MVP
Microsoft®
Most Valuable
Professional

无懈可击

——全方位构建安全Web系统

杨云 编著

操作性强

通过图文并茂的方式把复杂的问题简单化，让读者看得懂、学得会、用得上

内容全面

本书涵盖Web系统从设计到开发、防御等领域所需要的全部技术

案例经典

实例涵盖存储、通讯、钓鱼技术等与安全最相关的领域

通用性强

本书采用目前最通用的.NET Web程序，方便读者参考借鉴

清华大学出版社

——全方位构建安全web系统

杨 云 编著

清华大学出版社
北京

内 容 简 介

本书对常见的 Web 安全问题，依照分类的方式讲解每一个知识点，告诉读者如何开发安全高效的 Web 系统，如何使自己的系统免受黑客的攻击。本书内容是作者多年项目实施和管理经验的总结，在此基础上加以提炼，试图用最简明易懂的方式介绍网站开发的安全问题以及应对措施；内容涉及界面 UI 安全、代码安全、中间件安全、Session 安全等，并用典型实例作为引导，介绍各种安全类库和安全编程。

本书适合网站开发人员、应用程序设计和开发人员使用，也适合网站系统的管理维护人员阅读和参考。

图书在版编目（CIP）数据

无懈可击——全方位构建安全 Web 系统 / 杨云编著. —北京：清华大学出版社，2012.1
ISBN 978-7-302-26928-1

Ⅰ. ①无…　Ⅱ. ①杨…　Ⅲ. ①网站－安全技术　Ⅳ. ①TP393.08

中国版本图书馆 CIP 数据核字（2011）第 194601 号

责任编辑：袁金敏　赵晓宁
责任校对：徐俊伟
责任印制：杨　艳
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编**：100084
社　　总　　机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62795954，jsjjc@tup.tsinghua.edu.cn
质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185×260　**印　　张**：19.25　**字　　数**：481 千字
版　　次：2012 年 1 月第 1 版　　**印　　次**：2012 年 1 月第1 次印刷
印　　数：1～4000
定　　价：45.00 元

产品编号：043911-01

前　言

在互联网中遨游，最让人担心的 3 个头等问题，您知道都是哪些吗？好吧，我来告诉你，它们是：安全！安全！还是安全！

互联网每天都在上演着失去用户信任和安全感，从而倒闭的悲剧。作为开发人员的你，是否为总结安全点而苦恼，为不知道如何加固功能点而纠结？那么如何才能设计出高安全度的网络系统呢？这个问题已经关系到用户对网站系统的信任程度，关系到企业和系统的生死存亡了。在笔者看来，最关键的就是设计思想和每个功能点的完善性。

时下在网站系统中使用最广泛的就是 ASP.NET 和 Java 架构了。其中.NET 框架是微软公司为了满足广大用户的需求而开发的一种通用平台。Java 技术由于没有公司做统一的文档编写，导致设计安全模块完全需要工程师自己的经验。这些技术在帮助我们方便、快捷地完成网站开发的同时，在实际运行环境中的黑客入侵和安全隐患也成了不容忽视的问题，这就要求开发人员从设计和开发两方面都要关注系统的安全性。

在很多人的印象中，安全工作似乎都是在问题发生后才采取措施。本书旨在从根本上纠正了这一做法，通过对.NET\Java 平台安全问题的了解，做到风险早避免，问题早处理，在应用程序开发的全生命周期中严把安全关，保证系统的正常、稳定运行。

1．本书介绍

本书可以分为 4 部分。第一部分为第 1～第 3 章，介绍了应用安全的基本概念和安全控件；第二部分为第 4～第 6 章，讨论了如何进行数据加密和验证；第三部分为第 7～第 12 章，阐述了如何使用与程序运行平台有关的安全功能，包括组件安全、会话安全、代码信任和网站钓鱼技术的防与治；第四部分为第 13～15 章，重点介绍了服务器上如何安全的部署你的程序，以及代码的安全测试和审核。

2．本书特点

本书最大的特点就是教读者对症下药，根据不同的功能点或功能模块传授不同的代码安全防范和设计策略。书中采用时下最新的 ASP.NET 4.0 技术为范本，讲解如何将安全代码和防范技术植入系统。通过图文并茂的方式把复杂的问题简单化，让大家知道应该如何去做。

