浏览器 WEB 安全威胁检测技术研究与实现

江 导

（顺德职业技术学院 广东 528000）

【摘 要】WEB 浏览器是一种常见的客户端应用程序，是用户与网络交互的最主要平台之一，WEB 应用已经广泛应用到新闻资讯、电子商务、社交网络等多个领域，然而由于 WEB 应用程序功能性和交互性的不断增强，对应的 WEB 漏洞和恶意攻击层出不穷，现有的 WEB 安全措施主要集中于服务端，然而客户端的安全机制相对比较薄弱，因此，对于如何保证 WEB 应用的安全已成为安全界广泛关注的重点。本文主要研究浏览器端的 WEB 安全威胁检测技术与实现。

【关键词】浏览器；WEB 安全；安全威胁检测技术；同源策略；XSS 过滤器

中图分类号：TP393.08 文献标识码：A 文献编号：1009-6833（2014）02-100-02

Research and implementation of WEB browser security threat detection

Jiang Dao

Abstract: the WEB browser is a common client applications, is one of the most important platform for user interaction with the web, WEB has been widely applied to many fields of news and information, e-commerce, social networks, however, and interactive WEB applications to enhance the function, the corresponding WEB vulnerabilities and attacks emerge in an endless stream the existing WEB security measures, mainly concentrated in the server, the client security mechanism relatively weak, therefore, how to ensure the WEB application security has become the focus of attention of the security community. Technology and implementation of WEB security threat detection this paper mainly studies the browser.

Keywords: WEB browser; safety; safety threat detection technology; homologous strategy; XSS filter

1 WEB 浏览器简介

WEB 浏览器是一个软件程序，用于与 WWW 建立联结，并与之进行通信，它可以在 WWW 系统中根据链接确定信息资源的位置，并将用户感兴趣的信息资源取回来，对 HTML 文件进行解释，然后将文字图像显示出来，或者将多媒体信息还原出来。浏览器主要包括用户界面、浏览器引擎、网络、JS 解释器、数据存储等组件。目前典型的 WEB 浏览器有 Internet Explorer、360 极速浏览器、360 安全浏览器、搜狗浏览器、猎豹浏览器等，它们适用于各种不同的环境。

2 WEB 安全简介

WEB 漏洞：

WEB 应用程序的正常运行，涉及到客户端浏览器、网络协议传输、服务器响应、数据库查询等许多方面。其中无论哪一方面出现漏洞，均可能导致 WEB 安全问题。漏洞即某个程序（包括操作系统）在设计时未考虑周全，当程序遇到一个看似合理，但实际无法处理的问题时，引发的不可预见的错误。系统漏洞又称安全缺陷，如漏洞被恶意用户利用，会造成信息泄漏，如黑客攻击网站即利用网络服务器操作系统的漏洞。任何事物都非十全十美，作为应用于桌面的操作系统—Windows 以及运行于该环境中的 WEB 浏览器也是如此。这直接危害到我们使用计算机的安全行为，漏洞受病毒及恶意代码利用，容易导致巨大损失。OWASP 于 2010 年发布的 Top 10 应用程序，其中涉及到 WEB 应用程序的方面的有：客户端的注入漏洞、跨站脚本漏洞、服务端的授权管理、安全误配置、不安全的密码存储、网络传输层的失效的 URL 访问重定向、弱保护等。

3 浏览器 WEB 安全威胁检测与实现

3.1 XSS 动态检测技术

跨站脚本是服务端代码漏洞产生的问题，因此唯有从服务端入手才能彻底解决。下面我们就主要讨论浏览器 XSS 动态检测技术。

动态检测技术之所以称为“动态”，是指它不直接在文本层次分析可执行代码的行为，而是在代码运行时进行动态调试、分析。动态检测技术将代码语义分析工作交给现成的代码解析器，可以极大降低系统复杂度，避免语法语义分析不到位导致的误判。

