

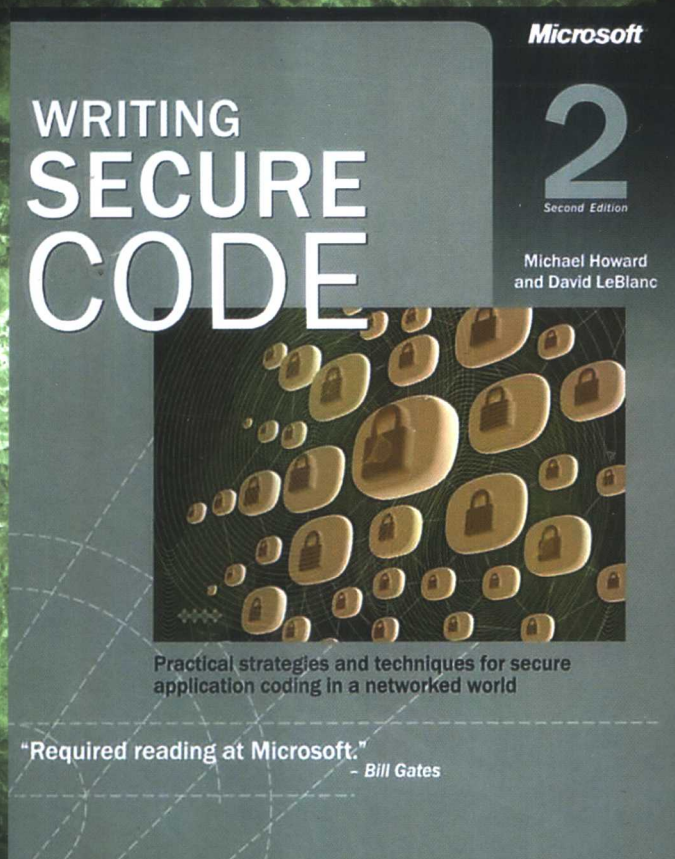
Microsoft
Press

计 算 机 科 学 丛 书

第2版

编写安全的代码

(美) Michael Howard David LeBlanc 著 程永敬 翁海燕 朱涛江 等译



Writing Secure Code
Second Edition



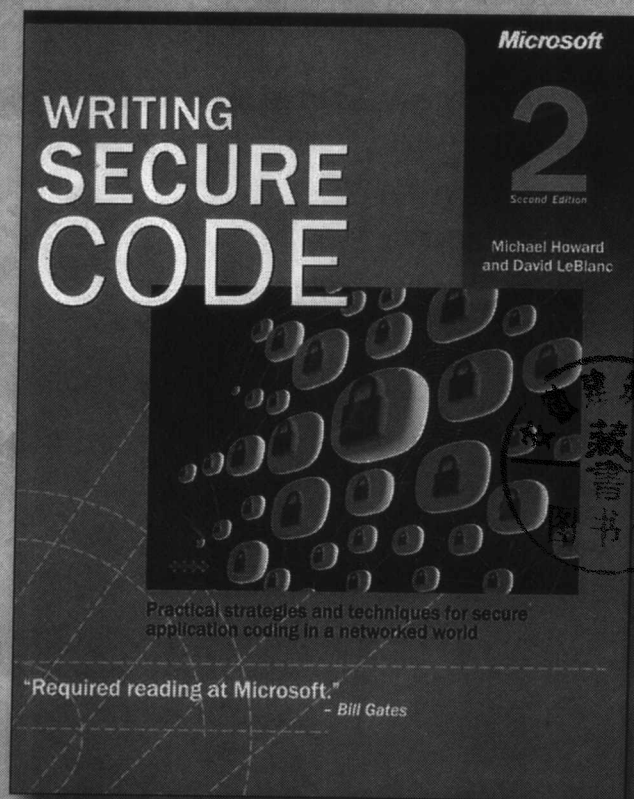
机械工业出版社
China Machine Press

计 算 机 科 学 丛 书

第2版

编写安全的代码

(美) Michael Howard David LeBlanc 著 程永敬 翁海燕 朱涛江 等译



Writing Secure Code
Second Edition

 机械工业出版社
China Machine Press

本书分为五部分。第一部分概括说明了保护系统安全的意义,以及设计安全系统的原则和所采用的技术;第二部分概括介绍了几乎适用于任何应用程序的重要编码技术;第三部分重点介绍了网络应用程序和.Net代码安全;第四部分讨论的安全问题在一般图书中少有论及,如测试、进行安全代码审查、隐私策略以及安全的软件安装等问题;第五部分是附录,分别介绍危险的API以及分别适用于设计人员、开发人员和测试人员的安全措施核对清单。

本书第1版曾作为Windows开发组全体成员的必读教材,而第2版更是总结了在针对微软产品的多次安全活动中的许多新发现。本书曾被比尔·盖茨指定为微软员工必读书籍,是软件设计、开发、测试、系统管理等人员必读的优秀教材。

Michael Howard, David LeBlanc: Writing Secure Code, Second Edition (ISBN 0-7356-1722-8).

