

《AJAX安全技术》一书对AJAX安全这一未开发领域进行了非常严谨、彻底的探讨。每个AJAX工程师都应该去掌握本书中的知识——至少应该明白其中的原理。



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

AJAX 安全技术

AJAX SECURITY



[美] Billy Hoffman Bryan Sullivan 著

张若飞 主 译

飞思科技产品研发中心 监制



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



AJAX 安全技术

AJAX SECURITY



[美] Billy Hoffman Bryan Sullivan 著
张若飞 主 铮 译
飞思科技产品研发中心 监制

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

Authorized translation from the English language edition, entitled AJAX SECURITY, First Edition, 0321491939 by Billy Hoffman and Bryan Sullivan, published by Pearson Education, Inc, publishing as Addison Wesley Professional, Copyright ©2008 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

CHINESE SIMPLIFIED language edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY Copyright ©2009v.

本书简体中文版由电子工业出版社和 Pearson Education 培生教育出版亚洲有限公司合作出版。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书简体中文版贴有 Pearson Education 培生教育出版集团激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字: 01-2008-5667

图书在版编目(CIP)数据

AJAX 安全技术 / (美)霍夫曼(Hoffman,B.), (美)苏里沃(Sullivan,B.)著;张若飞,王铮译. — 电子工业出版社, 2009.1

(网络安全专家)

书名原文: AJAX SECURITY

ISBN 978-7-121-07930-6

I. A… II. ①霍…②苏…③张…④王… III. 计算机网络—程序设计—安全技术 IV. TP393.09

中国版本图书馆 CIP 数据核字(2008)第 188773 号

责任编辑: 杨 鹂

印 刷: 北京机工印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 720×1000 1/16 印张: 26.5 字数: 691.2 千字

印 次: 2009 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 55.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

反侵权盗版声明

电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

“《AJAX 安全技术》一书对 AJAX 安全这一未开发领域进行了非常严谨、彻底的探讨。每个 AJAX 工程师都应该去掌握本书中的知识——至少应该明白其中的原理。”

Jesse James Garrett

“终于，我们等来了这本既通俗易懂，又囊括了多方面 AJAX 安全问题的书籍。在这之前，许多人根本没有考虑安全的问题，就直接加入了 AJAX 的潮流中。而现在，是那些人阅读这本书的时候了，并且应该根据读者在书中指出的安全缺陷，重新检查他们的应用程序。”

Jeff Forristal

“如果你正在编写或者检查 AJAX 代码，那么一定需要参考这本书。Billy 和 Bryan 已经在这个尚未探索过的领域进行了大量的工作，并取得了一定的成功。我已经迫不及待想去买这本书了。”

Andrew van der Stock, OWASP¹ 执行主席

“像 AJAX 这样的 Web 技术正在创造一种新的网络商业结构，并且解决了新经济体系中的矛盾。但是，由于技术本身的缺点及开发者的粗心而造成的安全问题，却使得这些进步大打折扣。至今为止，很少有书籍能够全面阐述 AJAX 的安全问题，并教导那些正在使用及即将使用 AJAX 的人如何去正确地编写 AJAX 代码。而这正是本书的目的。”

管理伙伴, Trellum 科技股份有限公司²

¹ OWASP (Open Web Application Security Project), 开放式 Web 应用程序安全项目，是全球性的安全咨询组织。

² Trellum Technologies, Inc. 是位于纽约的一家信息安全咨询公司。

Billy Hoffman

HP 软件公司 HP 安全实验室的主要研究员。在 HP, Billy 主要关注于 JavaScript 源代码分析、Web 应用程序安全漏洞的自动监测, 以及 Web 爬虫技术。自从 2001 年他为《黑客季刊》³ 写了一篇关于如何破解软件的文章, 并且发现人们对此非常感兴趣之后, 便一直从事关于安全领域的工作。这几年, Billy 参与过各种各样的项目, 包括对文件格式的反向工程、微控制器、JavaScript 恶意软件及磁条 (Magstripe)。他创建了 Stripe Snoop——一套能够捕获、修改、验证、生成、分析并且共享磁条中数据的研究工具。Billy 的成果已经应用于 Make 杂志、Slashdot、G4 Tech TV 及其他众多媒体和 Web 站点中。

