《漏洞利用及渗透测试基础》实验报告

姓名：管昀玫 学号：2013750 班级：计科一班

**实验名称：**

跨站脚本攻击

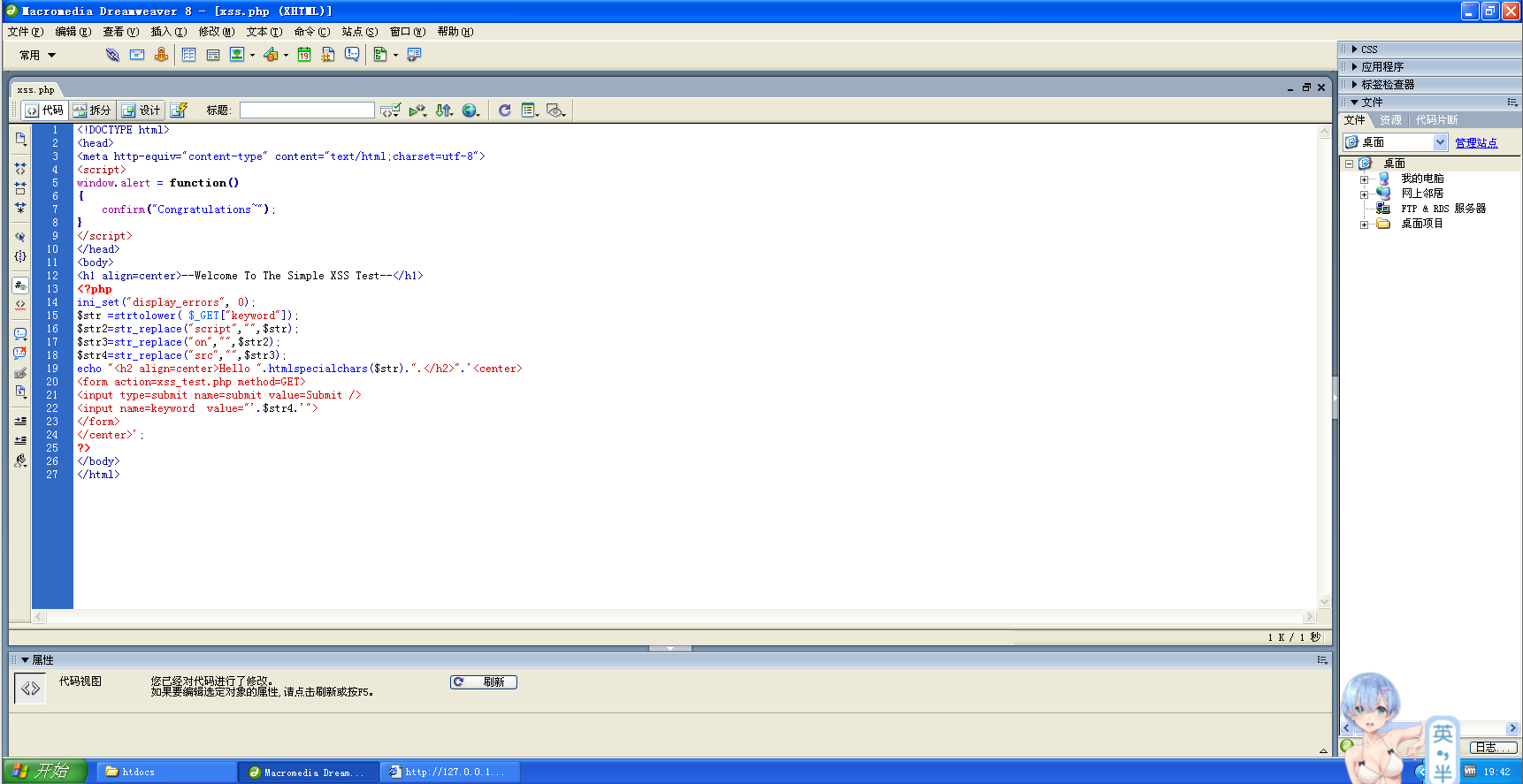
**实验要求：**

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：**

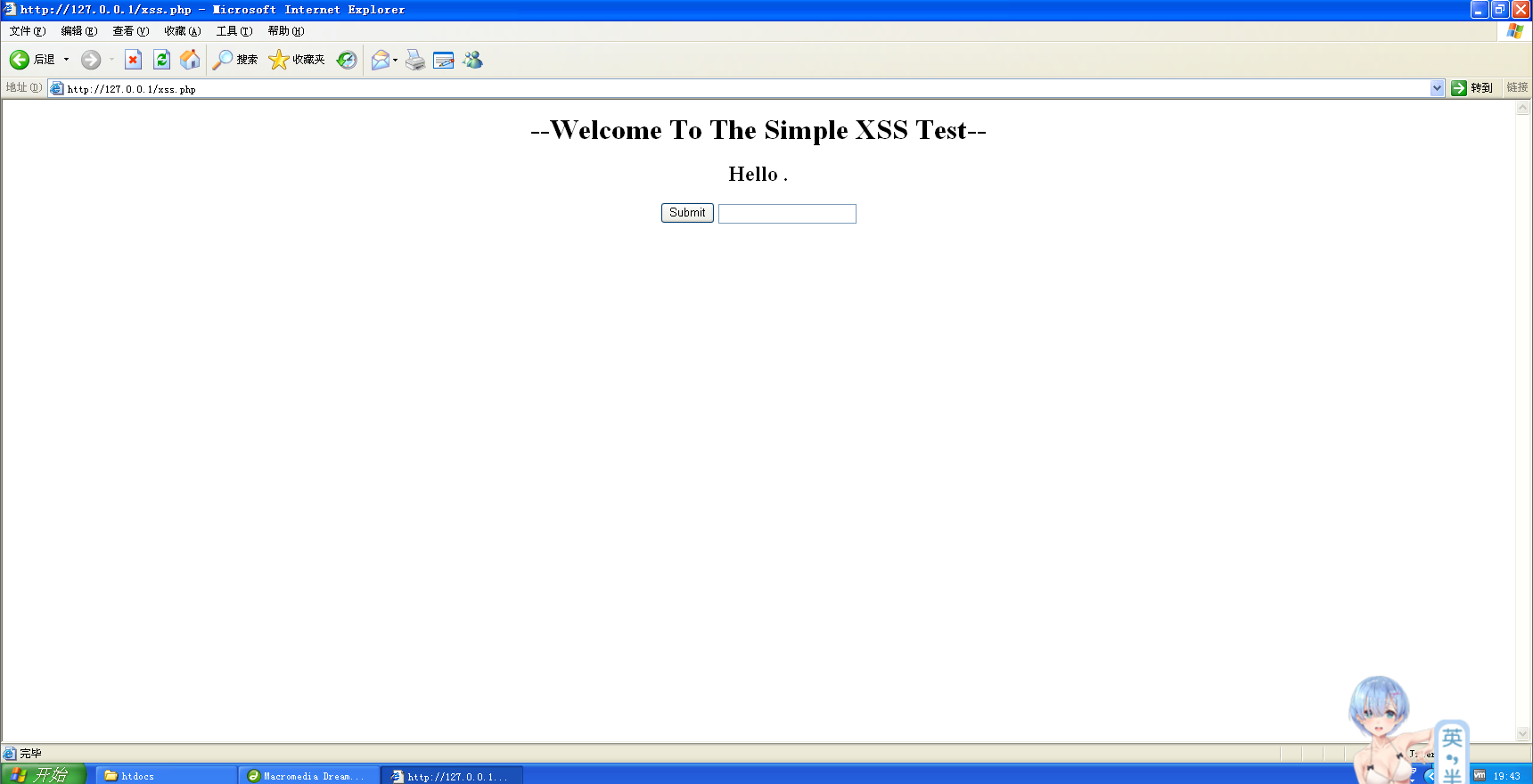
1. 实验准备

创建xss.php，写入书中代码



首先从黑盒测试的角度来进行实验，访问URL：http://192.168.19.131/xss\_test.php

访问页面查看效果：



如图可以看到一个Submit按钮和输入框，并且还有标题提示XSS。

1. 输入尝试

首先在文本框中输入<script>alert('first test')</script>，可以看到效果：

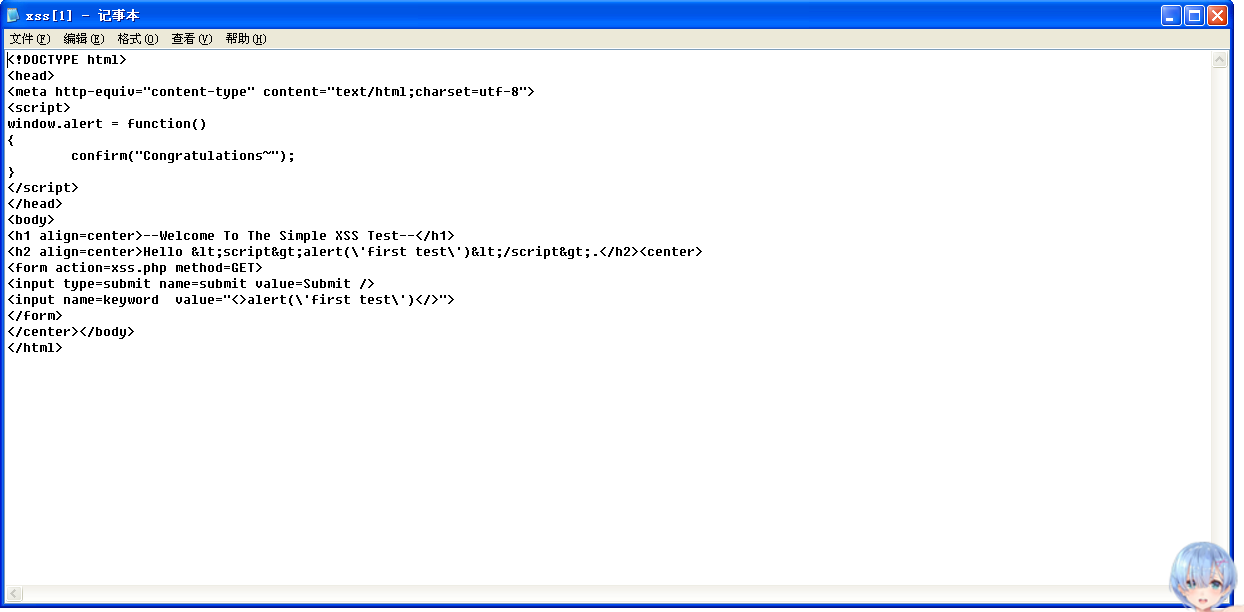


发现在上面显示出了之前输入的字样，但是稍微有一点不同，alert中的’变成了\’，也就是转义字符

