

**Microsoft®**

secure software  
DEVELOPMENT SERIES

**BEST PRACTICES**

# WRITING SECURE CODE

# 2

*Second Edition*



Practical strategies and techniques for secure  
application coding in a networked world

**"Required reading at Microsoft."**

*– Bill Gates*

Michael Howard and David LeBlanc

**Microsoft®**

# WRITING SECURE CODE

**2**  
*Second Edition*

Michael Howard  
and David LeBlanc

Practical strategies and proven techniques for building  
secure applications in a networked world

**“Required reading at Microsoft.”**

*- Bill Gates*

# Writing Secure Code

## Table of Contents

Copyright

Contents at a Glance

Table of Contents

Introduction

### Part I. Contemporary Security

#### Chapter 1. The Need for Secure Systems

Applications on the Wild Wild Web

The Need for Trustworthy Computing

Getting Everyones Head in the Game

Some Ideas for Instilling a Security Culture

The Attackers Advantage and the Defenders Dilemma

Summary

#### Chapter 2. The Proactive Security Development Process

Process Improvements

The Role of Education

Design Phase

Development Phase

Test Phase

Shipping and Maintenance Phases

Summary

#### Chapter 3. Security Principles to Live By

SD3: Secure by Design, by Default, and in Deployment

Security Principles

Summary

#### Chapter 4. Threat Modeling

Secure Design Through Threat Modeling

Security Techniques

Mitigating the Sample Payroll Application Threats

A Cornucopia of Threats and Solutions

Summary

### Part II. Secure Coding Techniques

# **Table of Contents**

## **Chapter 5. Public Enemy #1: The Buffer Overrun**

- Stack Overruns
- Heap Overruns
- Array Indexing Errors
- Format String Bugs
- Unicode and ANSI Buffer Size Mismatches
- Preventing Buffer Overruns
- The Visual C++ .NET /GS Option
- Summary

## **Chapter 6. Determining Appropriate Access Control**

- Why ACLs Are Important
- What Makes Up an ACL?
- A Method of Choosing Good ACLs
- Creating ACLs
- Getting the ACE Order Right
- Be Wary of the Terminal Server and Remote Desktop SIDs
- NULL DACLs and Other Dangerous ACE Types
- Other Access Control Mechanisms
- Summary

## **Chapter 7. Running with Least Privilege**

- Least Privilege in the Real World
- Brief Overview of Access Control
- Brief Overview of Privileges
- Brief Overview of Tokens
- How Tokens, Privileges, SIDs, ACLs, and Processes Relate
- Three Reasons Applications Require Elevated Privileges
- Solving the Elevated Privileges Issue
- A Process for Determining Appropriate Privilege
- Low-Privilege Service Accounts in Windows XP and Windows.NET Server 2003
- The Impersonate Privilege and Windows .NET Server 2003
- Debugging Least-Privilege Issues
- Summary

## **Chapter 8. Cryptographic Foibles**

- Using Poor Random Numbers
- Using Passwords to Derive Cryptographic Keys
- Key Management Issues
- Creating Your Own Cryptographic Functions

# **Table of Contents**

Using the Same Stream-Cipher Encryption Key

Bit-Flipping Attacks Against Stream Ciphers

Reusing a Buffer for Plaintext and Ciphertext

Using Crypto to Mitigate Threats

Document Your Use of Cryptography

## **Chapter 9. Protecting Secret Data**

Attacking Secret Data

Sometimes You Dont Need to Store a Secret

Getting the Secret from the User

Protecting Secrets in Windows 2000 and Later

Protecting Secrets in Windows NT 4

Protecting Secrets in Windows 95, Windows 98, Windows Me, and Windows CE

Not Opting for a Least Common Denominator Solution

Managing Secrets in Memory

Locking Memory to Prevent Paging Sensitive Data

Protecting Secret Data in Managed Code

Raising the Security Bar

Trade-Offs When Protecting Secret Data

Summary

## **Chapter 10. All Input Is Evil!**

The Issue

Misplaced Trust

A Strategy for Defending Against Input Attacks

How to Check Validity

Tainted Variables in Perl

Using Regular Expressions for Checking Input

Regular Expressions and Unicode

A Regular Expression Rosetta Stone

A Best Practice That Does Not Use Regular Expressions

Summary

## **Chapter 11. Canonical Representation Issues**

What Does Canonical Mean, and Why Is It a Problem?

Canonical Filename Issues

Canonical Web-Based Issues

Visual Equivalence Attacks and the Homograph Attack

Preventing Canonicalization Mistakes

Web-Based Canonicalization Remedies

# **Table of Contents**

A Final Thought: Non-File-Based Canonicalization Issues

Summary

## **Chapter 12. Database Input Issues**

The Issue

Pseudoremedy #1: Quoting the Input

Pseudoremedy #2: Use Stored Procedures

Remedy #1: Never Ever Connect as sysadmin

Remedy #2: Building SQL Statements Securely

An In-Depth Defense in Depth Example

Summary

## **Chapter 13. Web-Specific Input Issues**

Cross-Site Scripting: When Output Turns Bad

Other XSS-Related Attacks

XSS Remedies

Dont Look for Insecure Constructs

But I Want Users to Post HTML to My Web Site!