在 XSS 攻击检测领域，动态检测意味着浏览器端 XSS 过滤器不再是一个相对独立的附加组件，而是与浏览器各组件结合更紧密的一个安全机制。从浏览器架构的角度看，动态检测技术工作在 HTML 解析器和 Javascript 解析器之间，即在 HTML 解析器生成的文档对象模型（Document Object Model，简称 DOM）树中检测脚本节点，完成后才将 DOM 树中的脚本结点传递给 JavaScript 引擎执行。因此，动态检测技术可以完全规避浏览器的 HTML 解析“怪癖”，直接在 DOM 树中命中 HTML 文档中的可执行脚本，准确率大幅提升。动态检测技术有其优点，动态检测技术在 DOM 树的基础上作检测，因此与静态检测技术相比，它最显著的优点就是不受浏览器解析怪癖的影响。无论多么复杂晦涩的 HTML 文档，只要浏览器能解析出 DOM 树，动态检测时就不会产生歧义。同时，动态检测直接从 DOM 树的脚本节点下手还有个优点，它无须重复扫描分析 HTML 文档。与传统的模拟解析方法相比，这可以提升一定的性能。有优点，自然也少不了缺点，XSSAuditor 假设服务端只采用简单的几种参数变换，并使用字符串精确匹配算法从 URL 中查找可疑脚本，这显然不能覆盖所有场景。一旦攻击者发现服务端采用了 XSSAuditor 未知的参数转换方法，就可以绕过它的检测。另外，检测策略不够完善，对间接脚本注入无能为力。常见的反射型 XSS 攻击都是将可执行脚本作为参数直接注入 HTML 响应中的，而 XSSAuditor 也作了针对性的匹配检测。然而，对于间接注入的恶意代码，

XSSAuditor 就无能为力了。

针对 XSSAuditor 未考虑复杂的服务端参数转换，攻击者可以针对存在复杂参数转换的服务端发起反射型 XSS 攻击。

例如，下面的服务端代码会把参数中的“you”改成“me”，然后返回给浏览器：

```
1<html>
2 <body> Hello
3 <?php
4 $name = $_GET [ 'name' ];
5 $new_name = str_replace（"you", "me", $name）;
6 echo $new_name;
7 ?> </body> </html>
```

正常的 URL 请求如下所示：

http：//l 27.0.0. l/xss5.php?name=you

服务端收到请求后，会将参数中的“you”替换成“me”，因此正常的 HTTP 回应内容如下段代码所示：

```
2 <body>
3 Hello
4 me</body>
5 </html>
```

然而，XSSAuditor 并不知道服务端有这样奇怪的转换，攻击者就可以构造一个恶意请求 URL：

http：//l 27.0.0.1 /xss5.php?name=<script>alert（'you'）；</script>

服务端收到请求后，仍然把参数中的“you”替换成“me”，于是 HTML 响应如下：

```
1 <html>
2 <body>
3 Hello
4 <3cripc>alert（'me'）；</3crxpt></body>
5 </html>
```

XSSAuditor 从 DOM 树中知道返回的 HTML 文档中存在一段 JS 脚本。但当它把这段脚本与恶意请求 URL 做匹配时，因为服务端把“you”替换成 “me”，匹配失败。XSSAuditor 未能识别它是一个恶意攻击。

3.2 XSSBreaker 的设计与实现

现有的浏览器端 XSS 检测技术均存在一定问题，这里将重新设计并实现一个浏览器端的 XSS 检测机制，新的 XSS 检测机制被命名为“XSSBreaker”。

3.2.1 XSSBreaker 的核心架构和工作流程

XSSBreaker 的核心是采用动态检测技术，这里着重参考 XSSAuditor 的架构。XSSBreaker 架构中 Web 服务器和浏览器之间通过 Internet 连接。深入到浏览器内部，XSSBreaker 是浏览器的一个安全组件，将与浏览器的 HTML 解析器、DOM 树、JavaScript 引擎等组件交互。HTML 解析器负责将 HTML 文档解析成 DOM 树，DOM 树由 HTML 解析器生成，JavaScript 引擎负责解释执行网页内容中的 JavaScript 脚本。在这个架构内，XSSBreaker 的主要工作流程是：浏览器向网站服务器提交一个 HTTP 请求，Web 服务端向浏览器返回一个 html 响应文档，浏览器调用 XSSBreaker 的 init 方法，进入 XSS 检测初始化流程，浏览器内置的 HTML 解析器开始解析 html 响应文档，HTML 解析器生成文档对应的 DOM 树，其中包含文本节点和脚本节点，XSSBreaker 进入检测流程，检测每一个脚本节点，XSSBreaker 使用字符串近似匹配算法，在脚本中查找 GET/POST 参数，DOM 树的剩余脚本节点传递给 JavaScript 引擎执行，若 JavaScript 引擎遇到 evalO 之类的动态函数，那么新脚本也将被传递到 XSSBreaker 的 check（）方法，重复前面两步，若文档通过 document.writeO 方法或 DOM 节点的 innerHTML 属性产生了新的 HTML 文档，那么新内容也将被传递给 HTML 解析引擎，即重复第 5 步之后的步骤。