Billy 经常会在 Toorcon、Shmoocon、Phreaknic、Summercon 及 Outerz0ne 等黑客会议上发表演讲, 并且是东南部黑客中的一位积极分子。偶尔, 他也会脱掉“黑色”的 T 恤, 作为安全专家参加 RSA、Infosec、AJAXWorld 和 Black Hat 等知名安全会议。

Billy 于 2005 年从佐治亚理工学院 (Georgia Institute of Technology) 以计算机学士学位毕业, 其在读时专业为网络与嵌入式系统。现在, 他和妻子, 以及两只又肥又懒的猫一起居住在亚特兰大。

Bryan Sullivan

HP 软件公司应用程序安全中心的软件开发经理。他曾经是一名专业的软件开发人员, 并且有着超过 12 年的开发管理经验, 而最近 5 年他主要关注于互联网安全软件行业。在进入 HP 之前, Bryan 曾是 SPI Dynamics⁴ 公司的一名安全研究员。在 SPI 的时候, 他开发了一个名为 DevInspect 的产品, 可以在 Web 应用程序的开发阶段分析可能存在的安全漏洞。

Bryan 经常会在 AJAXWorld、Black Hat 及 RSA 等业界会议上发表演讲。他参与了应用程序安全漏洞描述语言 (Application Vulnerability Description Language, AVDL) 的制定, 并且在安全评估和处理方法论上有 3 项正在审核的专利。他以科学学士学位毕业于佐治亚理工学院的应用数学专业。

Bryan 将他的业余时间大多花在打高尔夫球上, 如果读者中有奥古斯塔国家高尔夫夜总会 (Augusta National)⁵ 的成员, 那么 Bryan 非常乐意在一两个回合内, 告诉你他对 AJAX 安全所知道的一切。

³ 由美国黑客戈德斯坦创办的著名黑客杂志。

⁴ SPI Dynamics 公司位于亚特兰大, 是领先的网络应用安全评估软件和服务供应商, 于 2007 年被惠普收购。

⁵ 位于美国佐治亚州的著名高尔夫球场, 每年都会举办著名的高尔夫球名人赛。

在 AJAX 的面前，火、车轮、电的意义都显得淡然无光¹。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 James Garrett 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²（Aztecs）的西班牙殖民者科尔蒂斯（Cortés），或者像《星球大战（Star War）》³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。

¹ 此处作者使用了夸张手法。

² 中世纪时代墨西哥中央高原建立的庞大帝国，最终被西班牙殖民者毁灭。

³ 美国导演乔治·卢卡斯著名的系列电影。



关于飞思

我们经常感谢生活的慷慨，让我们这些原本并不同源的人得以同本，为了同一个梦想走到一起。

因为身处科技教育前沿，我们深感任重道远；因为伴随知识更新节奏，我们一刻不敢停歇。虽然我们年轻，但我们拥有：

“严谨、高效、协作”的团队精神

全方位、立体化的服务意识

实力雄厚的作者群和开发队伍

当然，最重要的是我们拥有：

恒久不变的理想和永不枯竭的激情和灵感

正因如此，我们敢于宣称：

飞思科技=丰富的内容+完美的形式

这也是我们共同精心培育的品牌的承诺。
www.fecit.com.cn

“问渠哪得清如许，为有源头活水来”。路再远，终需用脚去量；风景再美，终需自然抚育。

年轻的飞思人愿为清风细雨、阳光晨露，滋润您发芽、成长；更甘当坚实的铺路石，为您铺就成功之路。

飞思科技产品研发中心

联系方式

咨询电话：（010）68134545 88254161-67

电子邮件：support@fecit.com.cn

服务网址：<http://www.fecit.com.cn> <http://www.fecit.net>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

