**《软件安全》实验报告**

姓名：胡博浩 学号：2212998 班级：信息安全

**一、实验名称：**

跨站脚本攻击

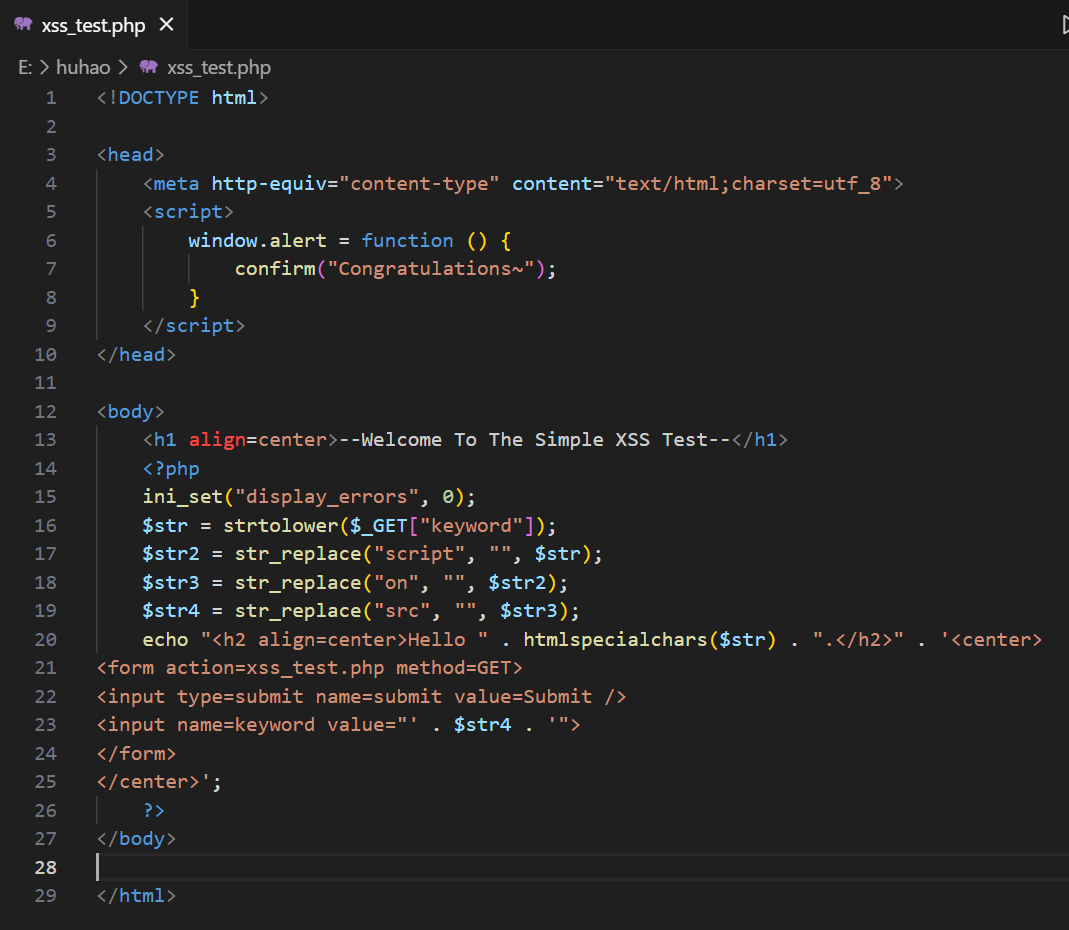
**二、实验要求：**

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。

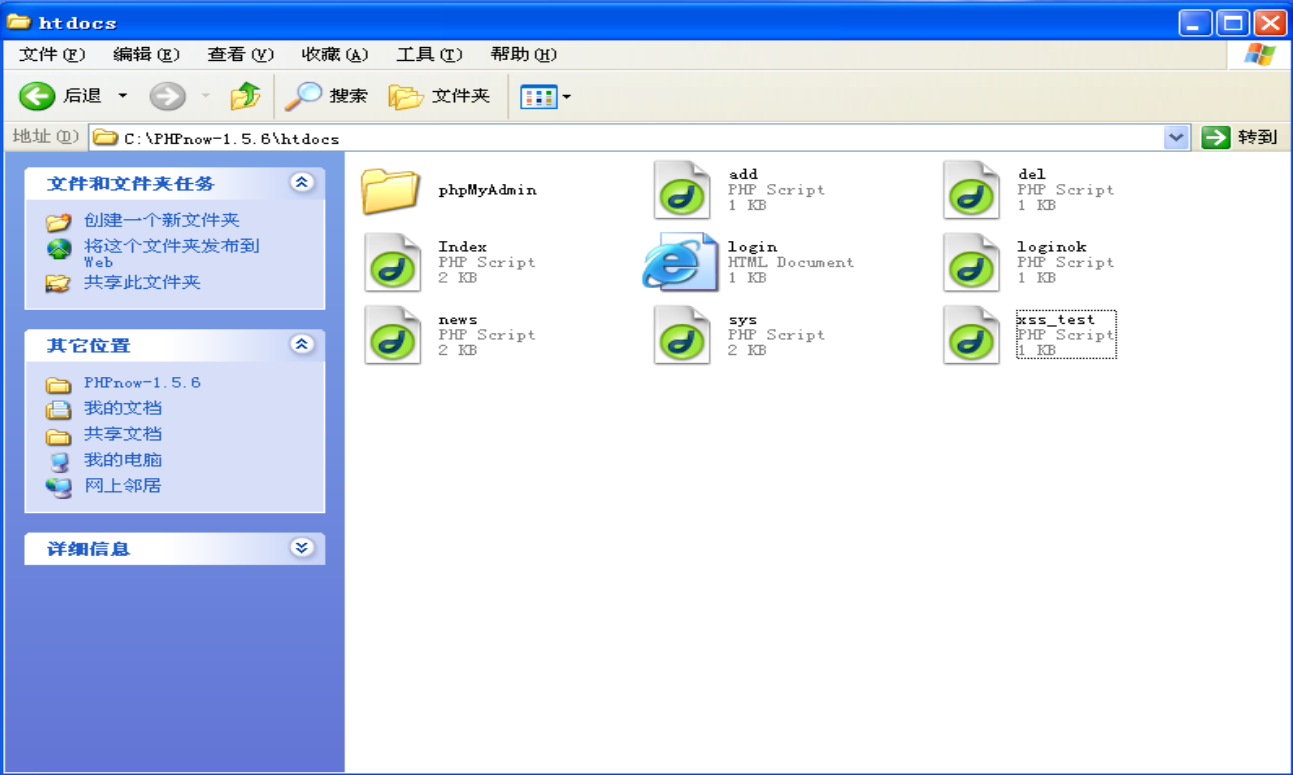
**三、实验过程：**

1.创建xss攻击测试网站：