Copyright 2005 by Microsoft Corporation.

Original English language edition copyright © 2002 by Microsoft Corporation.

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A. All rights reserved.

本书中文简体字版由美国微软出版社授权机械工业出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号:图字:01-2004-5741

图书在版编目(CIP)数据

编写安全的代码(第2版)/(美)霍华德(Howard, M.), (美)莱布兰克(LeBlanc, D.)著;程永敬等译. —北京:机械工业出版社, 2005.1
(计算机科学丛书)
书名原文: Writing Secure Code, Second Edition
ISBN 7-111-11210-5

I. 编… II. ①霍… ②莱… ③程… III. 电子计算机-安全技术 IV. TP309

中国版本图书馆CIP数据核字(2004)第100858号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑:刘晖 刘立卿

北京中兴印刷有限公司印刷·新华书店北京发行所发行

2005年1月第2版第1次印刷

787mm×1092mm 1/16·31.25印张

印数:0 001-5 000册

定价:55.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010) 68326294

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证,但我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方式如下:

电子邮件: hzedu@hzbook.com

联系电话: (010) 68995264

联系地址: 北京市西城区百万庄南街1号

邮政编码: 100037

专家指导委员会

(按姓氏笔画顺序)

尤晋元
石教英
张立昂
邵维忠
周立柱
范明
袁崇义
谢希仁

王珊
吕建
李伟琴
陆丽娜
周克定
郑国梁
高传善
裘宗燕

冯博琴
孙玉芳
李师贤
陆鑫达
周傲英
施伯乐
梅宏
戴葵

史忠植
吴世忠
李建中
陈向群
孟小峰
钟玉琢
程旭

史美林
吴时霖
杨冬青
周伯生
岳丽华
唐世渭
程时端

秘 书 组

武卫东

温莉芳

刘江

杨海玲

前 言

在2002年2、3月份，人们都顾不上考虑Windows所有正常的安全特性了。在此期间，整个开发组的注意力都转向了如何提高此产品的下一版本——Windows Server 2003——的安全上。众所周知，“安全Windows倡议”的目的是使开发组成员了解最新的安全编码技术，从而找出设计缺陷、代码缺陷，改进测试代码和文档。在此期间，本书的第1版是Windows开发组全体成员的必读教材，而第2版收录了在这场活动中以及在针对微软其他产品的后续安全活动中的许多新发现，这些产品包括SQL Server、Office、Exchange、Systems Management Server、Visual Studio .NET、.NET公共语言运行库，等等。

Windows安全活动（以及许多其他的安全活动）的出发点，源于比尔·盖茨2002年1月15日发表的“可信计算”备忘录，其中概括说明了一种高级策略，即如何向用户提交更安全、更可靠的新型计算机系统。自从有了那个备忘录，我们已经与数千名开发人员交谈、合作过，而他们异口同声地说：“我们要做正确的事情……我们要开发安全的软件……但我们知道的还不够。”这种呼声是本书的直接催化剂：向他们传授学校里不教的技术——如何设计、生成、测试和注释安全的软件。所谓“安全的软件”，并不是指安全代码或实现安全特性的代码，而是指为抵挡恶意攻击而设计的代码。安全的代码也是健壮的代码。

本书的目标是绝对要保证切实可行，书中可能使读者产生不安全感，可能会感到自己的代码也许会受到攻击。更确切地说，如果你创建的应用程序要运行在一台或多台计算机上，而这些计算机连到了网络上，或最大的网络Internet上，那么你的代码就可能会受到攻击。

系统安全受到威胁后会造成各种严重后果，如生产力下降、信誉受损以及财产损失，如果攻击者能够破坏应用程序的安全（如，使之无法使用），你的客户就可能会转而使用竞争对手的产品。在使用基于Internet的服务时，大多数人都没有耐心，如果你的服务不好用，客户也会抛弃你的竞争对手。

对于众多软件开发商来说，真正的问题是，安全所带来的效益不像开发过程那样是显而易见的。因此，管理层不愿意在培训开发人员以编写安全的代码方面多花钱，在攻击得逞之前，他们不想在安全技术上投资。然而，亡羊补牢，则为时已晚。事后的修补工作，无论是在财力上还是在信誉上，都要付出高昂的代价。

历史证明，最好的方法是保护财产，使之免遭盗窃和攻击。我们的祖先早就制定了对盗窃、损坏或侵占他人财物者的惩罚措施。的确，人们都明白，私人财物应当受到保护。这些道德准则也适用于数字世界，因此，作为开发人员，我们的部分职责就是创建可保护数字资产的应用程序和解决方案。

