《漏洞利用及渗透测试基础》实验报告

姓名：申宗尚　学号：2213924　班级：信息安全

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程序。

实验过程：

1. 首先，利用 PHP 工具本地建站，在路径 PHP\htdocs 下新建 attack_web 文件夹在其中编辑 php 文件：xss_test.php

xss_test.php - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
<!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>';
?>

</body>
</html>
```

2. 然后，我们登录 127.0.0.1/attack_web/xss_test.php，可以看到如下界面：

接着，我们尝试两种攻击方式。

3. <1>img 脚本注入

这种攻击脚本的逻辑是，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。

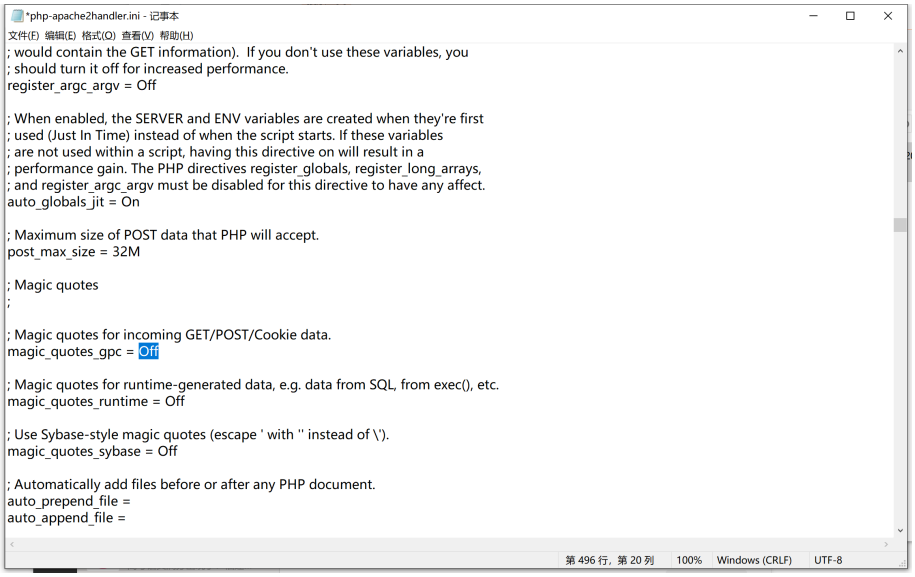
输入框中输入：

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

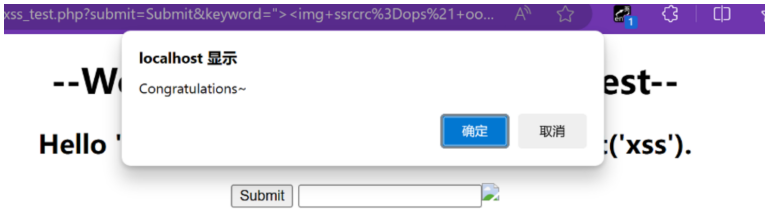
这段文本会被 php 代码按照我们原来的想法修正为<img src=ops! onerror=″alert('XSS')″>

（由于 xss_test.php 中有对输入 str 的字符替换）

这里发现攻击失败了，原因可能是 PHP 服务器自动对输入的双引号进行转义，从而预防用户的恶意特殊输入。因此为了完成实验，我修改一下 PHP 的设置：将 php-apache2handler.ini 的 magic_quotes_gpc 属性设置为“Off”：



然后，我们重新进行攻击：



攻击成功。

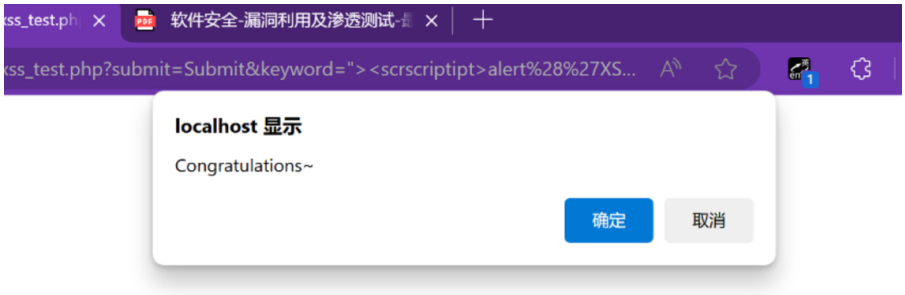
<2>script 脚本攻击

这里尝试 script 脚本攻击。脚本注入攻击的原理是在 PHP 页面中嵌入能够被浏览器引擎执行的 JavaScript 脚本。当浏览器渲染页面时，这些脚本会被触发执行，可能导致恶意行为的发生。