另外，书中选用的代码和实例都是被验证过的，能够切实提升安全的有效方法。使得读者在短期掌握别人长期才能总结出来的技术和经验，尤其对专业人员和学生的帮助很大。

本书在设计方式和方法上不受技术的限制，如果抛开技术本身，单说设计方法该书所讲解的方法将使读者能够“依葫芦画瓢”。

本书所涉及的网站安全问题仅限于学习研究，若读者据此攻击他人网站，责任自负。

3．本书适用人员

本书适合每一位网站开发人员阅读，作为应用程序的设计和开发人员，他们应该了解所用语言安全性特点和局限性，以便在设计和编码过程中进行安全性相关考虑。在本书的每一章节，都运用大量实例，帮助开发人员学习安全服务配置和代码编写。

另外，作为应用系统的管理维护人员和用户也应该阅读本书。通过本书，读者可以清晰地分辨哪些行为是危险的，而这些行为是我们每天都会遇到的。以最常见的数据加密和身份验证为例，了解.NET 平台安全运行的机制，就可以减少在系统操作过程中的低级错误，避免安全隐患。

4．致谢

本书主要由杨云编写，参加编写的还有李广平、王春中、冉剑、陈代勇、陈佳佳、陈家云、王磊、王宇晟、胡浩、蔡芳、刘扬、陈雪郊、谢晓锋、王毅、刘小鹿、胡建、杜华富、成梅花。

本书在编写过程中，得到清华大学出版社编辑袁金敏老师的指导和帮助，在此表示感谢。另外，感谢读者和我家人一直以来对我的支持和帮助。

由于水平有限，书中难免存在不足之处，欢迎广大读者批评指正。

编　者

2011 年 5 月

目　录

第一部分

第二部分

第三部分

第四部分

第一部分

第 1 章　网站安全技术概述

目前构建网站系统的语言无外乎 ASP.NET、JSP 或 PHP 等，它们自身的安全性和编程的方式是否安全是关键。随着近年来.NET 技术和 JSP 越来越为大家所熟知，各类基于它们的 Web 应用系统被广泛开发和应用。特别是基于.NET 框架的 ASP.NET 等技术使得 Web 程序更加容易开发和维护，但 Web 应用程序所带来的安全漏洞也越来越多，每年由于代码安全问题，都会造成巨额损失。目前大部分互联网犯罪和黑客行为，都是通过网站的漏洞展开的。例如，“世界著名社区网站 Facebook 近日测试新网站的用户界面，其中的一个漏洞竟然导致网站 8000 万用户的生日信息被泄露。类似的事情数不胜数，盛大、联想等公司也都出现过网站被攻破的事件。

.NET 框架本身具备很多编写安全代码的技术，运用这些技术，可以开发安全的 Web 应用程序。本章将从程序安全性的含义、用户角色与代码访问、安全模型、安全类库 4 个方面讲述 Web 应用程序安全的基本概念。

1.1　代码安全性的含义

Web 应用程序的安全性主要包括代码访问的安全性和基于角色的安全性。

一般来说，这两种安全性可以互相渗透，熟悉代码访问安全性的开发人员可以轻松使用基于角色的安全性，而熟悉基于角色的安全性的开发人员也可以轻松地使用代码访问的安全性。代码访问的安全性和基于角色的安全性都是用公共语言运行库提供的一个通用结构来实现的。

代码访问的安全性和基于角色的安全性使用相同的模型和结构，因此它们共享若干基础概念，包括安全权限、类型安全、安全策略、主体、身份验证、授权。

1．安全权限

公共语言运行库允许代码仅执行它有权执行的操作。运行库通过权限对象实现托管代码的强制限制。安全权限的主要用途如下：

（1）代码可以请求需要的权限，.NET 框架安全系统确定是否允许这样的请求。一般来说，仅当代码的验证区域值得授予这些权限时，系统才会授予权限。代码接收到的权限不会比安全性设置所允许的权限要多。

（2）运行库根据某些条件来授予代码权限，这些条件包括代码标识的特性、请求的权限和代码的信任程度（由管理员设置的安全策略确定）。

（3）代码可要求其调用方具有特定权限。如果在代码上设置了对特定权限的请求，则使用此代码的所有代码也都必须拥有该权限才能运行。

调用方一般可能拥有三类权限，各类权限都有特定的用途：

① 代码访问权限：表示对受保护资源的访问权或执行受保护操作的能力。

② 标识权限：表示代码具有支持特定类型的标识的凭据。

③ 基于角色的安全权限：确定用户（或用户的代理）具有特定的身份。PrincipalPermission 是唯一一个基于角色的安全权限。

运行库在很多命名空间中都具有内置权限类，此外，运行库还支持对设计和实现自定义权限类。

2．类型安全

类型安全又称为强类型，指不可以将原始类型强制的转换成另外一个目标类型，从而对这个转换后的原始类型进行目标类型上定义的操作。通俗点讲，类型安全指的是变量类型定义后，不能再转换到其他类型（非本类型或非本类型的子类型）。