本书包括一些基本概念，这些内容在学校课程里介绍“设计和构建安全的系统”主题时应当讲过。读者可能会认为，“设计”是设计人员或管理人员的事情，但作为开发人员和测试人员，同样也需要了解设计良好的系统结构以抵御攻击的有关过程。

众所周知，无论花多少时间和精力来开发软件，开发出的软件仍然会存在漏洞，因为人们无法预测未来的安全问题。对于Windows Server 2003来说，同样也存在一些漏洞，但按照本书

中所建议的方法，相信可以减少漏洞数，使黑客更难于发现和利用代码中的漏洞。

本书读者对象

本书适用于设计应用程序或生成、测试以及注释解决方案的人员。如果编写基于Web或基于Win32的应用程序，或者正在学习或开发基于Microsoft .NET框架的应用程序，本书同样是很好的读物。总之，只要工作中涉及到应用程序开发，都会在本书中找到许多值得学习的内容。

即使是正在编写不在Microsoft平台上运行的代码，书中的许多内容依然非常有用。除了个别几章是完全针对Microsoft平台以外，许多问题往往与平台无关。即便有时某种东西似乎仅适用于Windows，但其通常也有更加广泛的应用。例如，Everyone用户的“安全控制”访问控制列表和UNIX系统上设置为“全部可写”的文件，其实是同一问题，而跨网站的脚本问题是普遍存在的。

本书组织结构

本书分为五个部分。第一部分（第1~4章）概括说明了为什么要保护系统安全免遭攻击，以及设计这种系统的原则和分析技术。

第二部分和第三部分是本书的重点。第二部分（第5~14章）概括介绍了几乎适用于任何一种应用程序的重要的编码技术。第三部分（第15~18章）重点介绍了网络应用程序和.NET代码。

第四部分（第19~24章）讲述了一般图书中很少讨论的一些主题，如测试、进行安全代码审查、隐私策略以及安全的软件安装等问题。第23章介绍了放在其他各章都不太合适的一些一般原则。

第五部分是“附录”，包括5个附录，分别介绍危险的API，我们常听到的一些不考虑安全问题的荒谬借口，以及分别针对设计人员、开发人员和测试人员的安全措施核对清单。

与其他作者不同，我们不仅告诉你应用程序如何不安全，而且告诉你为什么人们不愿意构建安全的系统。本书绝对实用，书中解释了系统怎样会受到攻击，人们常犯的错误，以及如何构建安全的系统——这才是最重要的。

安装和使用范例文件

通过连接到站点<http://www.microsoft.com/mspress/books/5957.asp>，可以从Web上下载本书的范例文件（从Companion Content页下载）。要访问范例文件，请单击该页右侧More Information框中的Companion Content链接，打开Companion Content页，此页中包括下载范例文件的链接；也可以连接到Microsoft Press Support站点。下载链接可打开一个包含许可协议的可执行文件。要想把范例文件复制到硬盘上，请单击运行可执行文件的链接，然后接受许可协议。默认情况下，范例文件将被复制到My Document\Microsoft Press\Secureco2文件夹下。在安装过程中，可以改变目标文件夹。

系统要求

尽管可以使用包括Visual C++6.0在内的大多数编译器来编译本书中用C/C++编写的大多数范例，但还是要求读者安装Microsoft Visual Studio .NET。用Perl编写的范例已经使用ActiveState

Perl 5.6或ActivateState Visual Perl 1.0（可从<http://www.activestate.com>下载）测试过。VBScript和JScript代码已经用Windows Scripting Host测试过（Windows 2000及其以后的版本中包含Windows Scripting Host）。所有SQL范例都使用SQL Server 2000测试过。而Visual Basic .NET和Visual C#应用程序均是使用Visual Studio .NET编写和测试的。

本书中的所有应用程序（有两个例外），均可以在符合建议的系统要求的、运行Windows 2000的计算机上运行。第7章的Safer范例和第11章的UTF8 MultiByteToWideChar范例，必须在Windows XP或Windows Server 2003上才能正确运行。编译代码时必须使用更加健壮的机器（即比编译器所要求的基本条件更优）。

