软件安全实验报告

姓名：辛浩然　　学号：2112514　　班级：信息安全、法学

1　实验名称

跨站脚本攻击

2　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。

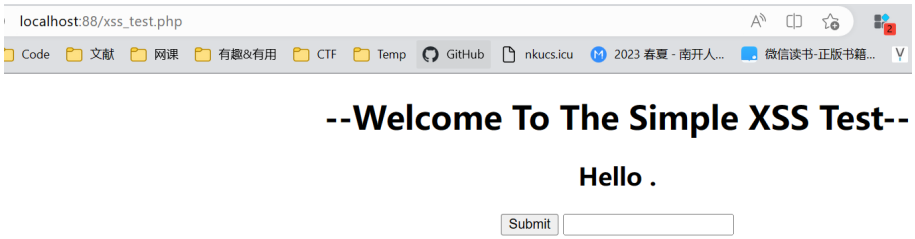
3　实验过程

编写 xss_test.php 文件

```
<!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>

```

访问页面，页面如图所示：



页面含有一个 Submit 按钮和输入框，且具有标题提示 XSS。

3.1 script 方法

输入最简单的 XSS 脚本：<script>alert('xss')</script> 进行测试。点击 Submit 按钮以后，效果如下：

发现 Hello 后面出现了输入的内容，并且输入框中的回显过滤了 script 关键字，只是最简单的一次过滤。可以利用双写关键字绕过，构造脚本：<scrscriptipt>alert('xss')</scscriptript> 测试。执行效果如下：

虽然输入框中的回显确实是想要攻击的脚本，但是代码并没有执行。

在页面点击右键查看源码，尝试从源码片段中分析问题。右键源码如下：


```
1   <!DOCTYPE html>
2
3   <head>
4       <meta http-equiv="content-type" content="text/html;charset=utf
    -8">
5       <script>
6           window.alert = function () {
7               confirm("Congratulations~");
8           }
9       </script>
10  </head>
11
12  <body>
13      <h1 align=center>--Welcome To The Simple XSS Test--</h1>
14      <h2 align=center>Hello &lt;scrscriptipt&gt;alert(&#039;xss
    &#039;)&lt;/scscriptript&gt;.</h2><center>
15  <form action=xss_test.php method=GET>
16  <input type=submit name=submit value=Submit />
17  <input name=keyword value="<script>alert('xss')</script>">
18  </form>
19  </center></body>
20
21  </html>
22
```

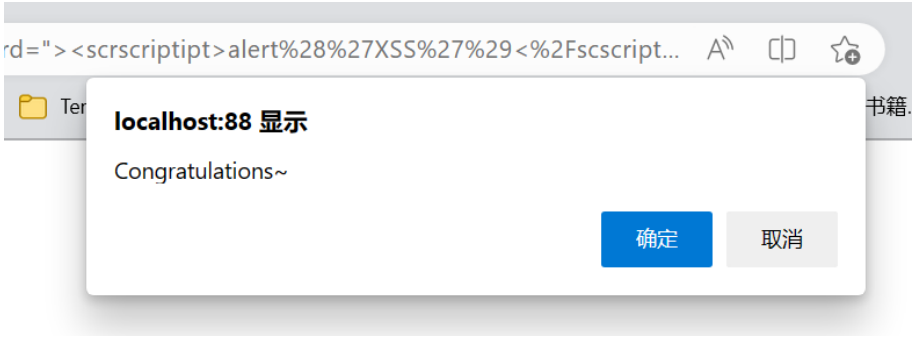
分析第 6 行重写的 alert 函数。如果可以成功执行 alert 函数的话，页面将会跳出一个确认框，显示 Congratulations 。这是 XSS 成功攻击的的标志。

向下查看 17 行的 <input> 标签，这是唯一能输入且有可能控制的地方。<input name=keyword value=”<script>alert(’xss’)</script>”> 分析这行代码可知，虽然成功插入了 <script></script>

标签组, 但是并没有跳出 input 的标签，使得脚本仅仅可以回显而不能利用。因此需要使前面的 <input> 标签闭合，构造如下脚本：

”><scrscriptipt>alert(’XSS’)</scscriptript><!—

弹出确认框，XSS 攻击成功。执行效果如下：



分析输入的内容：

<input name=keyword value=””><script>alert(’xss’)</script><!–”>

”> 用来闭合前面的 <input> 标签。而 <!– 其实是为了美观，用来注释掉后面不需要的”> , 否则页面就会在输入框后面回显”> 。

3.2 img 方法

使用 <img> 标签可用这样构造脚本：

<img src=ops! onerror=”alert(’XSS’)”>

<img> 标签是用来定义 HTML 中的图像，src 一般是图像的来源。而 onerror 事件会在文档或图像加载过程中发生错误时被触发。所以上面这个攻击脚本的逻辑是，当 img 加载一个错误的图像来源 ops! 时，会触发 onerror 事件，从而执行 alert 函数。

查看页面源码：

```
<?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>';
?>

```

可以发现，源码对 on 和 src 等关键词进行了过滤，利用双写关键词绕过，构造脚本：

”> <img srsrcc=ops! oonnerror=”alert(’XSS’)”><!–

其中，srsrcc、oonnerror 绕过源码对关键词的过滤，”> 用来闭合前面的 <input> 标签。<!–用来注释掉后面不需要的”> 。

弹出确认框，XSS 攻击成功。执行效果如下：

4 心得体会

本次实验利用了 img 和 script 两类方式实现跨站脚本攻击，加深了对跨站脚本攻击的理解和认识。