

<<开发更安全的ASP.NET 2.>>

图书基本信息

书名：<<开发更安全的ASP.NET 2.0实用程序>>

13位ISBN编号：9787115177483

10位ISBN编号：7115177481

出版时间：2008-7

出版时间：人民邮电出版社

作者：拜尔

页数：447

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<开发更安全的ASP.NET 2.>>

### 内容概要

《开发更安全的ASP.NET 2.0实用程序》以作者的实际经验为主，介绍了关于开发更安全的ASP.NET 2.0应用的各方面的内容。

全书共10章，内容包括：Web应用程序安全、ASP.NET 2.0构架、输入验证、存储机密、身份验证和授权、安全提供程序和控件、日志和监测、部分信任ASP.NET、部署和配置以及工具和资源。附录部分提供了创建自定义受保护配置提供程序、会话状态、分拆ASP.NET应用程序、安全的Web服务和使用Visual Studio Team Edition进行安全测试等内容。

《开发更安全的ASP.NET 2.0实用程序》提供的示例简练易懂，书中代码示例都经过认真的编写，读者无需记住所有的内容，而可以将《开发更安全的ASP.NET 2.0实用程序》的实例很容易地引入到现实的应用程序中。

《开发更安全的ASP.NET 2.0实用程序》适用于使用ASP.NET 2.0技术同时关注安全性的各方面读者。

## <<开发更安全的ASP.NET 2.>>

### 作者简介

Dominick Baier，为世界各地的公司咨询软件安全方面的问题，同时还负责DevelopMentor的安全课程和一家针对开发人员的培训公司。他是一位具有BS 7799/ISO 17799资格的主导审核员，还是Visual Developer-Security方面的MVP。另外，Dominick经常在业界的会议中发言，并为德国MSDN的安全主题内容撰稿，同时还撰写了一个受欢迎的博客。

