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
27             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

```

medium级别的防御措施

```

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

```

- 使用mysql_real_escape_string()函数对用户输入的id参数进行了过滤。
- 可以将单引号【'】、双引号【"】、反斜杠【\】、空字符【null】等进行转义。转义是把指定的字符转换成无意义的符号,比如PHP解析器不会把经过转义的引号当成引号来看待。
- PHP中另一个功能类似的函数: addslashes()

edu.51cto.com

`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

```
1 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

```
mysql> create table hack
-> (
-> id int,
-> username varchar(20),
-> password varchar(30)
-> );
Query OK, 0 rows affected (0.00 sec)

mysql>
```

3.show tables;

```
mysql> show tables;
+-----+
| Tables_in_test |
+-----+
| hack           |
+-----+
1 row in set (0.02 sec)
```

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

```
mysql> insert into hack values(1,"ad'min","123456");
Query OK, 1 row affected (0.00 sec)
```

5.select * from hack;

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

而后输入:

select * from hack where username ='admin';

这样是不行的!

```
mysql> select * from hack where username='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:

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

Vulnerability: SQL Injection

User ID:

ID: 1+2

First name: Hack

Surname: Me

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>

http://en.wikipedia.org/wiki/SQL_injection

<http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>

<http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-s>

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

Insecure CAPTCHA

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

Vulnerability: SQL Injection

User ID:

Submit

ID: 3
First name: Hack
Surname: Me

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
http://en.wikipedia.org/wiki/SQL_injection
<http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
<http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>

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

1 or 1=1

Submit

ID: 1 or 1=1
First name: admin
Surname: admin

ID: 1 or 1=1
First name: Gordon
Surname: Brown

ID: 1 or 1=1
First name: Hack
Surname: Me

ID: 1 or 1=1
First name: Pablo
Surname: Picasso

ID: 1 or 1=1
First name: Bob
Surname: Smith