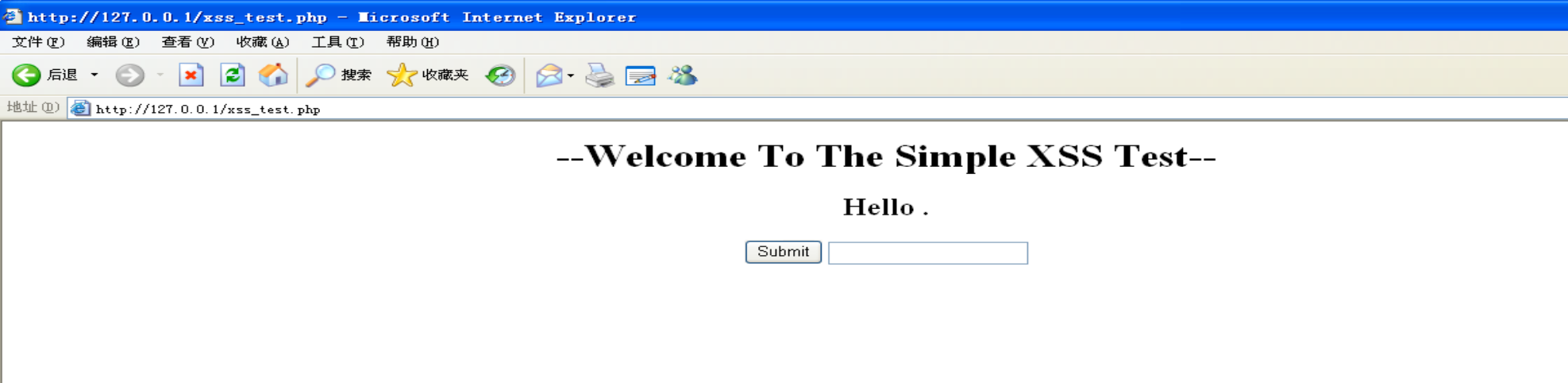
先编写php代码，文件名为xss\_test.php，代码如下：



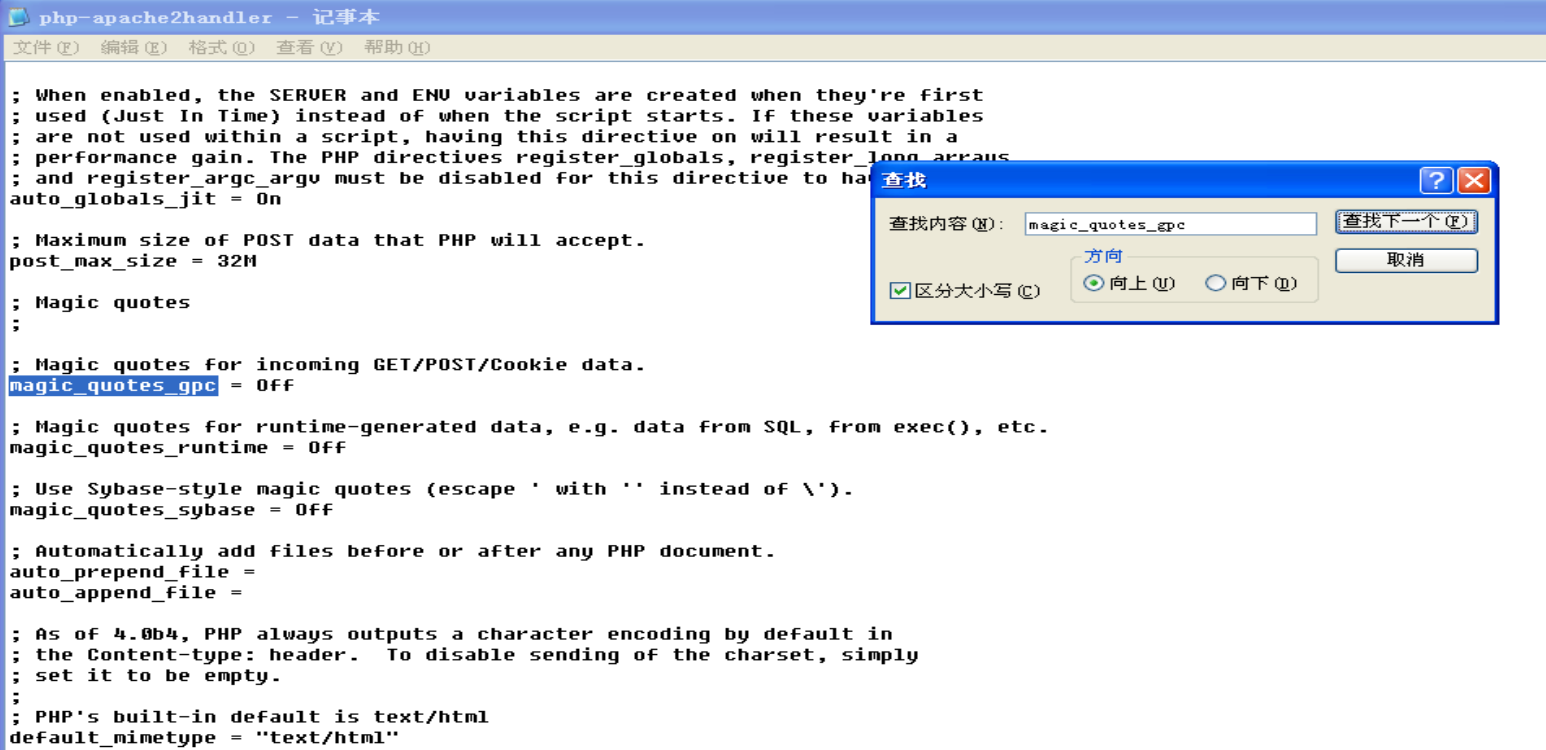
然后将php文件放到PHPnow-1.5.6/htdocs目录下,如下所示：



进入网址 http://127.0.0.1/xss\_test.php，能正常显示并输入信息



然后要关闭php服务器的安全检查机制，防止php服务器对与符号进行转义。即调整php服务器的安全等级，修改php-apache2handler.ini中的magic\_quotes\_gpc=Off。修改完这个配置选项之后，重启机器。



2.进行xss攻击：

（1）script脚本攻击

脚本攻击的原理是在 PHP 页面中嵌入一段 JavaScript 代码，这段代码能够被浏览器的引擎解析并执行。当浏览器渲染该页面时，这段脚本就会被触发执行。由于 PHP 页面的代码已经进行了基本的 script 注入检测，因此我们需要采取更加隐蔽的方式进行注入。

