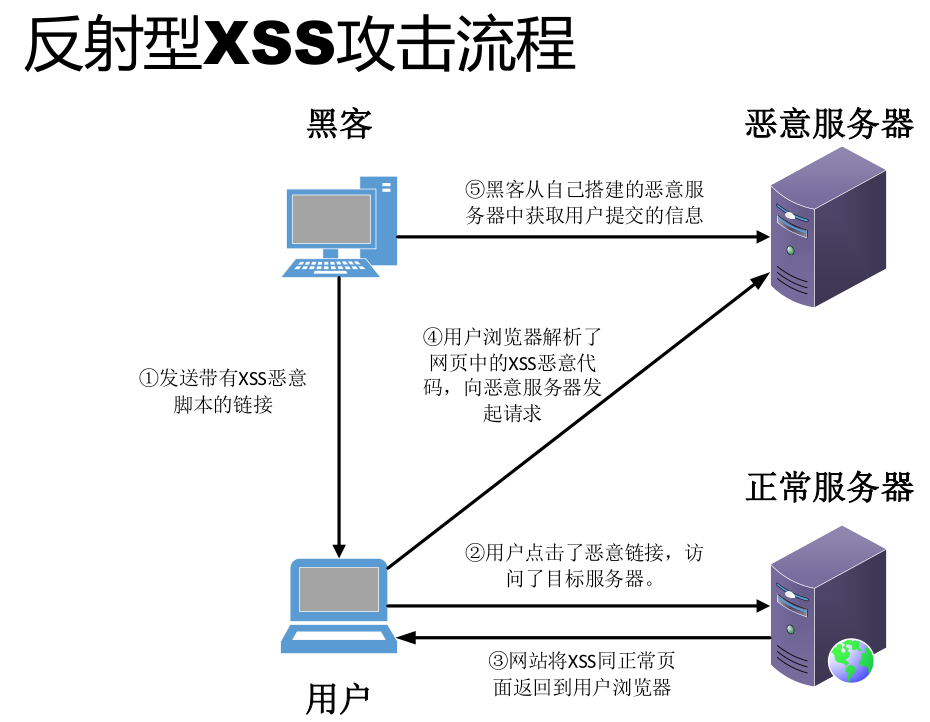
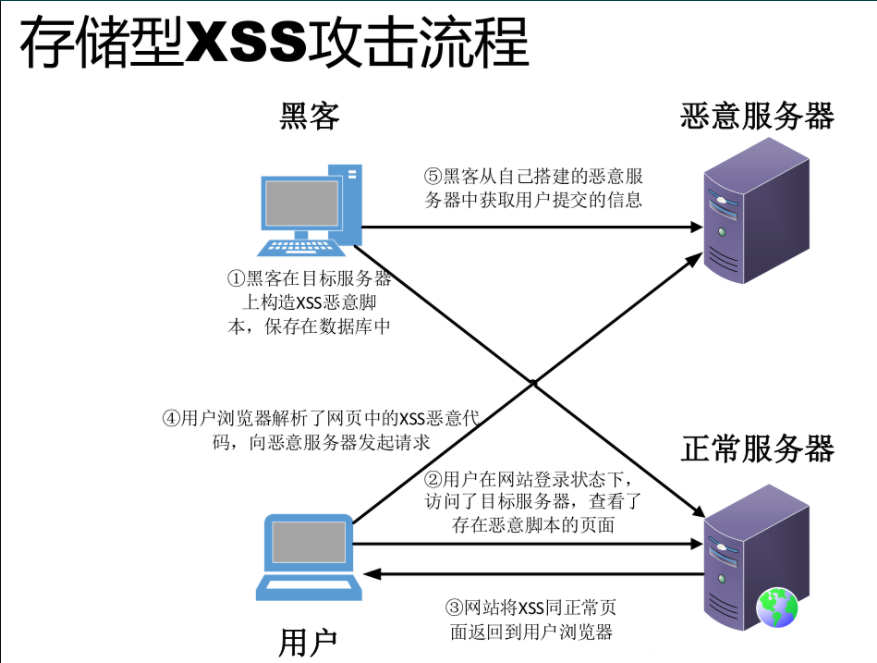
Web攻防-XSS跨站&CSP策略&HttpOnly属性&Filter代码&符号标签&AI绕过&工具项目

