“本书中介绍的技术对测试人员而言并不是可有可无的——你必须掌握它们，本书会告诉你如何使用这些技术!”

—— Harry Robinson，Google

PEARSON Addison Wesley

# Web入侵安全测试与对策

Mike Andrews 著
James A. Whittaker
汪青青 译


Pearson Education
清华大学出版社

# Web 入侵安全测试与对策

Mike Andrews
James A. Whittaker
著
汪青青 译

清华大学出版社
北 京

EISBN: 0-321-36944-0


**图书在版编目（CIP）数据**

Web 入侵安全测试与对策 /（美）安德鲁（Andrews, M.），（美）惠特克（Whittaker, J. A.）著；汪青青译.
—北京：清华大学出版社，2006.10
书名原文：How to Break Web Software
ISBN 7-302-13874-5

Ⅰ. W…　Ⅱ. ①安… ②惠… ③汪…　Ⅲ. 计算机网络-安全技术　Ⅳ. TP393.08

中国版本图书馆 CIP 数据核字（2006）第 114501 号

**出 版 者**：清华大学出版社　　**地　　址**：北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编**：100084
**社 总 机**：010-62770175　　**客户服务**：010-62776969
**责任编辑**：常晓波
**印 装 者**：北京鑫海金澳胶印有限公司
**发 行 者**：新华书店总店北京发行所
**开　　本**：185 × 230　**印张**：12　**字数**：264 千字
**版　　次**：2006 年 10 月第 1 版　2006 年 10 月第1 次印刷
**书　　号**：ISBN 7-302-13874-5/TP · 8341
**印　　数**：1 ~ 3000
**定　　价**：29.00 元（含光盘）

# 前　言

很多人问起“How to Break...”系列丛书下一部的出版时间和内容。读者们的需求主要集中于 Web 应用程序的话题上。很多测试人员在这一领域工作，需要测试那些用 World Wide Web 上流传的语言和特定的协议编写的应用程序。

虽然在 *How to Break Software*（Addison-Wesley，2002）和 *How to Break Software Security*（Addison-Wesley，2003）两本书中介绍的测试方法很多也和这样的环境有关，但 Internet 上的应用程序却存在着一些独有的问题。本书沿用前面两本书的风格分析上述问题，同时更倾向于讨论 Web 应用程序安全方面的话题。

在开始介绍本书的内容之前，首先需要说明它不涉及哪些内容。虽然这本书和黑客方面的书籍有相关的部分，但我们并不想重写 *Hacking Exposed* 一书，因为我们的目的不是为了介绍如何攻击 Web 服务器或 Web 应用程序，而是如何测试 Web 应用程序以防范由这些错误产生的常见故障。

本书是为帮助软件的开发人员、测试人员、管理人员以及质量控制方面的专家防范黑客攻击而编写的一本书。

这样的重点必然意味着本书中包含了关于黑客技术的知识，毕竟只有了解对手的技术才能防范他们。但本书是关于测试而非攻击的，我们的目的是引导测试人员了解那些可能出现问题的应用程序以及如何修复它们的方法。

本书也不讨论如何构建一个正确的 Web 应用程序结构，以及如何编写 Web 应用程序的代码。这些问题有其他的出版计划，而且每一种 Web 开发平台也有自己的话题，像 *Innocent Code* 之类的书就专门讨论这些话题。不过本书倒是包含了很多*不要*怎样构建和编写 Web 应用程序的内容。聪明的 Web 开发人员会把它作为安全的 Web 编程的一部分参考知识。

本书主要是为了向测试人员介绍一些用于测试 Web 应用程序的攻击方式，其中会包含一些恶意输入的典型例子，比如一些躲避校验和身份认证的方式，以及某些由配置、语言或结构带来的问题。但这些介绍都很简单，同时给出了如何查找和测试这些问题，以及解决这些问题的方法建议。希望本书能够成为人们在测试基于 Web 的应用程序方面获取信息（和灵感）的有力工具。