AJAX 已经完全改变了我们构建、部署 Web 应用程序的方式。对于那些运行在服务端的强大应用程序来说，浏览器只能作为其简单终端的日子已经一去不复返了。如今，不管是在客户端还是服务端，AJAX 应用程序已经能够在用户的浏览器中，实现与桌面程序一样的功能了。从 Google 和 Yahoo! 这样的公司，以及那些推行用 AJAX 实现客户端存储、离线应用程序和富 Web API 的开源社区中，便可窥其一斑。

作为 Web 开发者以及安全研究者，我们都在马不停蹄、尽可能地学习这些新的应用程序和技术。但是，在为 AJAX 带来的新功能感到激动的同时，我们不禁又有一些困扰：从没有人讨论过这项新应用架构的安全问题。我们不断能看见网络上的一些资源，以及 AJAX 领域所谓的专家，给出的却都是一些很差的建议和代码示例，其中到处都是容易引起 SQL 注入或者跨站脚本攻击的安全漏洞。再往深处探索，我们发现除了这些通常被忽视的 Web 漏洞之外，在开发 AJAX 应用程序时还存在着更大的安全隐患。例如过度划分的 Web 服务（即 Web Service）、应用程序控制流程篡改、开发 mashup 程序时的不良习惯，以及容易让人忽视的身份认证（Authentication）机制。也许，AJAX 同时继承了桌面及 Web 应用程序的优点，但是它也同时具有二者的安全缺陷。但是，对于大多数开发者来说，安全已经被抛之脑后了。

我们希望能改变这个现状。

这本书主要针对那些想要在其应用程序中实现最新、最炫功能的 AJAX 开发者，并教会他们如何防范一些恶意黑客的攻击（不管他们出于个人还是金钱目的）。在本书中，我们不仅仅指出了可能存在的安全问题，同时也提供了这些问题的解决方法，从而让开发人员编写出更严谨、更安全的代码。除此之外，我们还分析了 Prototype、DWR 及 Microsoft ASP.NET AJAX 等常用 AJAX 框架所采用的安全保护机制，以及作为一名开发者可以从中借鉴的地方。

同样，这本书也针对质量保证（Quality Assurance）工程师及专业的安全渗透测试人员。我们试着提供一些 AJAX 应用程序中常见的漏洞和安全缺陷。书中讨论了在评估一个 AJAX 应用程序时，所要面临的测试挑战，例如如何发现跟踪应用程序，以及如何发现程序中的缺陷，并且介绍了一些辅助性的工具。最后，我们对 JavaScript 劫持、持久化存储窃取，以及攻击 mashup 程序等新的 AJAX 攻击

技术，进行了详细的介绍。对于一些常见的攻击，我们也对其进行了新的演绎，例如一次基于 AJAX 的 SQL 注入攻击，只需要两个请求就可以暴露出整个后台数据库。

本书并不是讲解如何进行 AJAX 或者 Web 编程——因此我们希望读者对这些方面已经有足够的了解。我们将精力集中在设计和创建 AJAX 应用程序时，容易产生安全问题的错误和问题上，并给出如何开发安全 AJAX 应用程序的建议。同样，本书并不是一本编程语言指南，也并不要求读者使用某种特定语言来编写服务端代码。在所有的 AJAX 应用程序中，都会有一些常用的部分，例如 HTTP、HTML、CSS 和 JavaScript，而这些也正是我们分析的重点。当我们对如何编写安全的服务端代码提出建议时，会使用正则表达式或者字符串操作等形式，这样读者可以用任何语言来实现。

本书中还包含了大量对开发和测试人员非常有用的资料。从对大量实际案例的分析中，读者可以了解到，现实中的 AJAX 应用程序是如何被攻破的，例如 MySpace 的 Samy 蠕虫攻击及 Yahoo! 的 Yamanner 蠕虫攻击。此外，书中还包含了许多示例程序，例如在线旅行预订网站，这些都为测试和开发等人员，如何构建安全的 AJAX 应用程序提供了指导。