## 书籍目录

第1章 Web应用程序安全1.1 OWASP Top 101.2 总体原则1.2.1 安全是一种特性1.2.2 使用最低权限1.2.3 预防、监测和反应1.2.4 分层防御1.2.5 不存在可信的输入1.2.6 注意故障模式1.2.7 注意应用程序拒绝服务1.2.8 首选默认安全措施1.2.9 加密不能确保安全1.2.10 防火墙不能确保安全1.3 小结第2章 ASP.NET 2.0架构2.1 理解宿主2.2 理解管线2.2.1 HTTP模块2.2.2 编写模块2.2.3 处理程序2.2.4 检查管线2.3 编译ASP.NET页2.4 小结第3章 输入验证3.1 什么是输入3.2 输入验证的必要性3.3 输入验证技术3.3.1 黑名单3.3.2 白名单3.4 缓解技术3.4.1 输出编码3.4.2 沙盒3.4.3 完整性检查3.5 ASP.NET应用程序中的验证3.5.1 自动验证服务3.5.2 表单验证3.5.3 创建自定义验证控件3.6 小结第4章 存储机密4.1 识别攻击和攻击者4.2 加密术是救星吗4.3 哈希数据4.3.1 哈希算法4.3.2 .NET的哈希算法4.4 保存密码4.5 加密数据4.5.1 对称性加密4.5.2 加密算法4.5.3 密钥和密钥大小4.5.4 .NET的对称性加密4.5.5 完整性保护4.5.6 整合：设计使用对称性加密的应用程序4.5.7 非对称性加密4.5.8 证书4.5.9 在.NET中使用非对称性加密证书4.5.10 整合：设计使用非对称性加密和证书的应用程序4.6 使用Windows数据保护API4.7 保护配置数据4.7.1 配置和安装4.7.2 保护配置4.8 保护ViewState4.9 小结第5章 验证和授权5.1 基础知识5.1.1 术语5.1.2 应用程序设计5.1.3 ASP.NET安全管道5.1.4 .NET安全架构和基于角色的安全5.1.5 服务器验证5.2 使用Windows账户5.2.1 IIS验证方法5.2.2 授权5.2.3 模拟5.2.4 委托5.2.5 安全上下文和访问外部资源5.3 使用自定义账户5.3.1 表单验证5.3.2 表单验证机制5.3.3 配置表单验证5.3.4 确保表单验证的安全5.3.5 自定义表单验证5.3.6 Web场5.3.7 单点登录5.3.8 使用ASP.NET保护非ASP.NET资源5.4 混合方法5.4.1 手动Windows验证5.4.2 协议转换5.4.3 对自定义账户实现基本验证5.4.4 用户证书5.4.5 混合模式验证5.5 小结第6章 安全提供程序和控件6.1 理解成员功能6.1.1 方法6.1.2 事件6.1.3 成员配置6.1.4 SQL成员提供程序6.1.5 Active Directory成员提供程序6.1.6 与成员相关的控件6.2 理解角色管理器6.2.1 角色管理器模块6.2.2 角色管理器配置6.2.3 SQL角色提供程序6.2.4 Windows令牌角色提供程序6.2.5 授权存储角色提供程序6.2.6 与角色相关的控件6.2.7 成员和角色打包6.3 使用SiteMap导航6.4 创建功能和提供程序6.5 指南6.6 小结第7章 日志和监测7.1 错误处理7.1.1 获取401非授权错误7.1.2 错误处理7.2 日志和监测7.2.1 事件日志7.2.2 性能监视器7.2.3 电子邮件7.2.4 Windows管理监测7.2.5 ASP.NET跟踪和System.Diagnostics.Trace7.2.6 日志和部分信任7.3 健康监测框架7.3.1 创建事件7.3.2 配置健康检测7.3.3 SQL服务器提供程序7.3.4 WMI提供程序7.3.5 电子邮件提供程序7.3.6 编写自定义提供程序7.3.7 编写自定义缓冲提供程序7.3.8 状态监视和部分信任7.3.9 指南7.4 小结第8章 部分信任ASP.NET8.1 为什么选择部分信任8.2 配置部分信任8.3 理解策略文件8.3.1 安全类8.3.2 命名权限集8.3.3 代码组8.3.4 策略加载和解析8.4 自定义策略文件8.5 分割代码8.5.1 重构代码8.5.2 堆栈审核8.5.3 为经过分区的程序集修改策略8.5.4 限制调用组件的用户8.6 创建自定义权限8.6.1 权限类8.6.2 封装8.6.3 属性8.7 SecurityException的作用8.8 锁定配置8.9 小结第9章 部署和配置9.1 总指导原则9.2 操作系统强化9.2.1 自动更新9.2.2 禁用服务和协议9.2.3 包过滤9.2.4 保护Windows文件共享9.2.5 审核9.3 数据库服务器强化9.4 Web服务器强化9.4.1 应用程序池9.4.2 Web服务扩展9.4.3 Web内容9.4.4 HTTP头9.4.5 日志9.4.6 URLScan9.4.7 访问控制列表9.4.8 启用SSL9.4.9 验证方法9.5 ASP.NET强化9.5.1 配置锁死9.5.2 推荐设置9.5.3 预编译9.6 小结第10章 工具和资源10.1 工具类型10.2 确定合适的工具10.3 浏览代理服务器和HTTP协议检测工具10.3.1 Fiddler10.3.2 Paros10.3.3 WebScarab10.3.4 WSDigger10.4 黑盒扫描器10.4.1 SPI Dynamics WebInspect10.4.2 Watchfire AppScan10.4.3 Berretta10.5 配置分析10.5.1 SSL Digger10.5.2 PermCalc10.5.3 Desaware CAS Tester10.5.4 ANSA10.5.5 IIS Lockdown10.6 源代码分析器10.6.1 Foundstone CodeScout10.6.2 Microsoft PREfix和PREfast10.6.3 Compuware ASP.NET Security Checker10.6.4 SPI Dynamics DevInspect10.7 多功能工具10.8 二进制分析10.8.1 静态二进制分析工具10.8.2 动态(“运行时”)二进制分析10.8.3 调试器10.8.4 反编译器/模糊处理器10.9 数据库扫描器10.9.1 AppDetective10.9.2 MetaCoreTex10.9.3 NGSSquirrel10.10 博客10.11 小结附录A 创建自定义受保护配置提供程序附录B 会话状态B.1 会话状态如何工作B.1.1 Cookie vs.查询字符串B.1.2 超时设定B.1.3 会话模式B.2 会话存储B.2.1 进