祝你的 Web 测试成功！

Mike Andrews，加利福尼亚州奥伦奇县
James A. Whittaker，佛罗里达州墨尔本市

# 致　谢

**Mike Andrews**：我要感谢在 Foundstone Professional Services 的同事的支持和在那里培养出的对知识的好奇心，特别要感谢 Kartik Trivedi 在 Web 服务一章对我的帮助，和 Carric Dooley 提供的附录中的工具。还要感谢 Eric Heitaman、Rodolph Araujo 和 Shanit Gupta 详细讨论了哪些内容应当属于 Web 测试方法。我还要感谢 Toby Mikle 为本书绘制的蜘蛛和苍蝇的卡通画。最后还要提到我在佛罗里达科技研究中心时，Scott Chase、Matt Oertle 和 Hugh Thompson 激发了我的写作动机和灵感——祝你们一帆风顺。

**James A. Whittaker**：我要感谢我在 Security Innovation（SI）和佛罗里达科技研究中心的同事与搭档们对软件攻击的原理无数个小时的贡献。特别要感谢 SI 的 Hugh Thompson、Florence Mottay、Scott Chase 和 Jason Taylor，你们都是该领域的行家，所发现的 Web 漏洞比我知道的其他任何一个小组都要多。Web 因为你们而变得更加安全可靠。

# 关于作者

**Mike Andrews** 是 Founstone 的资深顾问，专攻软件安全，并且对 Web 应用程序进行安全评估，对 Web 攻击进行分类。他在大西洋两岸积累了丰富的教育和商业经验，而且还是一位著述颇丰的作者，同时又是一位出色的演讲家。

在加入 Foundstone 之前，Mike 是一名自由顾问，开发基于 Web 的信息系统；使用这套系统的客户有 The Economist（伦敦运输行业的龙头企业），以及英国的很多大学。在经历了多年讲师和研究人员的生活之后，Mike 于 2002 年以助理教授的身份加入了佛罗里达科技研究中心。在那里，他负责很多研究项目，同时还要为海军研究所、空军研究实验室以及 Microsoft 进行安全方面的评定，

Mike 还在英国坎特伯雷的 Kent 大学获得了计算机科学专业的博士学位，主要的研究课题就是调试工具以及程序员心理学。

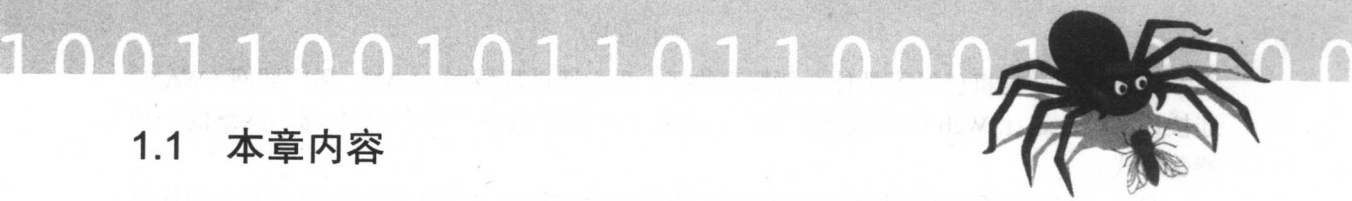
**James A.Whittaker** 是佛罗里达科技研究中心（Florida Tech）计算机科学方面的教授，并且是安全改革的发起人。1992 年，他在美国的田纳西州大学获得了博士学位。他的研究重点是软件测试、软件安全、软件漏洞检查、计算机安全等。

James 是 *how to break software*（Addison-Wesley，2002 年）的作者以及 *how to break software security*（Addison-Wesley，2003 年）的合著者（另一位作者是 Hugh Thompson），并且在软件开发以及计算机安全方面发表了五十多篇论文。他在软件测试领域和防御性安全应用方面取得了很多专利；在佛罗里达科技研究中心做教授时，还为基金会筹集到了几百万美元的赞助。他还为 Microsoft、IBM、Rational 等很多美国公司做过测试和安全方面的顾问。

2001 年，James 被指派参加 Microsoft 的可靠性计算理论会议，并且被“系统和软件”杂志的编辑们称为“顶级学者”，皆因他在软件工程方面的造诣。他在佛罗里达科技研究中心的研究小组也因其测试技术和工具而出名，其中包括广受称赞的工具 Holodeck，可以在程序运行的过程中注入错误。同时，他的团队研究对软件安全进行的攻击，包括破解密码、通过异常攻击突破软件防护，入侵受保护的网络等都取得了显著的成绩。