目 录

出版者的话

专家指导委员会

前言

第一部分 现代安全

第1章 对安全系统的需求	2
1.1 “疯狂的Web网”上的应用程序	3
1.2 可信计算的需要	4
1.3 让每个人都参与进来	5
1.3.1 巧妙地向企业推销安全	5
1.3.2 使用搞破坏的方法	7
1.4 灌输安全意识的一些主意	8
1.4.1 让老板发一封电子邮件	9
1.4.2 任命安全宣传员	9
1.5 攻击者的优势和防御者的劣势	12
1.5.1 因素1：防御者必须对所有的环节 都进行防御，而攻击者可以选择最 薄弱的环节	12
1.5.2 因素2：防御者只能针对已知的 攻击进行防御，而攻击者则可以 探测未知的漏洞	13
1.5.3 因素3：防御者必须永远保持警惕， 而攻击者却可以随意“罢工”	13
1.5.4 因素4：防御者的活动必须遵循 相应的规则，而攻击者则可以采 用一些卑鄙的手段	13
1.6 本章小结	13
第2章 主动的安全开发过程	14
2.1 不断改进开发过程	15
2.2 安全教育角色	15
2.2.1 强制培训的阻力	17
2.2.2 不断更新的培训	18
2.2.3 安全科学的进步	18
2.2.4 教育证明“更多的眼睛”不代表 更安全	18

2.2.5 有力的证据!	19
2.3 设计阶段	19
2.3.1 面试期间的安全问题	19
2.3.2 定义产品的安全目标	20
2.3.3 安全是产品的一种特性	22
2.3.4 要有足够的时间考虑安全问题	24
2.3.5 安全的设计源于威胁建模	24
2.3.6 终结不安全的特性	24
2.3.7 设置bug门槛	24
2.3.8 安全小组审查	25
2.4 开发阶段	25
2.4.1 只有核心成员能够查看新代码 (签字确认)	25
2.4.2 新代码的同级安全审查(签字 确认)	25
2.4.3 定义安全的编码准则	26
2.4.4 审查旧的缺陷	26
2.4.5 外部安全审查	26
2.4.6 安全推动活动	26
2.4.7 留心自己的错误数量	27
2.4.8 记录错误	27
2.4.9 没有惊喜，也没有圣诞节彩蛋!	27
2.5 测试阶段	28
2.6 发行和维护阶段	28
2.6.1 如何知道已完成	28
2.6.2 响应过程	28
2.6.3 责任制	29
2.7 本章小结	29
第3章 赖以生存的安全法则	30
3.1 设计安全、默认安全和部署安全	30
3.1.1 设计安全	30
3.1.2 默认安全	31
3.1.3 部署安全	31
3.2 安全法则	31
3.2.1 从错误中吸取教训	32

3.2.2	尽可能缩小攻击面	33
3.2.3	采用安全的默认设置	34
3.2.4	纵深防御	35
3.2.5	使用最小特权	36
3.2.6	向下兼容总是不安全的	37
3.2.7	假设外部系统是不安全的	38
3.2.8	失败时的应对计划	38
3.2.9	失败时进入安全模式	38
3.2.10	切记：安全特性不等于安全的特性	40
3.2.11	决不要将安全仅维系于隐匿	40
3.2.12	不要将代码与数据混在一起	40
3.2.13	正确地解决安全问题	40
3.3	本章小结	41
第4章	威胁建模	42
4.1	通过威胁建模进行安全的设计	42
4.1.1	成立威胁建模小组	43
4.1.2	分解应用程序	44
4.1.3	确定系统所面临的威胁	49
4.1.4	以风险递减的顺序给威胁分级	54
4.1.5	选择应付威胁的方法	63
4.1.6	选择缓和和威胁的方法	64
4.2	安全技术	64
4.2.1	身份认证	64
4.2.2	授权	68
4.2.3	防篡改和增强保密性的技术	69
4.2.4	保护秘密或最好不要保存秘密	69
4.2.5	加密、散列、MAC和数字签名	69
4.2.6	审核	70
4.2.7	过滤、节流和服务质量	70
4.2.8	最小特权	70
4.3	缓和工资表范例程序的威胁	71
4.4	各种威胁及解决方案	71
4.5	本章小结	71

第二部分 安全的编码技术

第5章	头号公敌：缓冲区溢出	76
5.1	堆栈溢出	77
5.2	堆溢出	83
5.3	数组下标错误	87
5.4	格式字符串错误	89
5.5	Unicode和ANSI缓冲区大小不匹配	93
5.6	预防缓冲区溢出	95
5.6.1	字符串处理方面的安全问题	96
5.6.2	关于字符串处理函数的警告	103
5.7	Visual C++ .NET的/GS选项	104
5.8	本章小结	106
第6章	确定适当的访问控制	107
6.1	ACL何以如此重要	107
6.2	ACL的组成	109
6.3	选择好的ACL的方法	111
6.4	创建ACL	113
6.4.1	在Windows NT 4中创建ACL	113
6.4.2	在Windows 2000中创建ACL	116
6.4.3	用活动模板库创建ACL	119
6.5	对ACE进行正确的排序	120
6.6	留意终端服务器和远程桌面的SID	122
6.7	NULL DACL和其他的危险ACE类型	123
6.7.1	NULL DACL和审核	124
6.7.2	危险的ACE类型	124
6.7.3	如果无法改变NULL DACL该怎么办	125
6.8	其他的访问控制机制	125
6.8.1	.NET框架角色	126
6.8.2	COM+角色	127
6.8.3	IP限制	127
6.8.4	SQL Server触发器和权限	128
6.8.5	一个医学方面的示例	128
6.8.6	关于访问控制机制的重要说明	129
6.9	本章小结	130
第7章	以最小特权运行	131
7.1	现实中的最小特权	131
7.1.1	病毒和特洛伊木马	132
7.1.2	破坏Web服务器	132
7.2	访问控制简介	133
7.3	特权简介	133
7.3.1	SeBackupPrivilege问题	134