类型安全和内存安全的关系非常紧密，类型安全的代码只允许访问授权可以访问的内存位置，例如不能从其他对象的私有字段读取值。在实时（Just In Time，JIT）编译期间，可选的验证过程检查要实时编译为本机代码的元数据和中间语言（Microsoft Intermediate Language，MSIL），以验证它们是否为类型安全。如果代码具有忽略验证的权限，则将跳过此检查过程。

对于托管代码来说，类型安全验证不是强制的，但类型安全在程序集隔离和安全性强制中起着至关重要的作用。如果代码是类型安全的，则公共语言运行库可以将程序集彼此间完全隔离。这种隔离有助于确保程序集之间不会产生负面影响，提高了应用程序的可靠性。即使类型安全组件的信任级别不同，它们也可以在同一过程中安全地执行。

如果代码为不安全的，则会出现副作用。例如运行库无法阻止非托管代码调用到本机（非托管）代码和执行恶意操作。所有非类型安全的代码必须通过传递的枚举成员（SkipVerification）授予（SecurityPermission）后才能运行。

3．安全策略

安全策略是一组可配置的规则，公共语言运行库在决定允许代码执行操作时遵循该策略。安全策略由管理员进行设置，并由运行库强制执行。运行库帮助我们确保代码只能访问或调用安全策略允许的资源或代码。

每当加载程序集时，运行库就使用安全策略确定授予程序集的权限。在检查了描述程序集标识的信息（称为证据）后，运行库使用安全策略决定代码的信任程度和由此授予程序集的权限。

4．主体

主体代表一个用户的标识和角色以及其拥有的用户操作。.NET 框架中基于角色的安全性支持三种主体：一般主体、Windows 主体和自定义主体。这里需要注意，一般主体独立于 Windows 用户和角色存在。

Windows 主体表示 Windows 用户及其角色。Windows 主体可模拟其他用户，这意味着此类主体在表示属于某一用户标识的同时，可代表此用户对资源进行访问。

自定义主体可由应用程序以任何方式进行定义，这种类型可以对主体的标识和角色进

行扩展。

5. 身份验证

身份验证检查用户的凭据，并根据用户的权限进行验证，找到主体标识。身份验证期间获得的信息可以直接由代码使用。也就是说，一旦找到了主体标识，就可以使用.NET 框架中基于角色的安全确定是否允许主体访问代码。

目前使用的身份验证机制种类繁多，其中大多都同.NET 框架中基于角色的安全一起使用。最常用的机制有 Passport 机制、操作系统机制和应用程序自定义机制。

6. 授权

授权是用来确定是否允许主体执行请求操作的过程。在身份验证之后，使用主体标识和角色信息来确定可以访问的资源，授权使用.NET 框架中基于角色的安全性来实现。

介绍了代码访问安全性和基于角色的安全性共有的基本概念后，下面对这两种安全性的机制进行具体介绍。

1.1.1 代码与代码的安全域

1. 概述

当今高度连接的计算机系统会遇到出自各种来源（可能包括未知来源）的代码。代码可能由电子邮件附带、包含在文档中或通过 Internet 下载。许多计算机用户都亲身体验过恶意代码（包括病毒和蠕虫）造成的严重后果，这些代码可能会损坏或毁坏数据，并浪费时间和资金。

多数普通安全机制根据用户的登录凭据（通常为密码）赋予用户权限，并限制允许用户访问的资源（通常为目录和文件）。但是，这种方法无法解决以下几个问题：用户从许多来源获取代码，这些来源中可能并不可靠；代码可能包含 bug 或具有脆弱性，有可能被恶意代码利用；代码有时候会执行一些操作，而用户并不知道。结果，当谨慎且可信的用户运行恶意软件或包含错误的软件时，计算机系统就可能会损坏，私有数据发生泄漏。

多数操作系统安全机制要求每一段代码都必须完全受信任（Web 页的脚本可能除外），然后才可运行。因此，需要一种可广泛应用的安全机制，即使两个计算机系统之间没有信任关系，该机制也允许在一个计算机系统上生成的代码能够在另一个计算机系统上安全执行。

