The Security Development Lifecycle: SDL: A Process for Developing Demonstrably More Secure Software
By Michael Howard, Steve Lipner

....
Publisher: **Microsoft Press**
Pub Date: **April 28, 2006**
Print ISBN-10: **0-7356-2214-0**
Print ISBN-13: **978-0-7356-2214-2**
Pages: **304**

Table of Contents | Index

Copyright
Foreword
Introduction
Part I: The Need for the SDL
Chapter 1. Enough Is Enough: The Threats Have Changed
Worlds of Security and Privacy Collide
Another Factor That Influences Security: Reliability
It's Really About Quality
Why Major Software Vendors Should Create More Secure Software
Why In-House Software Developers Should Create More Secure Software
Why Small Software Developers Should Create More Secure Software
Summary
References
Chapter 2. Current Software Development Methods Fail to Produce Secure Software
"Given enough eyeballs, all bugs are shallow"
Proprietary Software Development Methods
Agile Development Methods
Common Criteria
Summary
References
Chapter 3. A Short History of the SDL at Microsoft
First Steps
New Threats, New Responses
Windows 2000 and the Secure Windows Initiative
Seeking Scalability: Through Windows XP
Security Pushes and Final Security Reviews
Formalizing the Security Development Lifecycle
A Continuing Challenge
References
Chapter 4. SDL for Management
Commitment for Success
Managing the SDL

- Summary
- References

Part II: The Security Development Lifecycle Process

- Chapter 5. Stage 0: Education and Awareness
  - A Short History of Security Education at Microsoft
  - Ongoing Education
  - Types of Training Delivery
  - Exercises and Labs
  - Tracking Attendance and Compliance
  - Measuring Knowledge
  - Implementing Your Own In-House Training
  - Key Success Factors and Metrics
  - Summary
  - References
- Chapter 6. Stage 1: Project Inception
  - Determine Whether the Application Is Covered by SDL
  - Assign the Security Advisor
  - Build the Security Leadership Team
  - Make Sure the Bug-Tracking Process Includes Security and Privacy Bug Fields
  - Determine the "Bug Bar"
  - Summary
  - References
- Chapter 7. Stage 2: Define and Follow Design Best Practices
  - Common Secure-Design Principles
  - Attack Surface Analysis and Attack Surface Reduction
  - Summary
  - References
- Chapter 8. Stage 3: Product Risk Assessment
  - Security Risk Assessment
  - Privacy Impact Rating
  - Pulling It All Together
  - Summary
  - References
- Chapter 9. Stage 4: Risk Analysis
  - Threat-Modeling Artifacts
  - What to Model
  - Building the Threat Model
  - The Threat-Modeling Process
  - Using a Threat Model to Aid Code Review
  - Using a Threat Model to Aid Testing
  - Key Success Factors and Metrics
  - Summary
  - References
- Chapter 10. Stage 5: Creating Security Documents, Tools, and Best Practices for Customers
  - Why Documentation and Tools?
  - Creating Prescriptive Security Best Practice Documentation

Creating Tools
Summary
References
Chapter 11. Stage 6: Secure Coding Policies
Use the Latest Compiler and Supporting Tool Versions
Use Defenses Added by the Compiler
Use Source-Code Analysis Tools
Do Not Use Banned Functions
Reduce Potentially Exploitable Coding Constructs or Designs
Use a Secure Coding Checklist
Summary
References
Chapter 12. Stage 7: Secure Testing Policies
Fuzz Testing
Penetration Testing
Run-Time Verification
Reviewing and Updating Threat Models if Needed
Reevaluating the Attack Surface of the Software
Summary
References
Chapter 13. Stage 8: The Security Push
Preparing for the Security Push
Training
Code Reviews
Threat Model Updates
Security Testing
Attack-Surface Scrub
Documentation Scrub
Are We Done Yet?
Summary
References
Chapter 14. Stage 9: The Final Security Review
Product Team Coordination
Threat Models Review
Unfixed Security Bugs Review
Tools-Use Validation
After the Final Security Review Is Completed
Summary
Chapter 15. Stage 10: Security Response Planning
Why Prepare to Respond?
Preparing to Respond
Security Response and the Development Team
Summary
References
Chapter 16. Stage 11: Product Release
References
Chapter 17. Stage 12: Security Response Execution

- Following Your Plan
- Making It Up as You Go
- Knowing What to Skip
- Summary
- References

Part III: SDL Reference Material

- Chapter 18. Integrating SDL with Agile Methods
- Using SDL Practices with Agile Methods
- Augmenting Agile Methods with SDL Practices
- Summary
- References
- Chapter 19. SDL Banned Function Calls
- The Banned APIs
- Why the "n" Functions Are Banned
- Important Caveat
- Choosing StrSafe vs. Safe CRT
- Using StrSafe
- Using Safe CRT
- Other Replacements
- Tools Support
- ROI and Cost Impact
- Metrics and Goals
- References
- Chapter 20. SDL Minimum Cryptographic Standards
- High-Level Cryptographic Requirements
- Cryptographic Algorithm Usage
- Data Storage and Random Number Generation
- References
- Chapter 21. SDL-Required Tools and Compiler Options
- Required Tools
- References
- Chapter 22. Threat Tree Patterns
- Spoofing an External Entity or a Process
- Tampering with a Process
- Tampering with a Data Flow
- Tampering with a Data Store
- Repudiation
- Information Disclosure of a Process
- Information Disclosure of a Data Flow
- Information Disclosure of a Data Store
- Denial of Service Against a Process
- Denial of Service Against a Data Flow
- Denial of Service Against a Data Store
- Elevation of Privilege
- References
- Appendix

Index