7.3.2 SeRestorePrivilege问题	136	8.1.2 Win32中的加密随机数	168
7.3.3 SeDebugPrivilege问题	136	8.1.3 托管代码中的加密随机数	172
7.3.4 SeTcbPrivilege问题	136	8.1.4 Web页中的加密随机数	172
7.3.5 SeAssignPrimaryTokenPrivilege 和SeIncreaseQuotaPrivilege问题	137	8.2 使用口令生成加密密钥	173
7.3.6 SeLoadDriverPrivilege问题	137	8.3 密钥管理问题	174
7.3.7 SeRemoteShutdownPrivilege问题	137	8.3.1 长期密钥和短期密钥	175
7.3.8 SeTakeOwnershipPrivilege问题	137	8.3.2 使用合适的密钥长度保护数据	176
7.4 令牌简介	137	8.3.3 将密钥保存在靠近数据源的地方	176
7.5 令牌、特权、SID、ACL和进程之间 的关系	138	8.3.4 密钥交换问题	179
7.6 应用程序要求提高特权的三个理由	139	8.4 创建自己的加密函数	180
7.6.1 ACL问题	139	8.5 使用相同的流密码加密密钥	181
7.6.2 特权问题	140	8.5.1 人们为什么使用流密码	182
7.6.3 使用LSA秘密	140	8.5.2 流密码的缺陷	182
7.7 解决提高特权的问题	140	8.5.3 如果必须使用相同的密钥怎么办	184
7.7.1 解决ACL问题	140	8.6 针对流密码的位替换攻击	185
7.7.2 解决特权问题	141	8.6.1 解决位替换攻击	185
7.7.3 解决LSA问题	141	8.6.2 何时使用散列、密钥散列或数字 签名	186
7.8 确定适当特权的过程	141	8.7 重用明文和密文的缓冲区	190
7.8.1 步骤1: 找到应用程序使用的 资源	141	8.8 使用加密技术缓和威胁	191
7.8.2 步骤2: 找到应用程序使用的 特权API	142	8.9 在文档中说明你使用的加密算法	191
7.8.3 步骤3: 哪一个帐户是必需的	142	8.10 本章小结	191
7.8.4 步骤4: 获取令牌的内容	143	第9章 保护机密数据	192
7.8.5 步骤5: 所有SID和特权是否都是 必需的	147	9.1 攻击机密数据	192
7.8.6 步骤6: 调整令牌	148	9.2 有时并不需要保存秘密	193
7.9 Windows XP和Windows Server 2003 中的低特权级服务帐户	158	9.2.1 创建干扰散列	193
7.10 模拟特权和Windows Server 2003	159	9.2.2 使用PKCS #5增加攻击的难度	194
7.11 调试最小特权问题	160	9.3 获取用户的秘密信息	195
7.11.1 为什么以普通用户运行时应用 程序失败	160	9.4 保护Windows 2000及其以后版本 中的秘密信息	196
7.11.2 如何判断应用程序失败的原因	160	9.5 保护Windows NT 4中的秘密信息	199
7.12 本章小结	165	9.6 保护Windows 95/98/ME/CE中的 秘密	202
第8章 加密的弱点	166	9.7 不要选择最低共同点解决方案	205
8.1 使用不良的随机数	166	9.8 管理内存中的秘密	206
8.1.1 问题: rand函数	166	9.8.1 编译器优化警告	207
		9.8.2 对内存中的机密数据进行加密	210
		9.9 锁定内存以防敏感数据被分页	211
		9.10 保护托管代码中的机密数据	211
		9.11 提高安全门槛	217

