《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 图形用户界面, 文本, 应用程序

   描述已自动生成
2. 然后，我们登录127.0.0.1/attack\_web/xss\_test.php，可以看到如下界面：

图形用户界面, 文本, 电子邮件

描述已自动生成

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）

图形用户界面, 文本, 应用程序

描述已自动生成

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”：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

然后，我们重新进行攻击：

图形用户界面, 应用程序

描述已自动生成

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><!—

图形用户界面, 文本, 应用程序

描述已自动生成

攻击成功。

1. 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：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

然后我们打开网页进行之前的两种攻击：

图形用户界面, 文本, 应用程序

描述已自动生成

Img

图形用户界面, 文本, 应用程序

描述已自动生成

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。