同时发现在文本框中，原本的<script>alert('first test')</script>变成了<>alert(\'first test\')</>，里面的script字样都被替换成了’’，也就是给删除掉了。同时’也都加上了转义符变成了\’。由此猜测，可能在这个网页中存在有一定的防护机制，屏蔽了输入一句话脚本script。

1. 分析修改，再次尝试

按F12进入网页的开发者模式（这里是右键-查看网页源代码），得到源代码如下：



可以看到如下的几行代码：

<h2 align=center>Hello &lt;script&gt;alert(\'first test\')&lt;/script&gt;.</h2><center>

<form action=xss.php method=GET>

<input type=submit name=submit value=Submit />

<input name=keyword value="<>alert(\'first test\')</>">

发现确实是删除了script字样，猜测网站可能有保护机制，会对字符串进行匹配检测

接下来尝试输入两次script，并且将其中一个拆开，让他只检测到一个script并删除（即利用双关键字），让第二个在删除替换后形成完整的<script>。

第二次尝试输入如下代码：

<scriscriptpt>alert('second test')</scriscriptpt>

得到如下图所示结果：



可以看见这次在文本框中解决了上次的script被删除的问题，但是又发现了如下两个问题：

1 单引号前面还是加了转义字符

2 second变成了secd

那么对于以上两个问题，第一个通过将php-apache2handler.ini，并将“magic\_quotes\_gpc = On”设置为“magic\_quotes\_gpc = Off”可以解决；



第二个是“on”被删除了，猜测可能在这里面还有对于on的防护机制。

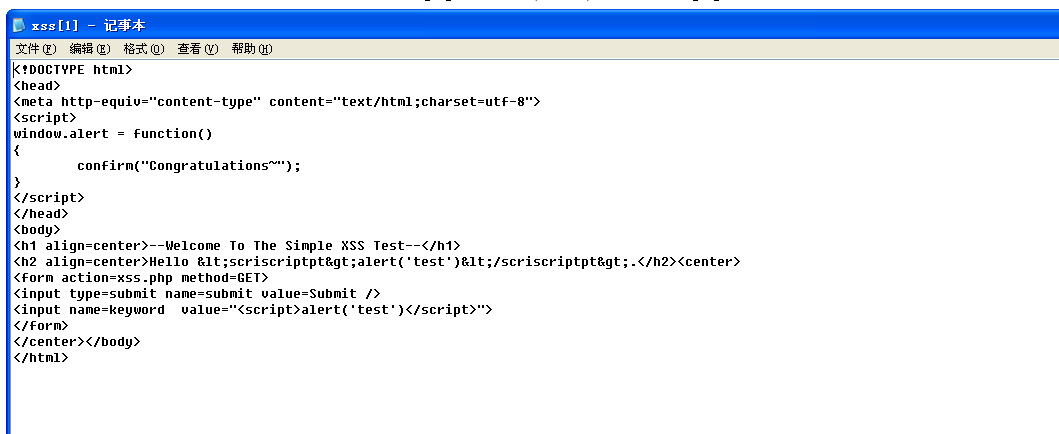
先绕开这个，输入<scriscriptpt>alert('test')</scriscriptpt>尝试



发现输入框中的文字正确了，但是没有按照预想的状态去执行alert(‘test’)

1. 源码分析

打开网页的源代码



首先可以看到我们想要执行的alert不再是在网页上输出一个字符串，而是经过重写变成了一个弹框，弹框中显示的是“congratulations~”字样

然后在下面可以看见：

<h2 align=center>Hello &lt;scriscriptpt&gt;alert('test')&lt;/scriscriptpt&gt;.</h2><center>