但由于 PHP 页面的代码已经对基本的脚本注入进行了检测（如关键字“script”），因此攻击者需要采取进一步的措施来隐藏他们的恶意代码，使其能够绕过 PHP 的过滤机制，成功执行。输入框中输入

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

这段会被 php 代码按照我们原来的想法修正为"><script>alert('XSS')</script><!—



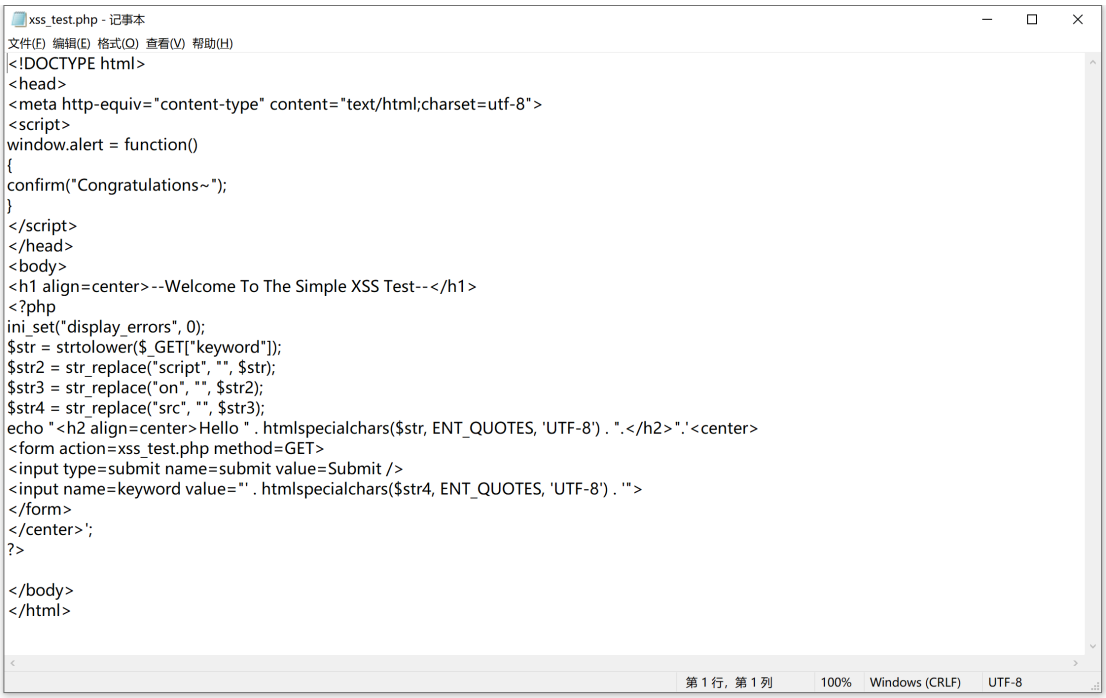
攻击成功。

3. 尝试更安全的过滤机制。

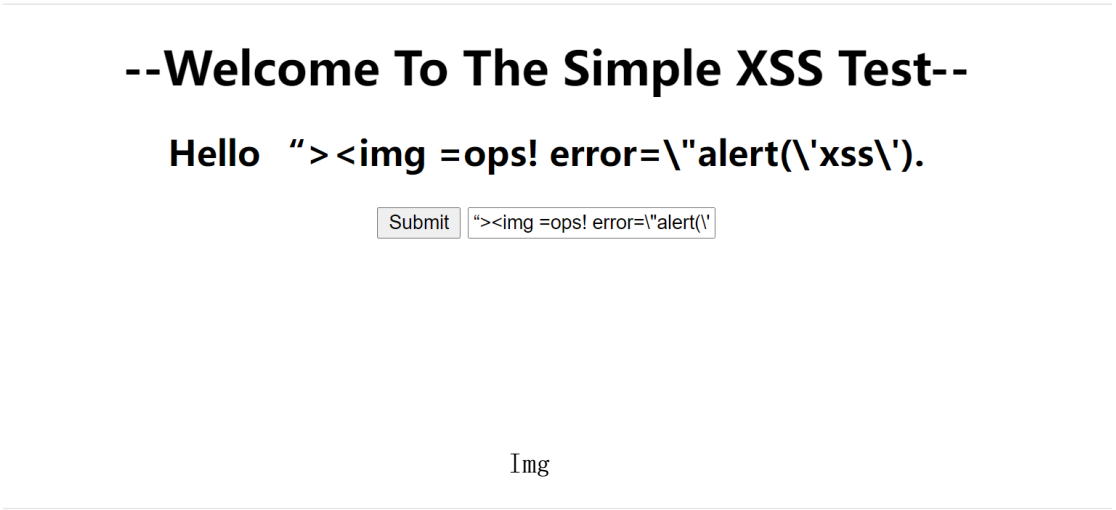
在 PHP 代码中，可以通过调用 htmlspecialchars 函数对用户输入的字符串进行处理，将与 HTML 语言冲突的字符（如<, >, &, ", '）进行特殊编码，从而避免这些字符在 HTML 中被解释为代码的一部分。

这种机制可以有效地防止 XSS（跨站脚本攻击）等安全问题，通过将潜在的恶意代码转化为普通文本，确保网页的安全性和稳定性。我尝试改进代码，改进后的代码使用了 HTMLPurifier 来清理用户输入，并确保所有用户输入都经过严格的安全处理。这样可以有效防止大部分的 XSS 攻击。

我们根据这个原理，更改 xss_test.php 如下：



然后我们打开网页进行之前的两种攻击：



Img

Script

可以发现，攻击均被成功防止了。可见 htmlspecialchars 函数的安全防护效果，成功设计了一种更安全的过滤机制。

心得体会：

在本次实验中，涉及了关于 XSS（跨站脚本攻击）的防御以及 PHP 中的安全编码实践，我实践并领悟到了 XSS 攻击的原理等知识：

XSS 攻击原理：实验中深入理解了 XSS 攻击的原理，即攻击者通过注入恶意脚本来利用网站的漏洞，使用户在浏览器中执行恶意代码。

XSS 攻击类型：学习了反射型 XSS 和存储型 XSS 两种主要类型的攻击，以及如何防范它们。

在网络安全方面，我深刻认识到 XSS 攻击对网站和用户的危害，学会了如何采取措施来防范这种类型的攻击，并且通过动手实践 img 和 script 两种攻击方式，和设计更安全的过滤机制，增加了对实际应用中可能遇到的安全问题的经验积累，提高了解决安全问题的能力。