# 目　录

# 第 1 章　与众不同的 Web





## 1.1　本章内容

本章罗列了一些和使用 Web 软件工作有关的内容。对于一个这方面的新手而言，它提供了一些很有趣的背景知识。无论你在 Web 开发方面是否有经验，也不管你是什么职位（项目经理、测试人员、开发人员或是其他的技术岗位），本章都能够让你很好地理解 Web 项目中的相关内容，并为阅读在后续章节中提到的 Web 攻击做好准备。

Web 是与众不同的。理解它的背景和细节能够让你的工作更有效率。

## 1.2　简介

今天我们所知的计算机软件诞生于二战时期。战争爆发的时候，人们非常需要计算的能力。飞弹的弹道轨迹需要计算得更加精确，也需要算得更快。我们也需要破译加密的电文，这样当我方舰队必须穿越危险的水域时，能够知道敌方舰队的准确位置。这些重大的需求刺激了计算机技术的迅速发展。

接下来的几十年里计算机和软件的首次应用带来了令人难以置信的变化。大学里开设了计算机科学的学位。大型的商业机构和政府机关采用自动化设备来完成原本复杂的手工工作。每年都会有很多新的自动化手段取代传统的手工工作方式，直到我们的世界变成了今天的样子：计算机和软件渗透了社会生活的方方面面。事实上，如果不运行几千行代码，早晨我们甚至很难下床吃早餐。

在不断变革的推动下，计算机技术的快速发展即使在今天仍然没有丝毫减缓。但有一项变革使得其他的一切都黯然失色。这项革新在这个充满了变革的领域中独树一帜，它将改变地球上几乎所有计算机用户的生活。这项革新比其他的任何技术都创造了更多的用户、商机以及成功的范例。

它就是 World Wide Web。

Web 改变了一切。它改变了软件工程，并迫使我们重新评价软件测试的技术。后面的章节会提到这些技术。在本章中，我们先反思一下 Web 带来的这些变化，并为学习本书后

面介绍的测试技术打下基础。

## 1.3　World Wide Web

网络化的计算机已经不是什么新鲜事物了。我们通过局域网（LAN）和广域网（WAN）连接计算机，这比 Web 的连接范围更广。事实上，Web 只是“客户-服务器”网络的一种特殊形式。

客户-服务器网络通过将复杂和耗时的计算交给被称为“服务器”的昂贵的大型计算机完成来有效利用计算资源。服务器往往具有很大的存储容量和内存以及多线程、高速的处理器。这些高速设备使其在完成高强度的计算处理时比一般计算机更快，并可以将结果利用一定的通信手段传递给称为“客户”的普通微型计算机。

在客户-服务器网络中，有三个重要的组成部分：

- 服务器计算机
- 一台或多台客户计算机
- 客户与服务器的连接，也就是网络

其基本构想如下：客户机需要数据或是网络资源（比如打印机），它通过网络连接服务器并请求获取数据或资源。服务器完成计算请求并通过网络将数据或结果返回给客户（参见图 1-1）。

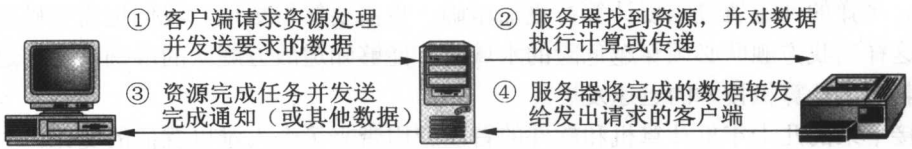



**图 1-1　客户-服务器网络流程图**

显然，这个简单的交互过程中包含了很多内容。在客户端，必须开发相应的软件来连接网络并收发请求及数据，而服务器端也是如此。

在网络层，需要有相关的协议来保证计算机的通信。这种协议必须考虑到网络带宽的问题、数据传输的丢失、冲突、出错，以及某一台或其他计算机（或者资源）不可用的情况。

好在上述问题都已经不同程度地被解决了。像传输控制协议（TCP）、用户数据报协议（UDP），以及它们所支持的协议（比如互联网协议（IP）、地址解析协议（ARP）和域名系统（DNS）等）都已经被实现，这使得客户和服务器端的开发者变得很轻松。现在最大的问题是：这样强大的计算机网络应该用于何处？