9.11.1	把数据存储在FAT系统的文件中	217	11.2.4	Sun公司的StarOffice /tmp目录 的符号链接漏洞	236
9.11.2	使用嵌入密钥和XOR对数据进行 编码	217	11.2.5	常见的Windows规范文件名 错误	237
9.11.3	使用嵌入密钥和3DES加密数据	218	11.3	基于Web的规范问题	241
9.11.4	使用3DES加密数据并把密码存 放在注册表中	218	11.3.1	绕过AOL的父母控制	241
9.11.5	使用3DES加密数据并把强密钥 存储在注册表中	218	11.3.2	绕过eEye的安全检查	241
9.11.6	使用3DES加密数据,把强密钥 存储在注册表中,并使用ACL 控制文件和注册表项	218	11.3.3	安全区域和E4的“无点IP地址” 错误	242
9.11.7	使用3DES加密数据,把强密钥 存储在注册表中,要求用户输入 密码,并使用ACL控制文件和注 册表项	218	11.3.4	IIS 4.0的::\$DATA漏洞	242
9.12	保护机密数据时的折衷方案	218	11.3.5	何时一行变成了两行	244
9.13	本章小结	219	11.3.6	另一个Web问题——转义	244
第10章	一切输入都是有害的	220	11.4	视觉等效攻击和同形异义词攻击	247
10.1	问题	220	11.5	预防规范化错误	247
10.2	误信他人	221	11.5.1	不要根据文件名进行决策	247
10.3	防御输入攻击的策略	222	11.5.2	使用正则表达式限制文件名的 格式	248
10.4	如何检查合法性	223	11.5.3	停止生成8.3格式的文件名	249
10.5	Perl中被污染的变量	225	11.5.4	不要相信PATH环境变量,要 使用完整的路径名	249
10.6	使用正则表达式检查输入	225	11.5.5	尝试规范化文件名	249
10.7	正则表达式和Unicode	228	11.5.6	安全地调用CreateFile	252
10.8	正则表达式的“罗塞塔石碑”	231	11.6	基于Web的规范化问题的补救措施	253
10.8.1	Perl中的正则表达式	231	11.6.1	限制合法输入	253
10.8.2	托管代码中的正则表达式	231	11.6.2	处理UTF-8字符时要谨慎	253
10.8.3	脚本中的正则表达式	232	11.6.3	ISAPI—岩石和硬地之间	254
10.8.4	C++中的正则表达式	232	11.7	最后的考虑:非基于文件的规范化 问题	254
10.9	不使用正则表达式的最佳做法	233	11.7.1	服务器名	254
10.10	本章小结	234	11.7.2	用户名	255
第11章	规范表示的问题	235	11.8	本章小结	257
11.1	规范的含义及其存在的问题	235	第12章	数据库输入问题	258
11.2	规范文件名的问题	235	12.1	问题	258
11.2.1	绕过Napster名称过滤	235	12.2	伪补救措施1:用引号将输入括 起来	260
11.2.2	Apple Mac OS X和Apache的 漏洞	236	12.3	伪补救措施2:使用存储过程	260
11.2.3	DOS设备名漏洞	236	12.4	补救措施1:永不以sysadmin身份 连接	261
			12.5	补救措施2:以安全的方式创建 SQL语句	262

12.6 纵深防御示例	263	14.4.2 不要使用LCMapString验证 字符串	286
12.7 本章小结	266	14.4.3 使用CreateFile验证文件名	286
第13章 Web特有的输入问题	267	14.5 字符集转换问题	287
13.1 跨网站脚本:输出何时变坏了	267	14.6 调用MultiByteToWideChar时使用 MB_PRECOMPOSED和MB_ERR_ INVALID_CHARS	287
13.1.1 有时攻击者不需要<SCRIPT>块	270	14.7 调用WideCharToMultiByte时使用 WC_NO_BEST_FIT_CHARS	287
13.1.2 攻击者不需要用户点击链接!	270	14.8 比较和排序	289
13.2 与XSS有关的其他攻击	270	14.9 Unicode字符属性	290
13.2.1 针对本地文件的XSS攻击	270	14.10 规范化	291
13.2.2 针对HTML资源的XSS攻击	272	14.11 本章小结	291
13.3 XSS的补救措施	272		
13.3.1 将输出编码	272	第三部分 更安全的编码技术	
13.3.2 为所有的标签属性加上双引号	272		
13.3.3 将数据插入innerText属性	273	第15章 Socket安全	294
13.3.4 强制使用代码页	273	15.1 避免服务器被劫持	294
13.3.5 IE 6.0 SPI的cookie选项 HttpOnly	274	15.2 TCP窗口攻击	300
13.3.6 IE的“Web标记”	275	15.3 选择服务器接口	300
13.3.7 IE的<FRAME SECURITY> 属性	276	15.4 接受连接	300
13.3.8 ASP.NET 1.1的ValidateRequest 配置选项	276	15.5 编写防火墙友好的应用程序	304
13.4 不要只是寻找不安全的结构	277	15.5.1 只用一个连接工作	305
13.5 我只是想让用户向我的Web站点发送 HTML	278	15.5.2 不要要求服务器连接回客户端	305
13.6 如何审查代码中的XSS错误	279	15.5.3 使用基于连接的协议	305
13.7 基于Web的其他安全主题	279	15.5.4 不要通过另外一个协议使你的 应用程序进行多路复用	306
13.7.1 eval()可能是坏的	279	15.5.5 不要在应用层数据中嵌入主机 IP地址	306
13.7.2 HTTP信任问题	279	15.5.6 让你的应用程序可配置	306
13.7.3 ISAPI应用程序和过滤器	280	15.6 欺骗、基于主机和基于端口的信任	306
13.7.4 警惕“可预知的Cookie”	282	15.7 IPv6即将到来	307
13.7.5 SSL/TLS客户端的问题	283	15.8 本章小结	308
13.8 本章小结	283	第16章 RPC、ActiveX控件和DCOM 安全	309
第14章 国际化问题	284	16.1 RPC入门	309
14.1 I18N安全的黄金准则	284	16.1.1 什么是RPC	310
14.2 在应用程序中使用Unicode	284	16.1.2 创建RPC应用程序	310
14.3 预防I18N缓冲区溢出	285	16.1.3 RPC应用程序的通信原理	311
14.4 验证I18N	286	16.2 RPC安全最佳实践	312
14.4.1 视觉验证	286		