How to Review Code for XSS Bugs

Other Web-Based Security Topics

Summary

## **Chapter 14. Internationalization Issues**

The Golden I18N Security Rules

Use Unicode in Your Application

Prevent I18N Buffer Overruns

Validate I18N

Character Set Conversion Issues

Use MultiByteToWideChar with MB\_PRECOMPOSED and MB\_ERR\_INVALID\_CHARS

Use WideCharToMultiByte with WC\_NO\_BEST\_FIT\_CHARS

Comparison and Sorting

Unicode Character Properties

Normalization

Summary

## **Part III. Even More Secure Coding Techniques**

### **Chapter 15. Socket Security**

Avoiding Server Hijacking

TCP Window Attacks

Choosing Server Interfaces

Accepting Connections

# **Table of Contents**

Writing Firewall-Friendly Applications

Spoofing and Host-Based and Port-Based Trust

IPv6 Is Coming!

Summary

## **Chapter 16. Securing RPC, Active**

An RPC Primer

Secure RPC Best Practices

Secure DCOM Best Practices

An ActiveX Primer

Secure ActiveX Best Practices

Summary

## **Chapter 17. Protecting Against Denial of Service Attacks**

Application Failure Attacks

CPU Starvation Attacks

Memory Starvation Attacks

Resource Starvation Attacks

Network Bandwidth Attacks

Summary

## **Chapter 18. Writing Secure .NET Code**

Code Access Security: In Pictures

FxCop: A Must-Have Tool

Assemblies Should Be Strong-Named

Specify Assembly Permission Requirements

Overzealous Use of Assert

Further Information Regarding Demand and Assert

Keep the Assertion Window Small

Demands and Link Demands

Use SuppressUnmanagedCodeSecurityAttribute with Caution

Remoting Demands

Limit Who Uses Your Code

No Sensitive Data in XML or Configuration Files

Review Assemblies That Allow Partial Trust

Check Managed Wrappers to Unmanaged Code for Correctness

Issues with Delegates

Issues with Serialization

The Role of Isolated Storage

Disable Tracing and Debugging Before Deploying ASP.NET Applications

# **Table of Contents**

- Do Not Issue Verbose Error Information Remotely
- Deserializing Data from Untrusted Sources
- Dont Tell the Attacker Too Much When You Fail
- Summary

## **Part IV. Special Topics**

### **Chapter 19. Security Testing**

- The Role of the Security Tester
- Security Testing Is Different
- Building Security Test Plans from a Threat Model
- Testing Clients with Rogue Servers
- Should a User See or Modify That Data?
- Testing with Security Templates
- When You Find a Bug, Youre Not Done!
- Test Code Should Be of Great Quality
- Test the End-to-End Solution
- Determining Attack Surface
- Summary

### **Chapter 20. Performing a Security Code Review**

- Dealing with Large Applications
- A Multiple-Pass Approach
- Low-Hanging Fruit
- Integer Overflows
- Checking Returns
- Perform an Extra Review of Pointer Code
- Never Trust the Data
- Summary

### **Chapter 21. Secure Software Installation**

- Principle of Least Privilege
- Clean Up After Yourself!
- Using the Security Configuration Editor
- Low-Level Security APIs
- Using the Windows Installer
- Summary

### **Chapter 22. Building Privacy into Your Application**

- Malicious vs. Annoying Invasions of Privacy
- Major Privacy Legislation
- Privacy vs. Security



# Table of Contents

Building a Privacy Infrastructure  
Designing Privacy-Aware Applications  
Summary

## Chapter 23. General Good Practices

Dont Tell the Attacker Anything  
Service Best Practices  
Dont Leak Information in Banner Strings  
Be Careful Changing Error Messages in Fixes  
Double-Check Your Error Paths  
Keep It Turned Off!  
Kernel-Mode Mistakes  
Add Security Comments to Code  
Leverage the Operating System  
Dont Rely on Users Making Good Decisions  
Calling CreateProcess Securely  
Dont Create Shared/Writable Segments  
Using Impersonation Functions Correctly  
Dont Write User Files to \Program Files  
Dont Write User Data to HKLM  
Dont Open Objects for FULL\_CONTROL or ALL\_ACCESS  
Object Creation Mistakes  
Care and Feeding of CreateFile  
Creating Temporary Files Securely  
Implications of Setup Programs and EFS  
File System Reparse Point Issues  
Client-Side Security Is an Oxymoron  
Samples Are Templates  
Dogfood Your Stuff!  
You Owe It to Your Users If  
Determining Access Based on an Administrator SID  
Allow Long Passwords  
Be Careful with \_alloca  
Dont Embed Corporate Names  
Move Strings to a Resource DLL  
Application Logging  
Migrate Dangerous C/C++ to Managed Code

## Chapter 24. Writing Security Documentation and Error Messages

# **Table of Contents**

- Security Issues in Documentation
- Security Issues in Error Messages
- A Typical Security Message
- Information Disclosure Issues
- A Note When Reviewing Product Specifications
- Security Usability
- Summary

## **Part V. Appendixes**

- Appendix A. Dangerous APIs
- Appendix B. Ridiculous Excuses Weve Heard
- Appendix C. A Designers Security Checklist
- Appendix D. A Developers Security Checklist
- Appendix E. A Testers Security Checklist
- A Final Thought

Annotated Bibliography

Index