考虑到网络最早的用户是科研人员，而他们的主要问题是对数据的访问。大学里那些

和工业界及政府存在合作关系的科学家需要找到并获取开展研究所需要的数据，并和他们的合作伙伴共享这些数据。

当我们这样做的时候，为什么不能公布我们所有的研究，使得任何人都能访问和使用呢？难道不能扩展计算机网络的范围以包括世界上所有的计算机么？

在 21 世纪计算机网络早已遍布全球。但是想像一下所有计算机都通过拨号或者点对点方式连接到网络的世界——一个建立在无连接或“孤立”的计算机基础之上的世界。

这样的缺陷激发了大量技术创新。随着时间的流逝，广为传播的网络提供了比拨号更好的方式，连接到网络的计算机的数量大大增加了。这种增长带动了应用程序的种类、信息量以及网络上可用资源的增长。

World Wide Web 最初源自一种创造性的思想：新的网络层协议、新的服务软件实现网络连接并处理客户各种各样的需求。新的客户软件浏览远程的服务器并搜索整个服务器来寻找所需信息。

于是 World Wide Web 诞生了。这是一个全球性计算机网络，使用统一的语言和协议：超文本传输协议(HTTP)、超文本标记语言(HTML)、可扩展性标记语言(XML)和 JavaScript——那些原有的协议已经使得网络广为传播，World Wide Web 也是在此基础上建立并发展起来的。但今天的 Web 已经远远超越了当初。

Web 起初是为了取代 Internet 的主要功能：电子邮件和文件传输协议（FTP）——它们曾是通信和共享文件的主要途径。当初，多个用户之间共享文件是通过一种叫做 gopher 的系统。和今天的 Web 很类似，gopher 允许用户通过 Veronica（那时候的 Google）搜索文档，而文档也可以链接在一起供浏览。但到了 20 世纪 90 年代末，gopher 几乎已经消失了。gopher 消失的原因是多方面的，但最重要的一点是密歇根大学（该技术的发明者）决定对它的使用收费，这使得人们都流向了免费的 World Wide Web。而且 Web 使用的 HTML 语言也比 gopher 更加强大和富有表现力。

Web 的数据报文件以一种更加简单和直接的方式被共享（任何编码和解码过这种文件的人都会告诉你，它是献给人类的一份礼物！）。因而用户可以在自己的计算机上浏览其他计算机存储的文件。这和两台计算机距离的远近、系统平台的类型以及文件格式都没有关系。

其中的奥秘在于一个叫做 Web 服务器的服务程序，它允许远程客户端访问服务器硬盘上特定的部分。Web 彻底改变了我们共享文件和传递信息的方式。

现在我们需要一个浏览这些信息的办法。坐在一个基于提示符的操作系统前容易让人觉得像是在漫游太空，而且随着用户数量的增长，手动上网的效率也显得太低了。对客户端工具的需求变得越来越强烈，而这种需求也刺激了市场。

于是 Web 浏览器出现了。它是客户计算机的终端工具，用于访问世界各地的服务器上越来越多的网页。

即使有了这些新的工具，Web 的主要用户仍然是科学家、研究人员和那些非常需要数据共享的人。和任何科学创新一样，Web 应用层出不穷、Web 的能力也快速膨胀。这时候，商业界也开始关注它了。

至此我们已经拥有了一个可以通过计算机到达任何地方、联系任何人的工具——World Wide Web。市场调查人员看到了它的潜力，销售人员也赞同这样的看法，不久高层主管也被说服了。

Web 开始大众化。

现在，使用网络的不再仅仅是大学教授，每个人都有了自己的主页。通信社在新闻广播的结尾都带上他们的主页（诸如 http://and.com 之类的地址）。电视广告里也都包含 URL 地址，给用户提供关于公司和服务的详细信息。

但“详细信息”也是整个 Web 可以提供给我们的：更快、更容易地下载静态网页内容。

当然，Web 技术所能提供的还远远不止是静态的 Web 页面！除了在浏览器中显示的静态页面之外，Web 服务器就不能提供其他的内容吗？浏览器作为用户界面从服务器那里获得了什么呢？如果一个住在爱荷华州的人访问一台阿姆斯特丹的服务器，他会得到些什么？

