<<AJAX安全技术>>

图书基本信息

书名：<<AJAX安全技术>>

13位ISBN编号：9787121079306

10位ISBN编号：7121079305

出版时间：2009-1

出版时间：电子工业出版社

作者：霍夫曼

页数：403

前言

在AJAX的面前，火、车轮、电的意义都显得淡然无光。
从AJAX诞生的那一刻起，人们终于实现了梦寐以求的理想——在Web应用程序中刷新局部页面。
相信那一天JamesGarrett站在浴室里时，一定是上帝给了他灵感才想到了这个词——AJAX。
但是像毁灭阿兹特克（Aztecs）的西班牙殖民者科尔蒂斯（Cortés），或者像《星球大战（StarWar）》前传中，最初被认为是救世主、最终却带来毁灭的达斯●维达一样，很多起初认为是美好的事物结局却不一定很好。
同样，对于AJAX来说，其安全上的巨大漏洞已经成为自身的最大威胁，如果任其发展，最终也会同前面的二者一样，产生混乱并最终毁于一旦。
在这巨大的恐惧面前，为了保护无辜的人们、战胜邪恶并重新恢复宇宙中的和平，终于有两个人站了出来，他们便是Billy和Bryan。
衷心感谢你阅读这本书。

<<AJAX安全技术>>

内容概要

一本防范AJAX安全漏洞的实用指南。

如今，越来越多的网站都被改写成AJAX应用程序，甚至传统的桌面软件也通过AJAX，迅速转向了Web领域。

但是在这个过程中，人们通常都没有考虑到安全的问题。

如果不恰当地设计、编写了AJAX应用程序，那么它们会比传统桌面程序存在更多的安全漏洞。

AJAX开发人员无时无刻都希望有一本指南，能够指导他们如何来保护自己的应用程序——他们终于等到了这一天。

《AJAX安全技术》一书，系统地分析了当今最危险的AJAX漏洞用现实中的代码阐述了大量关键性的安全理念，并对实际中的案例，例如MySpace的Samy蠕虫病毒，进行了详尽分析。

更重要的是，不管你使用何种主流的Web编程语言和环境，例如 .NET、Java或PHP，本书都给出了许多具体、前沿的建议。

通过本书你将了解到以下几点： 如何刚氐AJAX特有的安全风险，包}舌过度细分的Web服务、应用程序控制流程篡改以及对程序逻辑的操控。

如何预防针对AJAX的攻击手段，包括JavaScript劫持、持久化存储窃取以及对mashup程序的渗透。

如何避免基于XSS和SQL注入的攻击，包括由AJAX衍生出来的SQL注入攻击（只需要两次请求就可以暴露整个后台数据库）。

如何使用Google Gears和Doj0开发安全的离线AJAX应用程序。

如何发现Prototype、DWR及ASRNET AJAX等AJAX框架中的安全问题以及我们自己仍需实现哪些功能。

如何更安全地编写AJAX代码，如何确定并修改已有代码中的安全缺陷。

不管是编写或者维护AJAX应用程序的开发人员、架构师，还是打算或正在设计新AJAX程序的项目经理，以及包括QA和渗透测试人员在内的所有软件安全人士，《AJAX安全技术》一书都是必不可少的。

作者简介

Jesse James Garrett——Adaptive PATH公司主席及创始人。

Billy Hoffman是惠普安全实验室的首席安全研究员，专注于如何自动发掘Web应用程序中的漏洞。他经常在Black Hat、RSA、Toorcon、Shmoocon、Infosec及AJAXWorld等会议上发表演讲，也曾受到过FBI的邀请进行演讲。

<<AJAX安全技术>>

书籍目录

章节摘录

第1章 AJAX安全介绍　1.1 AJAx基础知识　1.1.1 什么是AJAX　通常来说，浏览器为了动态显示Web页面，会把整个页面通过一个请求发送给服务器。
服务端应用程序接收到响应后，会创建该页面的HTML代码，并返回给浏览器。
浏览器会丢弃掉当前的页面，并显示新的HTML页面。
这样，用户才能够在浏览器中看见新的页面，并与其进行交互。
虽然这个过程十分简单，但是却非常浪费资源。
通常，服务端为客户端生成的新页面几乎与被其丢弃的当前页面一样。
在网络中传输的整个页面占据了大量的带宽，而这根本就是毫无必要的。
同时，在请求处理的过程中，用户也无法再使用应用程序，他们不得不等着服务端返回响应信息。
当服务端最终将响应信息返回给浏览器时，浏览器在重新加载页面的同时，还不得不闪烁一下。
如果Web客户端只需要请求页面中的一小部分，而不是向服务端发送整个页面的请求，那么对各个方面都会带来益处。
服务端可以更快速地处理请求，并且发送响应信息时会占用更少的带宽。
同时，客户端由于请求的整体时间变短，不仅可以为用户提供交互性更好的界面，也可以消除由重新加载整个页面而导致的闪烁。
应该说，AJAx是为了解决该问题而产生的众多技术的融合，它使得Web应用程序的客户端可以不断地从Web服务端更新部分页面。
而用户也不必再提交表单，或者离开当前的页面。
客户端的脚本代码（通常都是JavaScript）可以向页面的部分片段（Fragment）发起异步的，或者非阻塞（Non-blocking）的请求。
这些片段可以是一些原始的数据，在客户端再被转换成HTML代码；也可以本身就是HTML代码，直接被插入到浏览器的文档（Document）对象中。
不管怎样，在服务端完成对请求的处理，并将响应片段返回给客户端浏览器之后，客户端的脚本代码都会使用这些数据来修改页面中的文档对象模型（Document Object Model，DOM）。

版权说明