虽然我们建议读者一页一页、从头至尾地阅读本书，但是每一个章节都是独立的一部分。如果读者急需查看某一方面的内容，例如对某个 AJAX 框架安全性的分析（参阅第 15 章“AJAX 框架分析”），可以直接跳至该章阅读。

AJAX 在创建 Web 应用程序方面，提出了许多令人激动的新理念，本书并没有从安全方面贬低或者磨灭其贡献的意思。相反，我们希望能够借助本书，帮助读者创建功能强大、丰富的 AJAX 应用程序，并且同时增强对恶意攻击的安全保护能力。

希望读者能够喜欢本书。

著 者

共同致谢

虽然本书封面所写的作者是 Billy Hoffman 和 Bryan Sullivan, 但是实际上其中倾注了许多颇具天赋, 并且甘愿奉献的人们的心血。如果没有他们的帮助, 整本书的内容读起来会更像是“我们很难保证 AJAX 的安全”。对于他们为本书花费的时间及提出的专业意见, 我们无法一一言谢, 但是在这里还是要一表谢意。

首先, 也是最重要的, 我们要感谢我们可爱、聪明和热情的妻子们, Jilly 和 Amy, 感谢她们去年一整年的支持。我们唯一可以想象的是, 当她们在说“赶快回去写书!”的时候是多么的困难, 因为她们实际想说的是“忘掉那本书, 赶紧过来吃晚饭!”。你们都是最令人着迷的女人, 是上帝赐予我们的礼物。

我们希望感谢本书的技术编辑, Trellum 技术股份有限公司的 Jeff Forristal、Joe Stagner 和 Vinnie Liu, 是你们让本书比我们预想的还要好, 出于这点原因, 你们再怎么严格也不为过。希望我们永远都是好朋友。

我们还想感谢 SPI 中的每一位成员, 感谢他们的贡献和理解。虽然他们中的很多人都给予过帮助, 但是有两个人需要特别感谢: Caleb Sima, 如果没有你无穷的智慧和根本不会有本书, 你建立了一个如此出色的公司, 我们非常荣幸能成为其中的一员; Ashley Vandiver, 你做了许多超出我们期望的工作。衷心感谢你们。

尤其要感谢 Samantha Black 为“Web 攻击”和“持久层攻击”两章给出的帮助。

最后, 我们要感谢 Addison-Wesley Professional 出版社及 Pearson 教育集团的人员, 是你们带来了本书生命: Sheri Cain、Alan Clements、Romny French、Karen Gettman、Gina Kanouse、Jake McFarland、Kathy Ruiz、Lisa Stumpf、Michael Thurston 及 Kristin Weinberger。我们尤其要感谢 Marie McKinley (以及 Black Hat 的成员), 感谢她对市场的准确把握; Linda Harrison, 感谢你让我们听上去更像专业的作者, 而不是计算机程序员; Chelsey Marti, 感谢你为编写一个能够被杀毒软件拦截的文档所付出的努力 (最终采用了 Rot-13 的加密方式)。最后, 但也是非常重要的,

我们要感谢组稿编辑 Jessica Goldstein，感谢对我们这两个新人的信任，以及帮助我们顺利完成整本书的编写。

这一切都源自一个头发又短又卷的孕妇，向我们问了这样一个问题“你们有没有想过写一本书？”不得不说，这对于我们是一段多么有趣、新奇的经历。

Billy 的致谢

感谢我的妻子 Jill。她总是在我将要放弃的时候给我动力，没有她我根本不可能完成这本书。

感谢我的父母，Mary 和 Billy，以及我的兄弟 Jason。如果没有他们坚定的爱与支持，也就不可能有今天的我。

当然，还要感谢本书的合著者 Bryan。在众多漫漫长夜，以及邻近交稿那几天的疯狂工作中，我们成了最亲密的朋友，这也是最值得我们骄傲的事情。我想不出本书我还能与谁一起合著。

