

<<软件安全实现>>

图书基本信息

书名：<<软件安全实现>>

13位ISBN编号：9787302222613

10位ISBN编号：7302222614

出版时间：2010-6

出版时间：清华大学出版社

作者：郭克华 编

页数：267

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>

## &lt;&lt;软件安全实现&gt;&gt;

## 前言

安全编程技术是一门学科，涵盖的编程语言较多，所需要讨论的问题也较广，因此，目前对于安全编程技术的讲解很容易陷入误区：要么只是针对某一门语言讲解安全问题；要么泛泛而谈，缺乏具体案例。

本书针对编程中常见的安全问题进行了阐述，不局限于某一门特定语言，但是每一个话题却以简单、通俗、易懂的案例进行讲解，逐步引领读者从基础到各个知识点进行学习，从而开发出安全可靠的系统。

本书涵盖了内存安全、线程 / 进程安全、异常 / 错误处理安全、输入安全、国际化安全、面向对象的编程安全、web编程安全、权限控制、远程调用和组件安全、避免拒绝服务攻击、数据加密保护、数字签名、安全测试和程序性能调优等内容。

每章后面都有配套练习，用于对本章内容进行总结演练。

1. 本书的知识体系学习本书，需要具有一定的编程基础，至少要对常见语言，如C++、.NET、Java有所了解。

本书的知识体系结构如图1所示，遵循循序渐进的原则，逐步引领读者从基础到各个知识点的学习。

## <<软件安全实现>>

### 内容概要

本书共分为16章，针对安全编程技术进行讲解，主要涵盖了基本安全编程、应用安全编程、数据保护编程以及其他内容共四大部分：第一部分包含内存安全、线程/进程安全、异常/错误处理安全、输入安全，第二部分包含国际化安全、面向对象的编程安全、Web编程安全、权限控制、远程调用和组件安全、避免拒绝服务攻击等内容，第三部分包含数据加密保护、其他保护、数字签名等内容，最后一部分包含软件安全测试和代码性能调优。

每章后面都有配套练习，用于对本章进行总结演练。

针对安全编程技术，本书不局限于某一门特定语言，而是将编程过程中的通用安全问题进行全面总结，逐步引领读者从基础到各个知识点进行学习，以便能开发出安全可靠的系统。

全书内容由浅入深，并辅以大量的实例说明，每一个章节以实际案例为起点进行讲解，通俗易懂。

全书所有实例的源代码均可在清华大学出版社的网站上下载，供读者学习参考使用。

本书可作为有一定编程基础的程序员的学习用书，也可供有经验的开发人员深入学习使用，更可以为高等学校、培训班作为教材使用，对于缺乏安全编程实战经验的程序员而言，阅读本书可以快速积累经验，提高编程水平。

## 书籍目录

第1章 安全编程概述 1.1 软件的安全问题 1.1.1 任何软件都是不安全的 1.1.2 软件不安全性的几种表现 1.1.3 软件不安全的原因 1.2 在软件开发生命周期中考虑安全问题 1.2.1 软件设计阶段威胁建模 1.2.2 安全代码的编写 1.2.3 软件的安全性测试 1.2.4 漏洞响应和产品的维护 1.3 本书的内容 1.3.1 编程中的安全 1.3.2 针对信息安全的编程 1.3.3 其他内容 小结 练习 参考文献第2章 内存安全 2.1 缓冲区溢出 2.1.1 缓冲区 2.1.2 缓冲区溢出 2.1.3 缓冲区溢出案例 2.1.4 堆溢出 2.1.5 缓冲区溢出攻击 2.1.6 防范方法 2.2 整数溢出 2.2.1 整数的存储方式 2.2.2 整数溢出 2.2.3 解决方案 2.3 数组和字符串问题 2.3.1 数组下标问题 2.3.2 字符串格式化问题 小结 练习 参考文献第3章 线程/进程安全 3.1 线程机制 3.1.1 为什么需要线程 3.1.2 线程机制和生命周期 3.2 线程同步安全 3.2.1 线程同步 3.2.2 案例分析 3.2.3 解决方案 3.3 线程协作安全 3.3.1 线程协作 3.3.2 案例分析 3.3.3 解决方案 3.4 线程死锁安全 3.4.1 线程死锁 3.4.2 案例分析 3.4.3 解决方案 3.5 线程控制安全 3.5.1 安全隐患 3.5.2 案例分析 3.5.3 解决方案 3.6 进程安全 3.6.1 进程概述 3.6.2 进程安全问题 小结 练习 参考文献第4章 异常/错误处理中的安全 4.1 异常/错误的基本机制 4.1.1 异常的出现 4.1.2 异常的基本特点 4.2 异常捕获中的安全 4.2.1 异常的捕获 4.2.2 异常捕获中的安全 4.3 异常处理中的安全 4.3.1 finally的使用安全 4.3.2 异常处理的安全 4.4 面向过程异常处理中的安全问题 4.4.1 面向过程的异常处理 4.4.2 安全准则 .....第5章 输入安全第6章 国际化安全第7章 面向对象中的编程安全第8章 Web编程安全第9章 权限控制第10章 远程调用和组件安全第11章 避免拒绝服务攻击第12章 数据的加密保护第13章 数据的其他保护第14章 数字签名第15章 软件安全测试第16章 程序性能调优

## 章节摘录

插图：(5) 维护阶段。

本阶段主要根据软件运行的情况，对软件进行适当修改，以适应新的要求；以及纠正运行中发现的错误。

本阶段工作在已完成对软件的研制（分析、设计、编码和测试）工作并交付使用以后进行，一般所做的工作是编写软件问题报告、软件修改报告。

维护阶段的成本是比较高的，设计不到位或者编码测试考虑不周全，可能会造成软件维护成本的大幅度提高。

以一个中小规模软件为例，如果设计、编码和测试需要一年的时间，在投入使用后，其运行时间可能持续三年。

那么维护阶段也就要持续三年。

这段时间内，软件的维护者除了要解决研制阶段所遇到的各种问题，如排除障碍外，还要扩展软件的功能，提高性能。

所以，事实上，和软件开发工作相比，软件维护的工作量和成本都要大得多。

在实际开发过程中，软件开发并不一定是从第一步进行到最后一步，而是在任何阶段，在进入下一阶段前一般都有一步或几步的回溯。

如在测试过程中发现问题可能要求修改设计，用户可能会提出一些需要来修改需求说明书等。

以下主要基于安全问题，针对软件工程中的各个阶段进行阐述。

1.2.1 软件设计阶段威胁建模软件在设计阶段达到的安全性能，将是软件整个生命周期的基础。

如果在设计阶段没有考虑某些安全问题，那么在编码时就几乎不被考虑。

这些隐患将可能成为致命的缺陷，在后期以更高的代价的形式爆发出来。

所以，安全问题，应该从设计阶段就开始考虑，设计要尽可能完善。

编辑推荐

《软件安全实现:安全编程技术》：高等学校信息安全专业规划教材

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>