《软件安全》实验报告

姓名：禹相祐　　学号：2312900　　班级：计算机科学与技术

实验名称：

跨站脚本攻击

实验要求：

复现课本第十一章实验三，通过 img 和 script 两类方式实现跨站脚本攻击，撰写实验报告。有能力者，可以自己撰写更安全的过滤程序。

实验过程：

1. Script 方式

建立源文件，输入源代码如下：

```
<!DOCTYPE html>
<head>
<meta http-equiv="content-type" content="text/html;charset=utf-8">
<script>
window.alert = function()
{
confirm("Congratulations~");
 }
 </script>
 </head>
<body>
<h1 align=center>--Welcome To The Simple XSS Test--</h1>
<?php
ini_set("display_errors", 0);
$str =strtolower( $_GET["keyword"]);
$str2=str_replace("script","",$str);
$str3=str_replace("on","",$str2);
$str4=str_replace("src","",$str3);
echo "<h2 align=center>Hello ".htmlspecialchars($str).".</h2>".'<center>
<form action=xss_test.php method=GET>
<input type=submit name=submit value=Submit />
<input name=keyword  value="'.$str4.'">
```

```
</form>
</center>';
?>
</body>
</html>
```

运行后我们打开网页输入网址：http://127.0.0.1/xss_test.php，发现页面如下图：

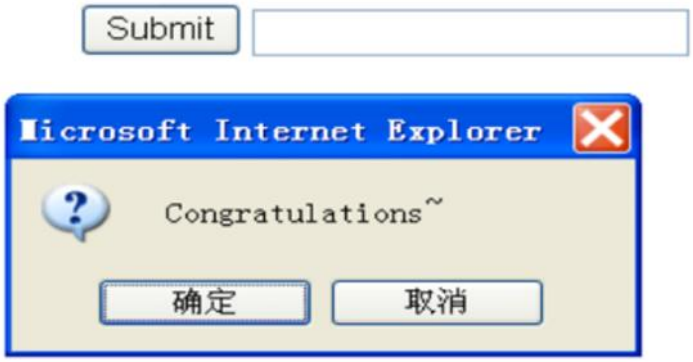
然后我们输入：<script>alert(‘xss’)</script>,效果如下图：

然后我们再次输入：<scrscriptipt>alert(’xss’)<scscriptript>,效果如下图：

然后查看代码可以知道，我们需要将最前面的 input 闭合掉才能实现我们想要的效果，所以再次输

入：”><scrscriptipt>alert(‘xss’)</scscriptript><!--，

并且将 magic_quotes_gpc 由 On 修改为 Off，如下图：



2. Img 方式

我们将原先代码进行修改，如下：

```
<!DOCTYPE html>
 <head>
 <meta http-equiv="content-type" content="text/html;charset=utf-8">
 <script>
 window.alert = function()
{
 confirm("Congratulations~");
 }
 </script>
 </head>
 <body>
 <h1 align=center>--Welcome To The Simple XSS Test--</h1>
 <?php
ini_set( "display_errors", 0);
$str=strtolower($_GET["keyword"]);
$str2=str_replace("script", "", $str);
$str3=str_replace("on", "", $str2);
$str4=str_replace("src", "", $str3);
echo "<h2 align=center>Hello ".htmlspecialchars($str).".</h2><center>
 <form action=xss_test.php method=GET>
 <input type=submit name=submit value=Submit />
 <input name=keyword value='".htmlspecialchars($str4)."'>
 </form>
 </center>";
```

```
?>
 <img src="ops!" onerror="var
payload=String.fromCharCode(99,111,110,102,105,114,109,40,39,88,83,83,39,41);eval
 (payload)">
 </body>
 </html>
```

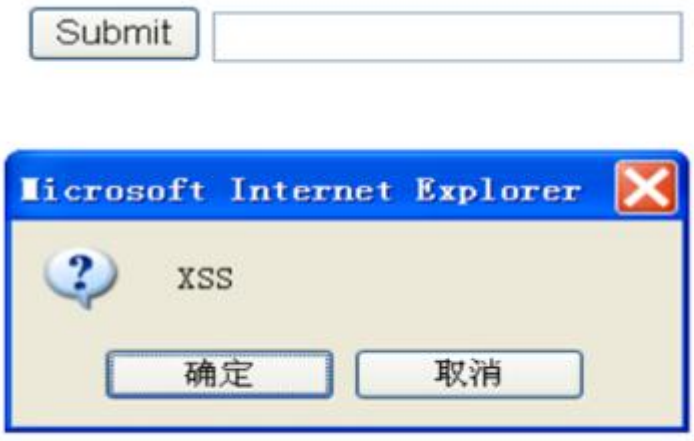
我们对上述代码进行简单分析：

1. 创建会报错的 img，故意使用错误的地址，方便触发 onerror 事件；

2. 用 ASCII 码拼出攻击的指令，即 let payload = String.fromCharCode(97, 108, 101, 114, 116, 40, 39, 88, 83, 83, 39, 41)；

3. 使用 eval 函数直接执行，绕过检测，实现攻击。

输入<img src=ops! onerror=”alert(‘XSS’)”>运行结果如下：



心得体会：

这次我学习了两个比较简单的跨站脚本攻击的方式，分别是 Script 方式，另一个是 img 方式。二者都能利用基础的特性实现攻击，但原理相差的比较大。通过这次实验，我也学习到了 script 过

滤关键字的实现方式，也算是收获满满了。