为了帮助保护计算机系统免受恶意移动代码的危害，让来源不明的代码安全运行，防止受信任的代码有意或无意地危害安全，.NET 框架提供了一种称为“代码访问安全性”的安全机制。代码访问安全性使代码可以根据它所来自的位置以及代码标识，获得不同等级的受信度。

代码访问安全性还实施不同级别的代码信任，最大限度地减少了必须完全信任才能运行的代码数量。使用代码访问安全性，可以减小恶意代码或包含错误的代码滥用代码的可能性。我们可以指定允许代码执行的一组操作，同时还可指定永远不允许代码执行的一组操作。代码访问安全性有助于最大限度地减少由于代码的脆弱性而造成的损害。

以公共语言运行库为目标的托管代码都会受益于代码访问安全性，即使托管代码不进行一次代码访问安全性调用时也是如此。但是，正如代码访问安全性基础知识中所描述的那样，所有应用程序都应该进行代码访问请求。

代码访问安全性是帮助限制代码对受保护的资源和操作的访问权限的机制。在.NET 框架中，代码访问安全性执行下列功能：

（1）定义权限和权限集，它们表示访问各种系统资源的权限。

（2）管理员能够通过将权限集与代码组关联来配置安全策略。

（3）代码能够请求运行所需权限，指定代码不能拥有的权限。

（4）根据代码请求的权限和安全策略允许的操作，向加载的程序集授予权限。

（5）使代码能够要求其调用方拥有特定的权限。

（6）使代码能够要求其调用方拥有数字签名，只允许特定组织或特定站点的调用方调用受保护的代码。

（7）比较调用堆栈上每个调用方所授予的权限与调用方必须拥有的权限，加强运行时对代码的限制。

为了确定是否已授予代码访问资源或执行操作的权限，运行库的安全系统遍历调用堆栈，比较每个调用方所授予的权限与目前要求的权限。如果调用堆栈中的任何调用方没有要求权限，则会引发安全性异常，并拒绝访问。堆栈旨在防止引诱攻击，这种攻击中，受信程度较低的代码调用高度信任的代码，并使用高度信任的代码执行未经授权的操作。运行时要求所有调用方都拥有权限会影响性能，但对于帮助保护代码免遭受信程度较低的代码的引诱攻击至关重要。若要优化性能，可以使代码执行较少的堆栈步；但是，这样做必须确保不会暴露安全缺陷。

图 1-1 所示为“程序集 A4”中的方法要求其调用方拥有权限 P 时引起的堆栈步。

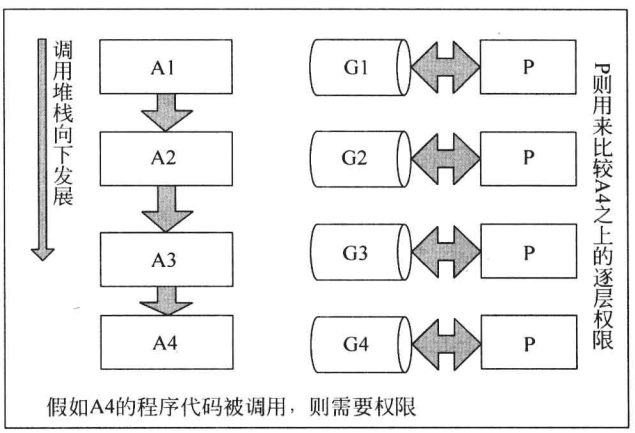

图 1-1　安全堆栈步

代码访问安全性的一种常见应用场合是，应用程序将控件从本地 Intranet 宿主网站直接下载到客户端，用户在控件输入数据，该控件是使用安装的类库生成的。下面是在此方案中使用代码访问安全性的一些方法。

（1）加载前，如果代码拥有特定的数字签名，则管理员可配置安全策略，指定给予该代码特殊权限（比本地 Internet 代码通常收到的权限大）。默认情况下，预定义的 LocalIntranet

命名权限集与从本地 Intranet 下载的所有代码关联。

（2）加载时，除非控件拥有受信任的签名，否则，运行库仅授予控件与 LocalIntranet 命名权限集关联的权限。在控件拥有受信任签名的情况下，会被授予与 LocalIntranet 权限集关联的权限，同时因为控件拥有受信任签名，还可能被授予其他一些权限。

（3）运行时，每当调用方（在此情况下为寄宿的控件）访问公开受保护资源的库或调用非托管代码的库时，该库就会提出安全要求，导致对调用方的权限进行检查，看调用方是否被授予了适当权限。这些安全检查可防止控件在客户端执行未经授权的操作。