## <<开发更安全的ASP.NET 2.>>

程内提供程序B.2.2 状态服务器B.2.3 SQL ServerB.3 小结附录C 分拆ASP.NET应用程序C.1 创建服务器端C.2 创建客户端C.3 创建部分信任客户端C.4 小结附录D 安全的Web服务D.1 适用情况D.2 安全的通信和服务验证D.3 客户端验证D.4 小结附录E 使用Visual Studio Team Edition进行单元测试E.1 测试驱动开发E.2 运行测试E.3 测试现有代码E.4 测试列表和测试运行配置E.5 建立正确的测试环境E.6 测试私有方法E.7 预期的错误E.8 数据驱动测试E.9 数据驱动测试的数据管理E.10 测试Web服务代码E.11 在ASP.NET内部运行测试E.12 小结

## &lt;&lt;开发更安全的ASP.NET 2.&gt;&gt;

## 章节摘录

第1章 Web应用程序安全 20世纪90年代后期是针对网络和操作系统攻击快速增长的时期。近来，每个公司都碰到了缓存溢出的问题。

Windows NT、Windows 2000和Internet信息服务（Internet Information Services，IIS）几乎每天都会公布新的漏洞。

从那个时候开始，很多软件企业从中汲取了教训，并且认识到必须将安全作为产品的常规重要特性。

只有改善开发过程，创建更健壮和安全的代码，才能取得这种成效。

Microsoft就是一个最好的范例。

为了减少软件中安全漏洞的数量，Microsoft实施了安全特性设计和测试集成部件的合作开发过程。

同样重要的一点是，这些软件部件的设计过程嵌入到每一个周期的开始阶段。

读者可以在<http://msdn.microsoft.com/msdnmag/issues/05/11/SDL/default.aspx>

查阅Microsoft安全开发生命周期（Security Development Lifecycle，SDL）或者阅读由Michael Howard和Steve Lipner最新撰写的《The Security Development Lifecycle》（Microsoft出版社，2006），以便了解它的工作方式。

这些软件行业（有益的）的变化意味着攻击操作系统级别的系统变得越来越困难。

攻击开始寻找更有吸引力的目标。

他们向上转移了几个层次，从而上升到ISO模型，而多种原因让Web应用程序注定会成为新的目标。

首先，攻击Web应用程序非常容易。

HTTP是非常简单的底层协议。

它基于文本并且是无状态的，这也就意味着并不需要专业工具对二进制数据进行编码，简单的telnet客户端就已经完全能够解析HTTP包。

无状态协议意味着对Web应用程序每一轮访问都包含所有必要的的数据，通常也不需要事先设置会话以防范攻击，这使得与其交互很容易。

另外，还有一个无法改变的事实是，即使只显示一个登录页，基于HTTP的应用程序也允许匿名访问。

只要这个页面是实际应用程序的一部分，那么就能够用其攻击应用程序代码。

应用程序的其他特殊字符也是最常见的途径，换言之，它们是Internet和内部资源（例如数据库）之间的最后的堡垒。

另一方面，Web应用程序开发环境的进步使创建复杂的数据驱动Web应用程序变得非常容易。

将基于桌面Windows Forms的编程范例转移到Web开发中，吸引了很多企业和开发人员。

通常，在基于Intranet的传统应用程序中，并没有给予安全问题太多关注，但是向完全不同环境（例如Internet）的迁移彻底改变了这种情况。

也就是说，成千上万的开发人员，无论具有多少开发安全技术的经验，都可以开始编写能够预防所有Internet犯罪的应用程序。

## <<开发更安全的ASP.NET 2.>>

### 编辑推荐

构建更安全Web应用程序的核心指南 《开发更安全的ASP.NET 2.0实用程序》讲解有关构建更为安全的ASP . NET 2 . 0应用程序的专业技术。

在《开发更安全的ASP.NET 2.0实用程序》中，顶尖的安全专家将介绍极有价值的经验、实际建议和大量的使用Microsoft Visual C#编写的代码示例。

这些内容将帮助读者开发更健壮、更可靠和更可防御攻击的Web应用程序。

核心内容： 加强Web服务器、操作系统、通信协议和ASP . NET的安全；使用正则表达式、沙盒和其他技术验证输入数据；理解各种加密方法的设计方法和安全内涵；集成Microsoft Windows安全特性的方法，例如模拟、委托和协议转换；实现Web场、单点登录和混合模式验证； 使用基于提供程序的功能实现用户和角色的管理与验证； 使用错误处理、日志和规范跟踪攻击行为；使用部分信任锁定应用程序。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>