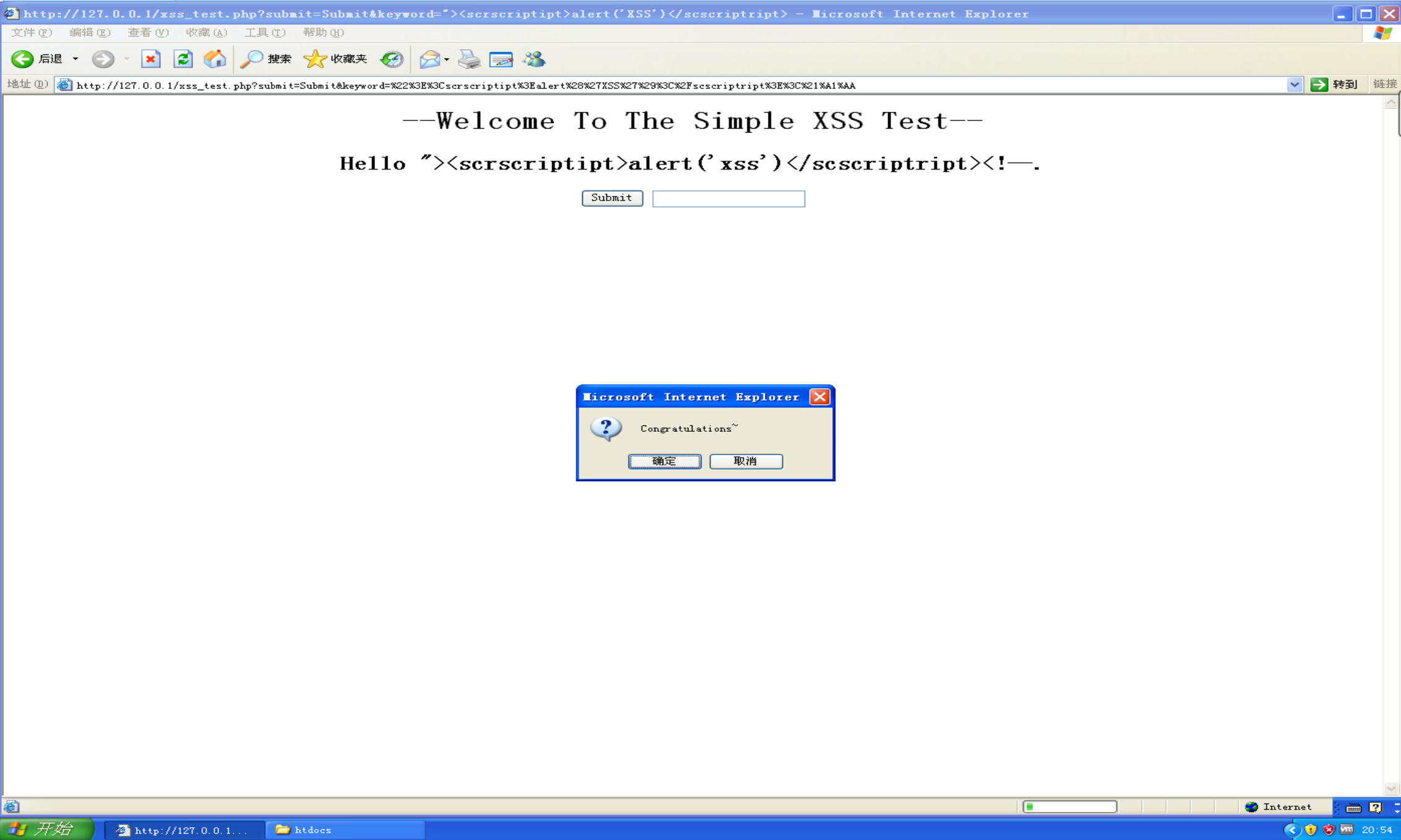
在输入框中输入以下内容：

"><scrscriptipt>alert('XSS')</scscriptript><!—

这段代码会被PHP修正为如下的代码，但修正后的结果正是我们所需要的。"><script>alert('XSS')</script><!—



提交之后执行效果如下：



（2）img脚本注入

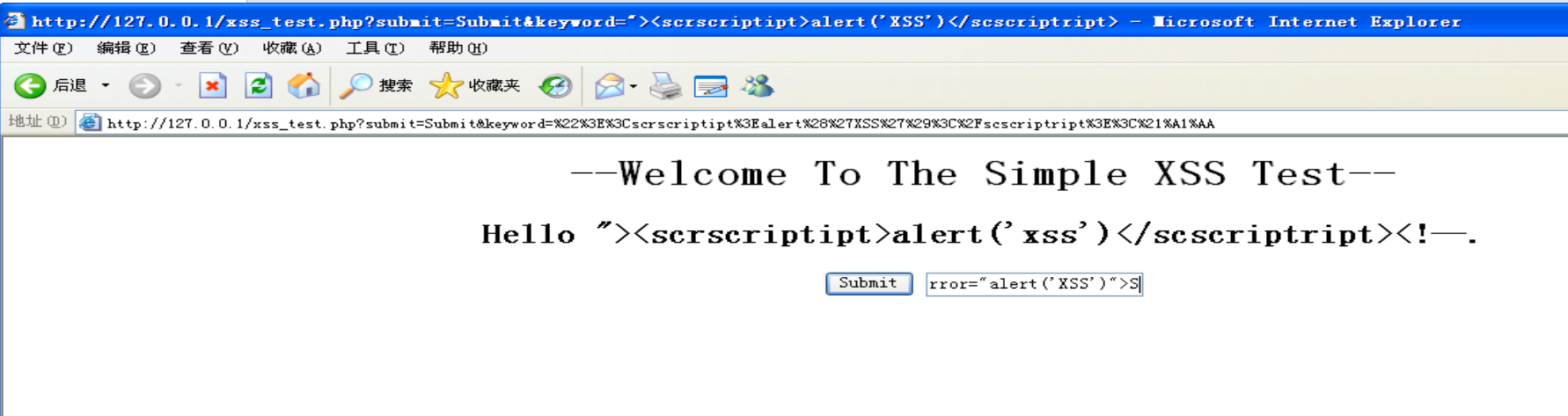