2. 基础知识

每种以公共语言运行库为目标的应用程序必须与运行库的安全系统进行交互。当应用程序执行时，运行库将自动进行计算，然后给出一个权限集。根据应用程序获得的权限不同，应用程序或正常运行，或发生安全性异常。特定计算机上的本地安全设置最终决定代码所收到的权限。这些设置可能因计算机而异，所以无法确保代码将收到运行所需的足够的权限。这与非托管开发领域不同，在非托管开发领域，不必担心运行代码所需权限。

每个开发人员都必须熟悉下面的代码访问安全性操作：

（1）编写类型安全代码：若要使代码受益于代码访问安全性，必须使用生成可验证为类型安全代码的编译器。

（2）强制性语法和声明式语法：与运行库安全系统的交互使用强制性安全调用和声明式安全调用执行。声明式调用使用属性执行，强制性调用在代码中使用类的新实例执行。有些调用只能强制性地执行，而有些调用只能以声明方式执行。有些调用可以这两种方式中的任一种方式执行。

（3）为代码请求权限：请求将应用到程序集范围，代码通知运行库在此范围内运行它所需的权限，运行库在代码加载到内存中时计算安全请求。代码使用请求通知运行库运行所需权限。

（4）使用安全类库：类库使用代码访问安全性指定所需权限。

3. 通过部分受信任的代码使用类库

系统一般不允许通过低于完全信任级别（该信任级别是运行库代码访问安全系统授予的）的应用程序调用共享托管库，除非库编写器通过使用 AllowPartiallyTrustedCallers Attribute 类明确允许调用。因此，应用程序编写器必须注意在部分受信任的上下文中不能使用的库。默认情况下，在本地 Intranet 或 Internet 区域中执行的所有代码都是部分受信任的。如果代码不会在部分受信任的上下文中执行或被部分受信任的代码调用，那么就不需要关心本节中的信息。但是，如果编写的代码必须与部分受信任的代码交互或在部分受信任的上下文中运行，则应该考虑以下因素：

（1）必须用强名称对库进行签名，这样该库就可以被多个应用程序共享。强名称允许代码放置在全局程序集缓存中，并允许使用者验证特定的移动代码。

（2）默认情况下，具有强名称的共享库自动为完全信任执行隐式连接请求（Link Demand），无须库编写器执行任何操作。

（3）如果调用方不是完全信任却仍尝试调用库，则运行库将引发安全处理程序 Security Exception，不允许该调用方链接到该库。

（4）为了禁用自动连接功能并防止引发异常，可以将程序的局部信任属性放置在共享库的程序集范围内，此属性允许通过部分受信任的托管代码调用库。

（5）通过部分信任属性被授予对库访问权限的部分信任的代码仍需要遵循本地计算机策略定义的进一步限制。

（6）部分受信任的代码无法通过编程方式调用不具备 AllowPartiallyTrustedCallers Attribute 属性的库。如果某个应用程序在默认情况下不接受完全信任，管理员必须选择修改安全策略并授予该应用程序完全信任，这样的话，应用程序才能调用该库。

特定应用程序专用库不需要强名称或 AllowPartiallyTrustedCallersAttribute 属性，这些库也不能被应用程序之外的潜在恶意代码引用。这样的代码不会受到部分受信任的移动代码有意或无意的滥用，无需开发人员或管理员进行额外操作。

对于以下类型的代码，应该考虑显式启用以供部分受信任的代码使用：

（1）已对安全脆弱性进行反复测试并且符合安全代码指南中所述准则的代码。

（2）专门为部分受信任的方案编写的具有强名称的代码库。

（3）签有强名称的任何组件（不管是部分受信任的还是完全受信任的），这些组件被从 Internet 或本地 Intranet 下载的移动代码调用。在默认安全策略下移动代码接受部分信任，所以这些组件将受到影响。

（4）安全策略授予低于完全信任的所有代码（如果修改了默认策略）。

4．编写安全类库

类库中的编程失误会导致安全漏洞的公开，这是因为类库经常访问受保护的资源和非托管代码。如果设计类库，则需要了解代码访问安全性，并仔细保护类库。

表 1-1 所示为保护类库时需要考虑的 3 种主要元素。

表 1-1　类库保护元素

安全元素	说　明
安全要求	要求是一种应用于类级别和方法级别的机制，它要求代码调用方拥有适当权限。当调用代码时，将在堆栈上检查直接或间接调用代码的所有调用方。要求通常在类库中用来帮助保护资源
安全重写	重写应用在类和方法范围上，为否决运行库作出的某些安全决策的方法。重写在调用方使用代码时进行调用。使用重写可停止堆栈步，并限制已某些调用方的访问权限
安全优化	组合使用要求和重写，可以增强代码与安全系统进行交互时的性能