Bryan 的致谢

我要感谢我的妻子 Amy，感谢她的爱与支持，不光是在写书的过程中，而是在一起度过的整整 14 年中。

最后，除了你 Billy，我想不出还有谁能同我一起在晚上与周末，一边喝着红牛饮料，一边争论不同 CSRF 防御策略的特点。虽然我们比预期花费了更多的精力与汗水，但是我们可以说，我们使整个一代程序员免于被后人指责。

第 1 章	AJAX 安全介绍	1
1.1	AJAX 基础知识	2
1.1.1	什么是 AJAX	2
1.1.2	动态 HTML (DHTML)	10
1.2	AJAX 架构 (Architecture) 的转变过程	11
1.2.1	胖客户端架构	11
1.2.2	瘦客户端架构	12
1.2.3	AJAX: 最适合的架构	14
1.2.4	从安全角度看胖客户端应用程序	15
1.2.5	从安全角度看瘦客户端应用程序	15
1.2.6	从安全角度看 AJAX 架构	17
1.3	一场完美的攻击风暴	17
1.3.1	不断增加的复杂度、透明度及代码量	18
1.3.2	社会学问题	20
1.3.3	AJAX 应用程序: 富有吸引力的、战略上的目标	21
1.4	本章小结	22
第 2 章	劫持	23
2.1	攻击 HighTechVactions.net	24
2.1.1	攻击票务系统	24
2.1.2	攻击客户端数据绑定	30
2.1.3	攻击 AJAX API	34
2.2	黑夜中的盗窃	39
第 3 章	Web 攻击	41
3.1	基本攻击分类	41
3.1.1	资源枚举	41
3.1.2	参数操纵	45
3.2	其他攻击	66
3.2.1	跨站请求伪造攻击	66
3.2.2	钓鱼攻击	68
3.2.3	拒绝服务 (Denial-of-Service, DoS)	68

3.3	保护 Web 应用程序免受资源枚举和参数操作的攻击	69
3.4	本章小结	70
第 4 章	AJAX 攻击层面	71
4.1	什么是攻击层面	71
4.2	传统 Web 应用程序的攻击层面	72
4.2.1	表单输入	73
4.2.2	cookie	74
4.2.3	报头	75
4.2.4	隐藏的表单输入	75
4.2.5	请求参数	76
4.2.6	上传文件	78
4.3	传统的 Web 应用程序攻击：一份成绩单	79
4.4	Web 服务的攻击层面	81
4.4.1	Web 服务的方法	81
4.4.2	Web 服务的定义	82
4.5	AJAX 应用程序的攻击层面	83
4.5.1	AJAX 应用程序攻击层面的来源	84
4.5.2	黑客的最爱	86
4.6	正确的输入验证	86
4.6.1	有关黑名单及其他补丁的问题	87
4.6.2	治标不治本	90
4.6.3	白名单输入验证	93
4.6.4	正则表达式	96
4.6.5	关于输入验证的其他想法	96
4.7	验证富客户端的用户输入	98
4.7.1	验证标记语言	98
4.7.2	验证二进制文件	100
4.7.3	验证 JavaScript 源代码	100
4.7.4	验证序列化数据	106
4.8	关于由用户提供的内容	109
4.9	本章小结	110