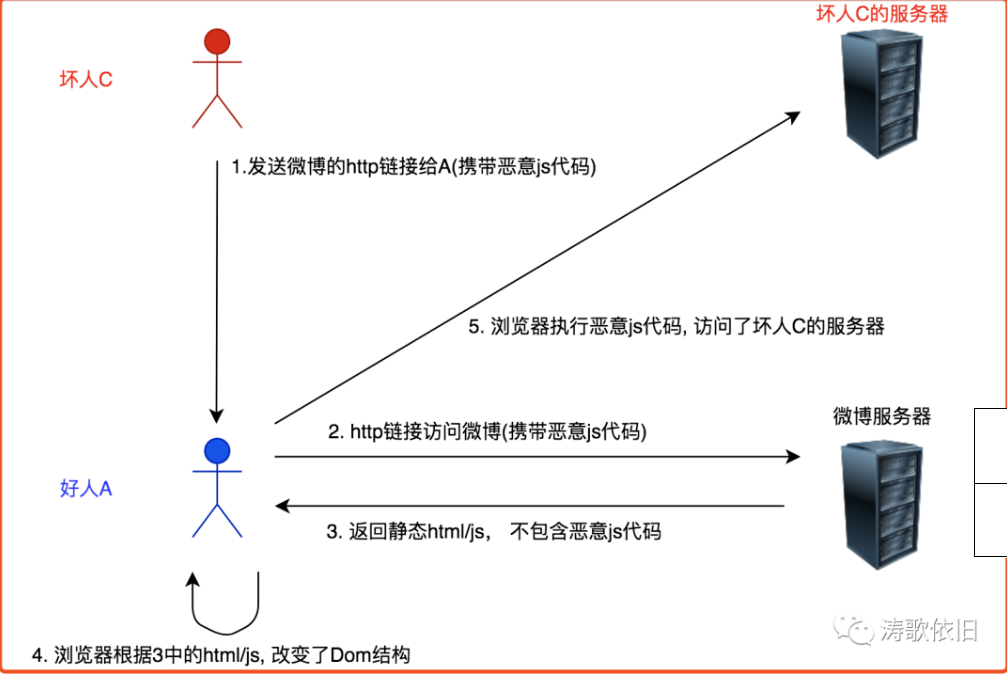


#知识点：  
1、Web攻防-SQL注入-产生原理&应用因素  
2、Web攻防-SQL注入-各类数据库类型利用  
----------------------------------------------  
1、Web攻防-SQL注入-参数类型\*参数格式  
2、Web攻防-SQL注入-XML&JSON&BASE64等  
3、Web攻防-SQL注入-数字字符搜索等符号绕过  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-HTTP头&UA&Cookie  
3、Web攻防-SQL注入-HTTP头&XFF&Referer  
----------------------------------------------  
1、Web攻防-SQL注入-操作方法&增删改查  
2、Web攻防-SQL注入-布尔&延时&报错&盲注  
----------------------------------------------  
1、Web攻防-SQL注入-高权限用户差异  
2、Web攻防-SQL注入-跨库&文件读写带外  
----------------------------------------------  
1、Web攻防-SQL注入-堆叠&二次注入  
2、Web攻防-SQL注入-SQLMAP进阶使用  
----------------------------------------------  
1、Web攻防-XSS跨站-原理&分类&利用点  
2、Web攻防-XSS跨站-反射&存储&DOM型  
----------------------------------------------  
1、Web攻防-XSS跨站-文件类型-html&pdf&swf&svg  
2、Web攻防-XSS跨站-功能逻辑-postMessage&localStorage  
----------------------------------------------  
1、Web攻防-XSS跨站-浏览器&转换-UXSS&MXSS  
2、Web攻防-XSS跨站-框架和库-VUE&React&Electron&JQuery  
----------------------------------------------  
1、Web攻防-XSS跨站-手工代码&框架工具&在线平台  
2、Web攻防-XSS跨站-Cookie盗取&数据提交&网络钓鱼  
----------------------------------------------  
1、Web攻防-XSS跨站-安全防护&CSP&Httponly&WAF等  
2、Web攻防-XSS跨站-工具项目&XSStrike&Chypass\_pro  
   
#章节点：(待补充)  
OWTOP10，SQL注入，文件安全（上传,下载,读取,删除,包含等），目录遍历，  
CSRF，SSRF，CRLF，SSTI注入，XML&XXE安全，RCE执行（代码或命令等）  
反序列化，业务逻辑（验证码，接口枚举，支付购买，机制验证，越权，并发等）  
未授权访问，失效访问控制，弱口令安全，第三方组件安全，CORS，JSONP等









演示案例：

* WEB攻防-XSS跨站-原理&分类&利用点
* WEB攻防-XSS跨站-反射&存储&DOM型
* WEB攻防-XSS跨站-SWF&SVG&PDF&HTML等
* WEB攻防-XSS跨站-PostMessage&LocalStorage
* WEB攻防-XSS跨站-框架&库&浏览器&插件&突变
* WEB攻防-XSS跨站-MXSS&UXSS&框架版本&写法
* WEB攻防-XSS跨站-手工代码&在线平台&Beef工具
* WEB攻防-XSS跨站-Cookie盗取&数据提交&网络钓鱼
* WEB攻防-XSS跨站-安全防护&CSP&Httponly&WAF等
* WEB攻防-XSS跨站-工具项目&XSStrike&Chypass\_pro