3.2.2 XSSBreaker 的实现

XSSBreaker 是基于 Firefox 浏览器的内容安全策略（Content Security Policies，简称 CSP）实现的。XSS 检测策略可以分为内联策略、外部策略和白名单检测策略。内联策略用于阻挡集成在内联脚本的 XSS 攻击，外部策略应用于外部代码，如果外部脚本的 URL 指向的主机不是由参数提供，则可以信任，即使外部脚本的 URL 指向的主机是由参数提供的，只要该主机属于当前页面的可信域，则直接放行。白名单并不是孤立的策略，而是贯穿于 XSS 检测过程中，在参数匹配算法中，XSSBreaker 首先排除小于 8 字节的参数，因为它甚至不能构成一个脚本标签，不可能是 XSS 参数。这些白名单策略既可以减少检测时间，又能降低误报率，是优化 XSSBreaker 的重要手段。

内容安全策略是一套开发者工具集，用于帮助开发者预防多种类型的 Web 攻击。它支持让开发者预定义安全规则，以减少跨站脚本、点击劫持、数据包嗅探等攻击。当 CSP 检测到网页内容违反安全规则时，会将信息报告给网站开发者，以帮助其修复网站程序漏洞。

总之，本文通过论述跨站脚本检测技术的原理、优缺点，并构造实例演示了攻击方法。研究表明，目前已有的浏览器端跨站脚本检测技术存在诸多安全缺陷，而且未能在安全性、兼容性、性能之间取得较好平衡，为了提高浏览器整体安全性，我们设计 XSSBreaker 来检测和实现浏览器的 WEB 标准的改进的兼容性。这不仅能让安全站点更容易部署，而且能够减少试图使网页在所有平台正确显示所花的时间。

参考文献：

[1]吴健君，周兵.浏览器/Web 服务器与 C-B-S 模式及其应用.

[2]国家互联网应急中心，2010 尔中国互联网网络安全报告，

[3]陈爱华.浏览器 Web 安全威胁检测技术研究与实现.北京邮电大学.2013（01）

[4]刘大勇.Web 的安全威胁与安全防护. 大众科技.2005(03)

[5]曾平.基于浏览器嵌入规则的非安全 JavaScript 检测与分析. 湖南大学 2011（05）

[6]王欣 .WEB 应用系统安全检测关键技术研究. 北京邮电大学.2011（05）

[7]韦银.浅谈 Web 的安全威胁与安全防护. 科学咨询（决策管理）.2010（01）

[8]刘庆平.浏览器安全问题的研究与解决方案. 上海交通大学.2011（10）

作者简介：

江导（1979—），男，汉，广东兴宁人，顺德职业技术学院讲师，硕士研究生，研究方向：信息集成，人工智能。

浏览器WEB安全威胁检测技术研究与实现

作者： 江导， Jiang Dao
作者单位： 顺德职业技术学院 广东528000
刊名： 网络安全技术与应用
英文刊名： Network Security Technology & Application
年，卷(期)： 2014(2)

参考文献(8条)

1. 吴健君;周兵 浏览器/Web服务器与C-B-S模式及其应用

2. 国家互联网应急中心 2010尔中国互联网网络安全报告

3. 陈爱华 浏览器Web安全威胁检测技术研究与实现 2013

4. 刘大勇 Web的安全威胁与安全防护 2005(03)

5. 曾平 基于浏览器嵌入规则的非安全JavaScript检测与分析 2011

6. 王欣 WEB应用系统安全检测关键技术研究 2011

7. 韦银 浅谈Web的安全威胁与安全防护 2010(01)

8. 刘庆平 浏览器安全问题的研究与解决方案 2011