16.2.1	使用/robust MIDL 开关参数	312
16.2.2	使用[range] 属性	313
16.2.3	要求对连接进行验证	313
16.2.4	使用数据包的保密性和完整性	317
16.2.5	使用严格的上下文句柄	318
16.2.6	不要依靠上下文句柄来进行 访问检查	319
16.2.7	注意空的上下文句柄	320
16.2.8	不要信任你的对等端	321
16.2.9	使用安全回调	321
16.2.10	在单一进程中驻留多个RPC 服务器	323
16.2.11	使用主流的协议	324
16.3	DCOM安全最佳实践	325
16.3.1	DCOM基础	325
16.3.2	应用层的安全	326
16.3.3	DCOM用户上下文环境	326
16.3.4	可编程实现的安全性	328
16.3.5	源端和接收端	331
16.4	ActiveX入门	331
16.5	ActiveX安全最佳实践	331
16.5.1	什么样的ActiveX组件是初始化 安全和脚本安全的	332
16.5.2	初始化安全和脚本安全的最佳 实践	333
16.6	本章小结	335
第17章	拒绝服务攻击的防范	336
17.1	应用程序失败攻击	336
17.2	CPU不足攻击	339
17.3	内存不足攻击	344
17.4	资源不足攻击	345
17.5	网络带宽攻击	346
17.6	本章小结	346
第18章	编写安全的.NET代码	348
18.1	代码访问安全概述	349
18.2	FxCop：“必备的”工具	350
18.3	程序集是强命名的	351
18.4	指定程序集权限需求	352
18.4.1	请求最小的权限集	353
18.4.2	拒绝不必要的权限	353

18.4.3	请求可选的权限	353
18.5	过分热衷于使用Assert方法	354
18.6	关于Demand和Assert方法的进一步 信息	355
18.7	及时禁用断言	356
18.8	请求和链接请求	357
18.9	慎用SuppressUnmanagedCodeSecurity Attribute属性	358
18.10	远程请求	359
18.11	限制代码的使用范围	359
18.12	不要在XML或配置文件中存储 敏感数据	360
18.13	审查允许部分信任的程序集	361
18.14	检查非托管代码的托管包装的 正确性	362
18.15	委托的问题	362
18.16	序列化的问题	362
18.17	隔离存储的作用	363
18.18	在部署ASP.NET应用程序之前 禁用跟踪和调试	364
18.19	不要远程发布冗长的错误信息	364
18.20	对来源不可信的数据进行反序 列化	365
18.21	失败时不要让攻击者知道太多	365
18.22	本章小结	366

第四部分 特殊的安全问题

第19章 安全性测试	368
19.1 安全性测试人员的角色	368
19.2 安全性测试与一般测试的区别	368
19.3 根据威胁模型制定安全性测试计划	369
19.3.1 分解应用程序	369
19.3.2 确定组件接口	370
19.3.3 按照潜在的漏洞对接口进行 分级	371
19.3.4 确定每一个接口使用的数据 结构	371
19.3.5 STRIDE类型的攻击程序	372
19.3.6 用数据变种攻击应用程序	373