第 5 章 AJAX 代码的复杂性	111
5.1 多种语言和架构	111
5.1.1 数组索引	112
5.1.2 字符串操作	113
5.1.3 代码注释	115
5.1.4 事不关己，高高挂起	115
5.2 JavaScript 的怪异之处	117
5.2.1 解释，而不是编译	117
5.2.2 弱类型	118
5.3 异步性	120
5.3.1 竞争条件	120
5.3.2 死锁及哲学家用餐问题	124
5.3.3 客户端同步化	127
5.3.4 留意你所采纳的建议	128
5.4 本章小结	129
第 6 章 AJAX 应用程序的透明度	131
6.1 黑盒对白盒	131
6.1.1 示例: mylocalweatherforecast.com	133
6.1.2 示例: 用 AJAX 实现的 mylocalweatherforecast.com	135
6.1.3 对比结果	139
6.2 像 API 一样的 Web 应用程序	140
6.3 一些特殊的安全错误	141
6.3.1 不恰当的身份认证	141
6.3.2 过度细化服务端 API	143
6.3.3 在 JavaScript 中存储会话状态	146
6.3.4 与用户相关的敏感数据	147
6.3.5 包含在客户端的注释及文档	148
6.3.6 在客户端进行的数据转换	149
6.4 通过隐藏来保证安全	152
6.5 本章小结	154
第 7 章 劫持 AJAX 应用程序	155
7.1 劫持 AJAX 框架	155

7.1.1 意外的方法冲突	156
7.1.2 人为的方法冲突	158
7.2 劫持“即时”的 AJAX	163
7.3 劫持 JSON API	167
7.3.1 劫持对象定义	172
7.3.2 JSON 劫持的根源	173
7.3.3 如何防范 JSON 劫持	173
7.4 本章小结	176
第 8 章 攻击客户端存储	179
8.1 客户端存储系统概述	179
8.2 HTTP cookies	181
8.2.1 cookie 访问控制规则	183
8.2.2 HTTP cookie 的存储能力	188
8.2.3 cookie 的生命期	191
8.2.4 cookie 存储的其他安全问题	192
8.2.5 cookie 存储总结	193
8.3 Flash 本地共享对象	194
8.4 DOM 存储	201
8.4.1 会话存储	202
8.4.2 全局存储	204
8.4.3 DOM 存储的细节讨论	205
8.4.4 DOM 存储安全	207
8.4.5 DOM 存储总结	208
8.5 Internet Explorer userData	209
8.6 一般客户端存储的攻击和防范方法	214
8.6.1 跨域攻击	214
8.6.2 跨目录攻击	215
8.6.3 跨端口攻击	216
8.7 本章小结	216
第 9 章 离线 AJAX 应用程序	219
9.1 离线 AJAX 应用程序	219
9.2 Google Gears	220

9.2.1	Google Gears 内置的安全特性及其缺点	221
9.2.2	探索工作者池	224
9.2.3	泄露并篡改本地服务器 (Local Server) 中的数据	226
9.2.4	直接访问 Google Gears 数据库	229
9.2.5	SQL 注入和 Google Gears	230
9.2.6	客户端 SQL 注入有多危险	234
9.3	Dojo.Offline	236
9.3.1	保证密钥安全	237
9.3.2	保证数据安全	238
9.3.3	可作为密钥的良好密码	239
9.4	再论客户端输入验证	240
9.5	创建离线应用程序的其他方式	241
9.6	本章小结	242
第 10 章	请求来源问题	243
10.1	Robots、Spiders、Browsers 及其他网络爬虫	243
10.2	请求来源不确定性和 JavaScript	245
10.2.1	从 Web 服务器的角度看 AJAX 请求	246
10.2.2	是你自己, 还是貌似你的某人	249
10.2.3	使用 JavaScript 发送 HTTP 请求	251
10.2.4	在 AJAX 出现之前的 JavaScript HTTP 攻击	252
10.2.5	通过 XMLHttpRequest 窃取其他内容	254
10.2.6	实战结合 XSS/XHR 进行攻击	258
10.3	防范措施	260
10.4	本章小结	261
第 11 章	Web Mashup 和聚合程序	263
11.1	互联网上计算机可以使用的数据	263
11.1.1	20 世纪 90 年代早期: 人类 Web 的黎明	263
11.1.2	20 世纪 90 年代中期: 机器 Web 的诞生	264
11.1.3	2000 年左右: 机器 Web 逐渐成熟	266
11.1.4	可公用的 Web 服务	266
11.2	Mashup: Web 中的弗兰肯斯坦	268
11.2.1	ChicagoCrime.org	269