

实验 1 XSS 漏洞分析与防范

一、实验题目

请依据课堂所讲解的关于 XSS 攻防的技术知识，结合你所学过的 Web 前端编程技术，自行设计一套 XSS 攻防流程。要求：

- 1、攻防技术的实现既要有深度也要有广度。
- 2、需要设计一个前端页面。

二、相关知识

XSS 攻击(Cross Site Scripting)通常指的是通过利用网页开发时留下的漏洞，通过巧妙的方法注入恶意指令代码到网页，使用户加载并执行攻击者恶意制造的网页程序。

这些恶意网页程序通常是 JavaScript，但实际上也可以包括 Java、VBScript、ActiveX、Flash 或者甚至是普通的 HTML。

攻击成功后，攻击者可能得到包括但不限于更高的权限（如执行一些操作）、私密网页内容、会话和 cookie 等各种内容。

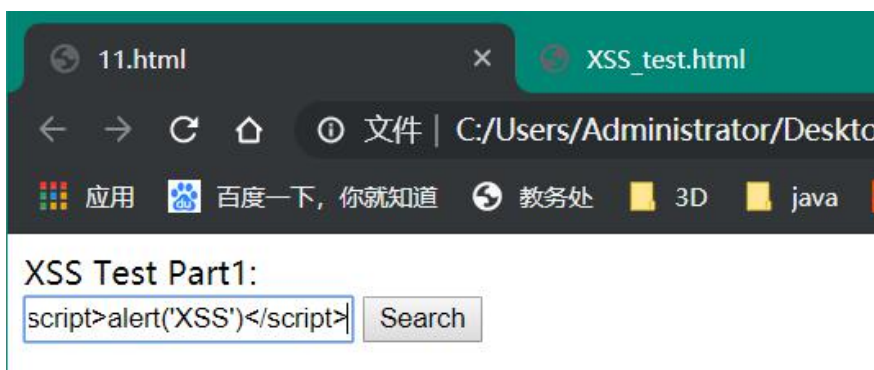
攻击原理：

HTML 是一种超文本标记语言，通过将一些字符特殊地对待来区别文本和标记，例如，小于符号（<）被看作是 HTML 标签的开始，<title>与</title>之间的字符是页面的标题等等。当动态页面中插入的内容含有这些特殊字符（如<）时，用户浏览器会将其误认为是插入了 HTML 标签，当这些 HTML 标签引入了一段 JavaScript 脚本时，这些脚本程序就将会用户在用户浏览器中执行。所以，当这些特殊字符不能被动态页面检查或检查出现失误时，就将会产生 XSS 漏洞。

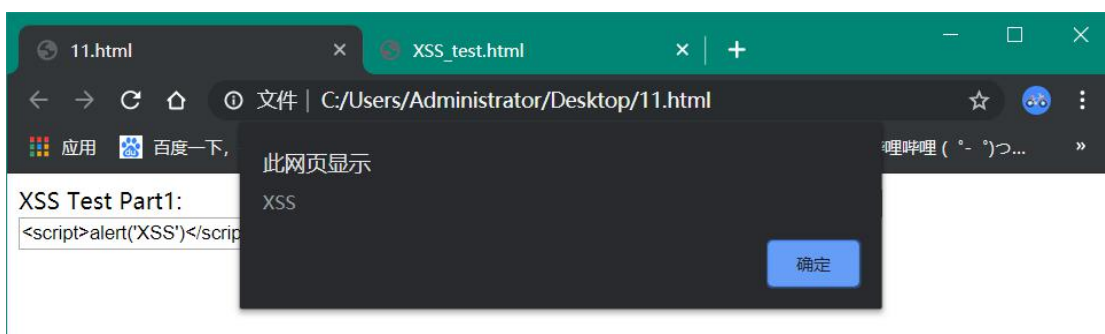
例如：当我的 HTML 代码中对用户输入的字符没有筛选 XSS 防御时

代码如下

当我们在对话框中输入含有<>的内容时,如: <script>alert('XSS')</script>



则这些脚本程序会在浏览器中执行



而当页脚本代码变为



攻击类型

XSS 有三类：反射型 XSS (非持久型)、存储型 XSS (持久型) 和 DOM XSS

1、反射型 XSS

发出请求时，XSS 代码出现在 URL 中，作为输入提交到服务器端，服务器端解析后响应，XSS 代码随响应内容一起传回给浏览器，最后浏览器解析执行 XSS 代码。这个过程像一次反射，故叫反射型 XSS。

