PHP 超级全局变量 PHP array_column() 函数

PHP 表单验证

本章节我们将介绍如何使用PHP验证客户端提交的表单数据。

PHP 表单验证

在处理PHP表单时我们需要考虑安全性。
本章节我们将展示PHP表单数据安全处理，为了防止黑客及垃圾信息我们需要对表单进行数据安全验证。

在本章节介绍的HTML表单中包含以下输入字段： 必须与可选文本字段，单选按钮，及提交按钮：

PHP 表单验证实例

* 必需字段。

名字: *

E-mail: *

网址:

备注:

性别: 女 男 *

Submit

您输入的内容是:

查看代码 »

上述表单验证规则如下：

字段	验证规则

名字	必须。 +只能包含字母和空格
E-mail	必须。 + 必须是一个有效的电子邮件地址（包含'@'和'.'）
网址	可选。如果存在，它必须包含一个有效的URL
备注	可选。多行输入字段（文本域）
性别	必须。 必须选择一个

首先让我们先看看纯HTML的表单代码：

文本字段

"名字", "E-mail", 及"网址"字段为文本输入元素，"备注"字段是 textarea。HTML代码如下所示:

```
“名字”: <input type="text" name="name">
E-mail: <input type="text" name="email">
网址: <input type="text" name="website">
备注: <textarea name="comment" rows="5" cols="40"></textarea>
```

单选按钮

"性别"字段是单选按钮，HTML代码如下所示：

```
性别:
<input type="radio" name="gender" value="female">女
<input type="radio" name="gender" value="male">男
```

表单元素

HTML 表单代码如下所示:

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

该表单使用 **method="post"** 方法来提交数据。

什么是 $_SERVER["PHP_SELF"] 变量?

$_SERVER["PHP_SELF"]是超级全局变量，返回当前正在执行脚本的文件名，与 document root相关。

所以， $_SERVER["PHP_SELF"] 会发送表单数据到当前页面，而不是跳转到不同的页面。

什么是 htmlspecialchars()方法?

htmlspecialchars() 函数把一些预定义的字符转换为 HTML 实体。
预定义的字符是：

- & （和号）成为 &
- " （双引号）成为 "
- ' （单引号）成为 '
- < （小于）成为 <
- > （大于）成为 >

PHP表单中需引起注重的地方？

$_SERVER["PHP_SELF"] 变量有可能会被黑客使用！

当黑客使用跨网站脚本的HTTP链接来攻击时，$_SERVER["PHP_SELF"]服务器变量也会被植入脚本。原因就是跨网站脚本是附在执行文件的路径后面的，因此$_SERVER["PHP_SELF"]的字符串就会包含HTTP链接后面的JavaScript程序代码。

XSS又叫 CSS (Cross-Site Script) ,跨站脚本攻击。恶意攻击者往Web页面里插入恶意html代码，当用户浏览该页之时，嵌入其中Web里面的html代码会被执行，从而达到恶意用户的特殊目的。

指定以下表单文件名为 "test_form.php":

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

现在，我们使用URL来指定提交地址 "test_form.php",以上代码修改为如下所示:

```
<form method="post" action="test_form.php">
```

这样做就很好了。

但是，考虑到用户会在浏览器地址栏中输入以下地址:

```
http://www.runoob.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

以上的 URL 中，将被解析为如下代码并执行：

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```

代码中添加了 script 标签，并添加了alert命令。 当页面载入时会执行该Javascript代码（用户会看到弹出框）。 这仅仅只是一个简单的实例来说明PHP_SELF变量会被黑客利用。

请注意， **任何JavaScript代码可以添加在<script>标签中！** 黑客可以利用这点重定向页面到另外一台服务器的页面上，页面代码文件中可以保护恶意代码，代码可以修改全局变量或者获取用户的表单数据。

如何避免 $_SERVER["PHP_SELF"] 被利用?

$_SERVER["PHP_SELF"] 可以通过 htmlspecialchars() 函数来避免被利用。

form 代码如下所示：

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

htmlspecialchars() 把一些预定义的字符转换为 HTML 实体。现在如果用户想利用 PHP_SELF 变量, 结果将输出如下所示：

```
<form method="post" action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

尝试该漏洞失败！

使用 PHP 验证表单数据

首先我们对用户所有提交的数据都通过 PHP 的 htmlspecialchars() 函数处理。

当我们使用 htmlspecialchars() 函数时，在用户尝试提交以下文本域:

```
<script>location.href('http://www.runoob.com')</script>
```

该代码将不会被执行，因为它会被保存为HTML转义代码，如下所示：

```
&lt;script&gt;location.href('http://www.runoob.com')&lt;/script&gt;
```

以上代码是安全的，可以正常在页面显示或者插入邮件中。

当用户提交表单时，我们将做以下两件事情：

1. 使用 PHP trim() 函数去除用户输入数据中不必要的字符 (如：空格，tab，换行)。
2. 使用PHP stripslashes()函数去除用户输入数据中的反斜杠 (\)

接下来让我们将这些过滤的函数写在一个我们自己定义的函数中，这样可以大大提高代码的复用性。

将函数命名为 test_input()。

现在，我们可以通过test_input()函数来检测 $_POST 中的所有变量, 脚本代码如下所示：

实例

```
<?php
// 定义变量并默认设置为空值
$name = $email = $gender = $comment = $website = "";
if ($_SERVER["REQUEST_METHOD"] == "POST")
{
$name = test_input($_POST["name"]);
$email = test_input($_POST["email"]);
$website = test_input($_POST["website"]);
$comment = test_input($_POST["comment"]);
$gender = test_input($_POST["gender"]);
}
function test_input($data)
{
```

```
$data = trim($data);
$data = stripslashes($data);
$data = htmlspecialchars($data);
return $data;
}
?>
```

运行实例 »

注意我们在执行以上脚本时，会通过$_SERVER["REQUEST_METHOD"]来检测表单是否被提交 。如果 REQUEST_METHOD 是 POST, 表单将被提交 - 数据将被验证。如果表单未提交将跳过验证并显示空白。

在以上实例中使用输入项都是可选的，即使用户不输入任何数据也可以正常显示。

在接下来的章节中我们将介绍如何对用户输入的数据进行验证。

← PHP 超级全局变量 PHP array_column() 函数 →

点我分享笔记