<img>标签是用来定义HTML中的图像， 其中src属性指定了图像的来源。而onerror事件会在文档或图像加载过程中发生错误时被触发。所以攻击脚本的逻辑是，当img加载一个错误的图像来源ops!时，会触发onerror事件，从而执行alert函数。另外，同样要注意对于脚本进行隐藏。

在输入框中输入以下内容：

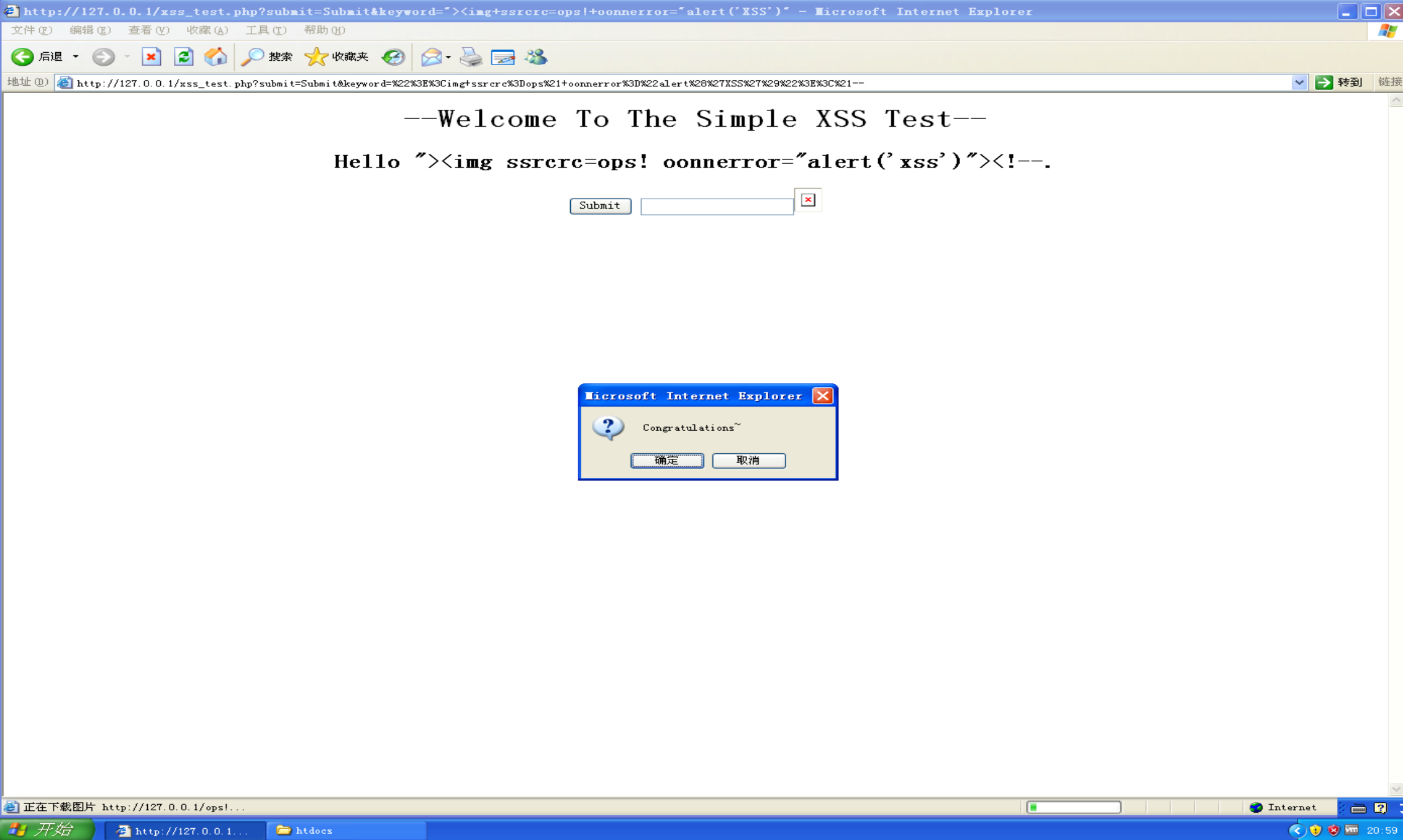
"><img ssrcrc=ops! oonnerror="alert('XSS')"><!--