2、存储型 XSS

存储型 XSS 和反射型 XSS 的差别仅在于，提交的代码会存储在服务器端（数据库，内存，文件系统等），下次请求目标页面时不用再提交 XSS 代码。

最典型的例子是留言板 XSS，用户提交一条包含 XSS 代码的留言存储到数据库，目标用户查看留言板时，那些留言的内容会从数据库查询出来并显示，浏览器发现有 XSS 代码，就当做正常的 HTML 与 JavaScript 解析执行，于是触发了 XSS 攻击。

3、DOM XSS

DOM XSS 和反射型 XSS、存储型 XSS 的差别在于 DOM XSS 的代码并**不需要服务器参与**，触发 XSS 靠的是浏览器端的 DOM 解析，完全是客户端的事情。

三、实验步骤与结果分析

1 前端页面

- ① `<input type="text" id='text1'>` 建立一个输入框
- ② 建立一个按钮，点击后使输入框中内容显示在页面上

```
<button onclick="search1()">Search</button><br>
```

```
<script>
```

```
function search1()
```

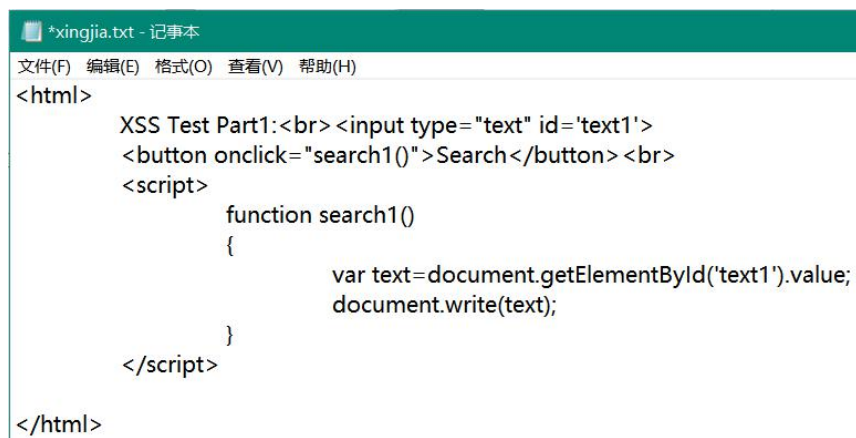
```
{
```

```
var text=document.getElementById('text1').value;
```

```
document.write(text);
```

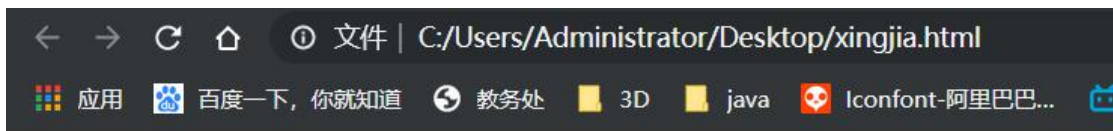
```
}
```

```
</script>
```



```
*xingjia.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

<html>
  XSS Test Part1:<br><input type="text" id='text1'>
  <button onclick="search1()">Search</button><br>
  <script>
    function search1()
    {
      var text=document.getElementById('text1').value;
      document.write(text);
    }
  </script>
</html>
```



XSS Test Part1:

XSS Test Part2:

XSS Test Part3:

XSS Test Part4:

2 完善 XSS 攻防

转换部分字符

```
XSS Test Part3:<br><input type="text" id='text3'>
<button onclick="search3()">Search</button><br>
<script>
    function search3()
    {
        var text=document.getElementById('text3').value;
        var newText=text.replace('&','&amp;');
        newText = newText.replace('<','&lt;');
        newText = newText.replace('>','&gt;');
        newText = newText.replace('"','&quot;');
        newText = newText.replace(' ','&nbsp;');
        document.write(newText);
    }
</script>
```

结果

<script>alert('XSS')

四、实验收获与总结

本次实验通过对 XSS 的攻击防御对网络的安全有部分的了解,对各种常见 XSS 攻击的尝

试，最常见的为两种，一种是反射型 XSS，通过发送链接形式，当用户点开时实施攻击；另一种是存储型 XSS，将恶意 js 代码上传或存储到漏洞服务器中，只要受害者浏览包含此恶意 js 代码的页面就会执行恶意代码。