<form action=xss.php method=GET>

<input type=submit name=submit value=Submit />

<input name=keyword value="<script>alert('test')</script>">

</form>

前面h2那里可以看见我们输入的符号都变成了&lt;等字样，这些是暂时没有办法改变的，不能从这里下手。<input name=keyword value="<script>alert('test')</script>">这句是唯一可能输入并控制的地方。

其中value="<script>alert('test')</script>"的字样，可以看见我们之前输入的一句话木马被完整的存在了这里，之所以没有执行是因为此时还在input的标签里，并且该语句的前后都被加上了””。

为了让他能独立成为一个可以执行的语句，需要将前面的<input给闭合，那也就是需要在左边加上”>。然后需要把右边剩下的也同样闭合，那就是在右边加上<”。

1. 第三次尝试

输入”><scriscriptpt>alert('test')</scriscriptpt><”



submit后可以看见攻击成功，弹出了对应窗口

1. img攻击尝试

对于img函数来说，需要指定显示图片的源，以及如果图片显示错误应当执行的函数。那么在这里，显示图片出现错误时，就可以使用之前重写过的alert函数，来显示弹窗。用来绑定的关键词是onerror

结合之前在第二次尝试的时候，输入了second后，文本框内变成了secd，也就是说这个程序也将on进行了屏蔽，那么在输入的时候调整输入，将onerror改成oonnerror，这样在删除了on后就能拼接成正确的语句。

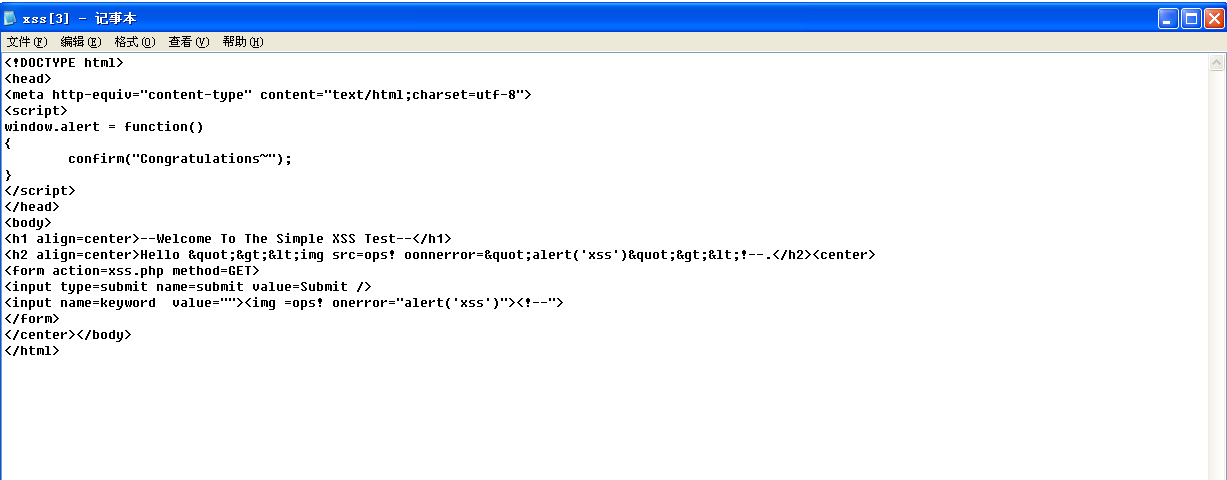
之前考虑的将前后括号进行闭合依旧保持不变进行闭合。

尝试输入语句 “><img src=ops! oonnerror=”alert(‘xss’)”><!--



"> 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 ">

发现还是没有显示弹窗，打开源代码进行分析



发现在value后面我们的src=ops!变成了=ops!，也就是说src也被作为关键词给屏蔽了，同样使用复写的方式将src改成srsrcc，再次尝试

本次输入语句“><img srsrcc=ops! oonnerror=”alert(‘xss’)”><!--

得到结果：



出现了弹窗，攻击成功

**心得体会：**

感觉xss跨站脚本攻击和sql注入有一些相似，都是由于其对于输入语句的检测不足引发的安全隐患。

对于以上的这种可以先使用这种简单的过滤，然后将已经过滤完了的语句利用正则表达式进行二次过滤，比如说如果其符合语句中含有<script>的，不论在这之间的任何两个字符之间穿插有多少个任意的字符，都是满足这样的正则表达式框架，那么这个就是可以被识别出来的一句话木马；同理可以再设置对于<img>等类似的正则表达式检测规则。

亦或者是针对上面需要将<input闭合而在开头输输入的”>这些也写成一条简单的筛选规则，将其进行过滤。