19.3.7 测试之前	381	22.4.2 隐私倡导者的作用	421
19.3.8 开发查找缺陷的工具	381	22.5 设计尊重隐私的应用程序	421
19.4 用欺诈性的服务程序测试客户软件	394	22.5.1 在开发过程中加入隐私策略	421
19.5 用户是否应看到或修改数据	395	22.5.2 了解隐私的特点	423
19.6 用安全模板进行测试	395	22.6 本章小结	429
19.7 发现一个错误时测试并未结束	396	第23章 常用的好做法	430
19.8 测试码应有很高的质量	397	23.1 不要向攻击者透露任何信息	430
19.9 测试端到端解决方案	397	23.2 关于服务的最佳做法	430
19.10 确定攻击面	397	23.2.1 安全、服务和交互式桌面	430
19.10.1 确定根攻击向量	398	23.2.2 服务帐户准则	431
19.10.2 确定攻击向量的偏差	398	23.3 不要以标志字符串的形式泄漏 信息	433
19.10.3 统计产品中有偏差的攻击向量	398	23.4 在补丁中改变错误信息时要谨慎	433
19.11 本章小结	399	23.5 复查错误路径	433
第20章 审查安全代码	400	23.6 让它保持关闭	433
20.1 处理大型应用程序	401	23.7 核心态错误	433
20.2 多遍审查方法	401	23.7.1 高级安全问题	433
20.3 从易处着手	402	23.7.2 句柄	434
20.4 整数溢出	403	23.7.3 符号链接	434
20.5 检查返回结果	405	23.7.4 配额	435
20.6 对指针代码进行额外的审查	406	23.7.5 序列化原语	435
20.7 绝不要相信网络上的数据	406	23.7.6 缓冲区处理问题	435
20.8 本章小结	407	23.7.7 IRP取消	436
第21章 安全的软件安装	408	23.8 在代码中添加关于安全的注释	437
21.1 最小特权原则	408	23.9 借助于操作系统的功能	437
21.2 安装后立即清除密码	410	23.10 不要依赖用户去做正确的选择	437
21.3 使用安全配置编辑器	410	23.11 安全地调用CreateProcess函数	438
21.4 低层的安全API	415	23.11.1 不要将lpApplicationName 设置为NULL	439
21.5 使用Windows Installer	416	23.11.2 用引号把pCommandLine中 可执行文件的路径括起来	439
21.6 本章小结	416	23.12 不要创建共享的/可写的代码段	439
第22章 在应用程序中加入隐私策略	417	23.13 正确使用模拟函数	440
22.1 对隐私的恶意侵犯和令人讨厌的 侵犯	417	23.14 不要将用户文件写入\Program Files 目录	440
22.2 主要的隐私立法	417	23.15 不要把用户数据写入HKLM	440
22.2.1 个人身份信息	418	23.16 不要以“完全控制”权限打开 对象	441
22.2.2 关于数据保护的欧盟法令	418	23.17 对象创建错误	441
22.2.3 安全海港原则	418	23.18 慎用CreateFile	442
22.2.4 其他隐私立法	419		
22.3 隐私与安全	419		
22.4 建立隐私基础设施	420		
22.4.1 首席隐私官的作用	420		

23.19	安全地创建临时文件	443
23.20	Setup程序和EFS文件系统的问题	445
23.21	文件系统重解析点问题	445
23.22	客户端安全是自相矛盾的说法	446
23.23	范例就是模板	446
23.24	以身作则，亲身体验	447
23.25	你得向用户负责	447
23.26	基于管理员SID确定访问权限	447
23.27	允许使用长口令	448
23.28	慎用_alloca	448
23.29	ATL转换宏	449
23.30	不要嵌入公司的名称	449
23.31	将字符串移至资源DLL中	450
23.32	应用程序日志	450
23.33	从危险的C/C++迁移到托管代码	450
第24章	编写安全文档和错误消息	451
24.1	文档中的安全问题	451
24.1.1	关于文档的基础知识	451
24.1.2	通过文档缓和威胁	452
24.1.3	编写安全性文档的最佳做法	452
24.2	错误消息中的安全问题	453
24.3	典型的安全消息	454
24.4	信息泄漏问题	454
24.4.1	知情同意	455
24.4.2	累进泄漏	456
24.4.3	消息要具体	456
24.4.4	最好不要提问	457
24.5	对安全消息进行可用性测试	458
24.6	审阅产品说明书时的注意事项	458
24.7	安全设置的可用性	458
24.8	本章小结	459
第五部分 附 录		
附录A	危险的API	462
附录B	安全误区	469
附录C	设计人员的安全措施核对清单	473
附录D	开发人员的安全措施核对清单	474
附录E	测试人员的安全措施核对清单	477
最后的一点思考		478
参考文献		479

第五部分 附 录

附录A	危险的API	462
附录B	安全误区	469
附录C	设计人员的安全措施核对清单	473
附录D	开发人员的安全措施核对清单	474
附录E	测试人员的安全措施核对清单	477
最后的一点思考		478
参考文献		479