这段代码会被PHP修正为如下的代码，但修正后的结果正是我们所需要的。

"><img src=ops! onerror="alert('xss')"><!—



提交之后执行效果如下：



3.尝试使用更安全的过滤机制：

Php代码中提供了一种能够强制将与HTML语言冲突的字符进行特殊编码的机制，能够避免用户输入被解释称HTML中的使用的字符，即使用htmlspecialchars函数进行转义。该函数会将特定的字符（如<、>、&、"和'）转换为HTML实体。例如，将<转换为&lt;，将> 转换为&gt;。这种转换确保了用户输入的数据在浏览器中不会被错误地解释为HTML或JavaScript代码，从而减少了潜在的XSS攻击风险。

通过正确使用htmlspecialchars函数，我们可以有效地保护PHP应用程序免受XSS攻击和其他安全威胁的侵害。并且浏览器引擎在解释执行的时候依然能够将其转化成原本的符号。

**四、心得体会：**

通过本次实验，我对跨站脚本攻击有了更为深入的理解和掌握。XSS作为一种极具威胁的网络安全漏洞，其潜在的破坏力不容小觑。它不仅可能导致用户敏感信息的泄露，还可能让恶意代码在受害者的浏览器中执行，进而对网站内容和用户计算机安全造成重大威胁。

在实验中，我尝试了通过嵌入简单的脚本标签和图像标签来发起攻击。尤其是当我通过巧妙的方式绕过“script”关键词的过滤时，我意识到防御措施必须足够智能才能有效应对此类攻击。而利用<img>标签的onerror事件进行隐蔽攻击，则让我深刻体会到在防范XSS时，必须全面考虑所有可能的攻击向量，并具备全局的安全视角。

同时，我也学习了如何运用htmlspecialchars函数对用户输入进行HTML实体编码，以防止恶意代码的执行。这一简单而实用的方法，让我认识到输入验证和输出编码在Web安全中的核心地位。输入验证能够确保不合规的输入被拦截在系统之外，而输出编码则能够确保即使输入中包含恶意代码，也无法在浏览器中执行。

此外，我还了解了内容安全策略这一高级防御手段，它通过限制脚本的来源和执行环境，为Web应用提供了更为严密的安全保护。同时，我也对现代浏览器内置的安全机制有了更深的了解，这些机制在保护用户方面发挥着重要作用，但开发者仍需保持警惕，不可完全依赖浏览器的保护。

这次实验的经历让我深刻认识到Web安全防护的复杂性和重要性。在未来的开发工作中，我将把严格的输入验证和输出编码作为不可或缺的开发标准，确保所有用户输入的数据都经过严格的安全处理。同时，我也会持续关注最新的安全技术和攻击手法，保持对潜在安全威胁的敏感度，为构建更加安全的Web应用贡献自己的力量。