#XSS跨站-输入输出-原理&分类&利用点  
漏洞原理：接受输入数据，输出显示数据后解析执行  
基础类型：反射(非持续)，存储(持续)，DOM-BASE  
拓展类型：jquery，mxss，uxss，pdfxss，flashxss，上传xss等  
业务类型：localStorage，postMessage，React，Electron，VUE等  
常用标签：<https://www.freebuf.com/articles/web/340080.html>  
攻击利用：盲打，COOKIE盗取，凭据窃取，页面劫持，网络钓鱼，权限维持等  
安全修复：字符过滤，实例化编码，http\_only，CSP防护，WAF拦截等  
测试流程：看输出想输入在哪里，更改输入代码看执行（标签，过滤决定）  
   
#XSS跨站-分类测试-反射&存储&DOM型  
-数据交互的地方  
 get、post、headers  
 反馈与浏览  
 富文本编辑器  
 各类标签插入和自定义  
-数据输出的地方  
 用户资料  
 数据输出  
 评论，留言等  
 关键词、标签、说明  
 文件上传  
   
#CSP  
内容安全策略是一种可信白名单机制，来限制网站中是否可以包含某来源内容。  
该制度明确告诉客户端，哪些外部资源可以加载和执行，等同于提供白名单，  
它的实现和执行全部由浏览器完成，开发者只需提供配置。  
禁止加载外域代码，防止复杂的攻击逻辑。  
禁止外域提交，网站被攻击后，用户的数据不会泄露到外域。  
禁止内联脚本执行（规则较严格，目前发现 GitHub 使用）。  
禁止未授权的脚本执行（新特性，Google Map 移动版在使用）。  
合理使用上报可以及时发现XSS，利于尽快修复问题。  
   
实验：  
1、设置任何域JS加载  
2、设置本地域JS加载  
绕过利用条件：文件上传JS绕过  
3、设置本地域JS外加限制目录  
绕过利用条件1：目录有302跳转绕过  
绕过利用条件2：文件上传JS（指定目录）  
4、其他绕过技术  
参考：<https://xz.aliyun.com/news/11816>  
   
#http\_only  
一个可以在设置Cookie时使用的标记;当一个Cookie被标记为HttpOnly时，JavaScript无法访问该Cookie，只能通过HTTP（或HTTPS）协议传输。  
这一特性有助于防止跨站脚本攻击（XSS），因为攻击者无法通过JavaScript代码窃取存储在 HttpOnly Cookie中的敏感信息。  
实验：  
1、开启HttpOnly时XSS窃取Cookie的加载情况  
2、未开启HttpOnly时XSS窃取Cookie的加载情况  
绕过：有但鸡肋  
(1) CVE-2012-0053  
(2) PHPINFO页面  
(3) Flash/Java  
blog.csdn.net/weixin\_42478365/article/details/116597222  
思路：一般可不采用获取Cookie利用，采用方式（钓鱼，浏览器攻击框架等）  
   
#WAF或代码Filter  
黑盒XSS手工分析：  
-页面中显示的数据找可控的（有些隐藏的）  
-利用可控地方发送JS代码去看执行加载情况  
-成功执行即XSS，不能成功就看语句输出的地方显示（过滤）  
-根据显示分析为什么不能执行（实体化，符号括起来，关键字被删除等）  
   
1、人工分析  
演示：xss-lab（关卡）  
https://xz.aliyun.com/t/4067  
https://xz.aliyun.com/news/11816  
https://github.com/Re13orn/xss-lab  
blog.csdn.net/2301\_80031208/article/details/139159525  
   
2、工具项目  
https://github.com/s0md3v/XSStrike  
https://github.com/wxwhhh/Chypass\_pro  
演示：  
https://xss.xiejiahe.com/  
   
1、无任何过滤  
<script>alert()</script>  
   
2、实体化 输入框没有  
"> <script>alert()</script> <"  
   
3、全部实体化 利用标签事件 单引号闭合  
" onfocus=javascript:alert() "  
   
4、全部实体化 利用标签事件 双引号闭合  
" onfocus=javascript:alert() "  
   
5、事件关键字过滤 利用其他标签调用 双引号闭合  
"> <a href=javascript:alert()>xxx</a> <"  
   
6、利用大小写未正则匹配  
"> <sCript>alert()</sCript> <"  
   
7、利用双写绕过匹配  
"> <a hrehreff=javascript:alert()>x</a> <"  
   
8、利用Unicode编码  
&#x006a&#x0061&#x0076&#x0061&#x0073&#x0063&#x0072&#x0069&#x0070&#x0074&#x003a&#x0061&#x006c&#x0065&#x0072&#x0074&#x0028&#x0029  
   
9、利用Unicode编码（内容检测）  
&#x006a&#x0061&#x0076&#x0061&#x0073&#x0063&#x0072&#x0069&#x0070&#x0074&#x003a&#x0061&#x006c&#x0065&#x0072&#x0074&#x0028&#x0029;('http://')  
   
10、隐藏属性触发闭合  
<script>alert('xss')</script>&t\_sort=" type="text" onclick="alert('xss')  
   
11-20  
https://blog.csdn.net/l2872253606/article/details/125638898

涉及资源：[资源下载地址](https://docs.qq.com/doc/DQ3Z6RkNpaUtMcEFr)