5．编写安全托管控件

托管控件是下载到用户计算机的网页所引用的程序集，这些程序集根据需要执行。从代码访问安全性角度来看，有两种类型的托管控件：默认安全策略下运行的控件和需要更高信任程度的控件。

若要编写在默认安全策略下运行的托管控件，只需要知道 Intranet 或 Internet 区域默认安全策略所允许的操作即可。只要托管控件在执行时需要的权限不超过它从原始区域收到的权限，该控件就可以运行（记住，管理员或用户可以决定不授予代码来自 Internet 或 Intranet 区域的全部权限）。若要执行，则需要更高程度信任的托管控件，管理员或用户必须调整将运行该代码的任何计算机的安全策略。

编写托管控件时应尽可能使这些控件需要默认情况下授予代码来自 Internet 或 Intranet 区域的权限。对于 Internet 区域，这意味着限制代码只显示 SafeTopLevelWindows 和 SafeSubWindows（由安全系统加以完善，从而防止它们模拟系统对话框），只能与其原始站点进行通信，并且使用有限的、独立的存储。

担负 Intranet 上的代码的权限稍大一些。详细信息参见默认安全策略相关内容。如果控件需要访问文件、使用数据库以及收集有关客户端计算机的信息等，则该控件需要更高程度的信任。

1）开发

高度信任控件在运行时采用的安全策略比它们的源（Intranet 或 Internet）通常认可的安全策略的限制性要低。安全库（如.NET Framework 类）发出的多数权限要求执行堆栈审核来检查所有调用方，以确保已授予它们要求的权限；同时，出于安全目的，尽管网页不是托管代码，但仍被作为调用方处理。执行堆栈审核技术是为了帮助防止受信程度较低的代码引诱高度信任代码执行恶意操作。

浏览器中承载的托管控件可以由网页上的动态脚本操作，所以可以将网页视为调用方，并在安全堆栈审核技术中对其进行检查，防止恶意网页利用信任程度更高的代码。将网页作为调用方处理的结果是，基于其强名称或发行者证书而被授予高级别信任并且从网页运行的控件，将被禁止执行与网页本身出自同一区域（Intranet 或 Internet）的代码所执行的操作。有关部署注意事项的更多信息，请参见 1.1.2 节。从表面上看，几乎不可能编写高度信任控件；但是，代码访问安全性通过有选择地重写安全堆栈审核技术提供实现此方案。

高度信任控件必须明智地使用 Asserts 来简化对它们的调用方（它们从中运行的网页）通常没有的权限执行的堆栈审核技术。使用 Asserts 时必须小心，不要公开可能会使恶意网页执行不当操作的危险 API。在编写高度信任控件时需要的谨慎程度和安全意识与编写安全类库时需要的谨慎程度和安全意识一样多。

下面是有关编写安全托管控件的一些提示：

- 封装需要高度信任的操作，使权限不会被控件公开。这样，就可以断言这些操作需要的权限，可以预见的是，使用该控件的网页无法滥用相应功能。
- 如果控件的设计需要公开执行的高度信任操作，请考虑发出一个站点或 URL 标识权限要求，以确保控件从其运行的网页调用控件。

2）部署

高度信任控件应当具有强名称或签有发行者证书（X.509），使得策略管理员可以将更高程度信任授予这些控件，而不会削弱它们在其他 Intranet/Internet 代码方面的安全性。对程序集签名后，用户必须创建关联有足够权限的新代码组，并指定只允许拥有用户的公司或组织签名的代码成为该代码组成员。以此方式修改安全策略后，高度信任控件将收到足够的权限来执行操作。

修改安全策略后，高度信任的控件才能正常运行，所以在企业的 Intranet 上部署这种类型的控件要容易得多，企业的 Intranet 通常设有企业管理员，管理员可将描述的策略更改部署到多个客户端计算机上。为了让没有共同组织关系的一般用户通过 Internet 使用高度信任控件，控件发行者和用户之间必须有一种信任关系。最终，用户必须愿意使用发行者的说明来修改策略，并允许高度信任控件执行。