考虑下面的情形。如果我们给旅行社打电话，旅行社会根据我们的需求向计算机输入数据，然后告诉我们所得的结果。为什么我们不能直接向旅行社的计算机中输入数据呢？为什么非要在客户和服务器之间额外增加一个人工操作？

上面的想法在技术上已经没有问题了，因为有了 Web 的交互方式。如今的 Web 页面已经可以包括表单，它能够直接输入数据并传送到远程服务器的数据库中。

World Wide Web 的真正潜力终于被发掘了出来。我们不仅能使用 Web 传递信息，还可以和远程的用户或顾客直接进行交互。

各大公司开始争相在 Web 上发布信息、提供交流平台，以获取更多的利润。Amazon、eBay 等网站成为了这个领域的先锋。他们用购物车和人性化的导购方式构建了一个电子商务平台，使我们的网上购物感觉更好。

从仅仅是学术界通过传统网络交流学术成果开始，走到今天取得了辉煌成就，互联网经历了漫长的发展过程。

## 1.4　Web 世界的价值

Internet 的繁荣给世界带来了重大的改变。“.com”成为了家喻户晓的词汇，甚至被一些电视广告用作噱头。这种大量公司投身于 Web 的现象在华尔街被称为“.com 爆炸”。

各大公司纷纷开设了自己的 Web 网站。HTML 开发人员发现他们比大多数 C++程序员的收入更高，而且 Web 网页正如同雨后春笋一般地出现。这些网页大多数是仅仅包含了一

些信息的静态页面，但有的网站已经打破了这种常规，变得更加具有交互性。

突然之间，我们已经可以通过网络购物了。

网络开始挑战杂志社、商场、零售中心、寻呼台，甚至是音像制品这些传统产业。电话公司及时察觉到了公众的变化，开始大量投资，把现有的电话网改造成为基于网络的通信方式。到了 20 世纪 90 年代末，已经几乎没有人不知道 Web，而今天已经几乎没有人没在 Web 上购买过商品了。

Web 领域的这些崭新的应用在软件和网络世界中开拓出了一片新天地。面向 Web 的应用程序和传统的客户-服务器模式的应用程序是不一样的，其中包含了很多新的计算机技术，急于求成只会导致错误的产生。

## 1.5　Web 和客户机-服务器

World Wide Web 是“客户机-服务器”模式的一种特殊形态。“客户机-服务器”模式指的是由一个或多个集中的服务器向大量的客户机提供数据、资源、程序等。传统意义上，这个模型包含了一个性能强劲的中央服务器，它们连接了很多远程的哑终端，这些哑终端并不进行实际的运算，只是提供了和服务器的接口。可以把哑终端想像成连接到远程服务器的键盘和显示器。

很多 UNIX 网络都是由连接到瘦客户机的服务器组成的。多数应用都在服务器端执行，客户端完成本地数据的存储和小型的计算任务。服务器则负责大多数的复杂运算。

Windows 系统的网络则恰好相反。胖客户机进行基本的办公应用和浏览，而独立的服务器则用来提供网络服务（Web 服务器、DNS 等），以及海量的存储。

Web 则是“客户机-服务器”模式的一种特殊形态，采用胖客户机，在诸如 HTTP、HTML、XML 以及 SOAP 之类协议上运行。此外，Web 还加入了对“不信赖”用户的验证。与此相反，传统的网络只存在于公司的内部，由防火墙保护起来；而 Web 则在任何时候都可以被任何人访问。

在传统的“客户机-服务器”网络中，究竟哪些任务由客户机执行，哪些又由服务器来完成，这是泾谓分明的。此外，客户机和服务器都仅仅存在于某个公司的内部。

但是在 World Wide Web 中却完全不是这样。因为 World Wide Web 中的前两个 W 就代表了全世界。在 Web 中，客户机是不在服务器的管辖范围之内的。与 LAN 不同的是，Web 中不存在需要保护的边界。所有的客户机都被假想为“不能信赖”的；客户机需要提交附加的要求，指明如何在客户机和服务器之间分配计算。

LAN 可以被设计为性能最大化的。客户机可以完成的运算越多，中央服务器的运行就越快。这可能就是胖客户机取代了瘦客户机的一个原因吧。运算负担被分担得越平均，全网的速度就越快。