Important Note Regarding This Microsoft PowerPoint Presentation

*** Do not include this slide in your presentation ***

- This slide deck has been intentionally provided with very limited graphics and formatting to simplify content integration into your own preferred PowerPoint themes and styles.

Security Development Lifecycle (SDL) Field Content

Microsoft SDL

Threat Modeling Principles

(Level 100)

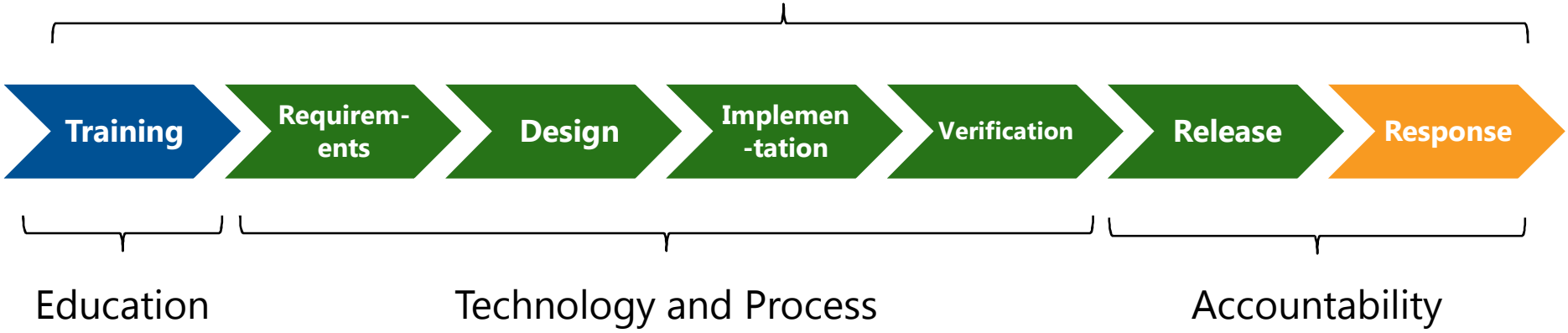
Agenda

- Overview of the Microsoft SDL Threat Modeling process
  - When to threat model
  - Who performs threat modeling
  - What to threat model
- The Microsoft SDL Threat Modeling process
- The Microsoft SDL Threat Modeling Tool
- The Microsoft SDL threat modeling requirements

Microsoft Security Development Lifecycle (SDL)

Delivering secure software requires:

Executive commitment SDL a mandatory policy at Microsoft since 2004



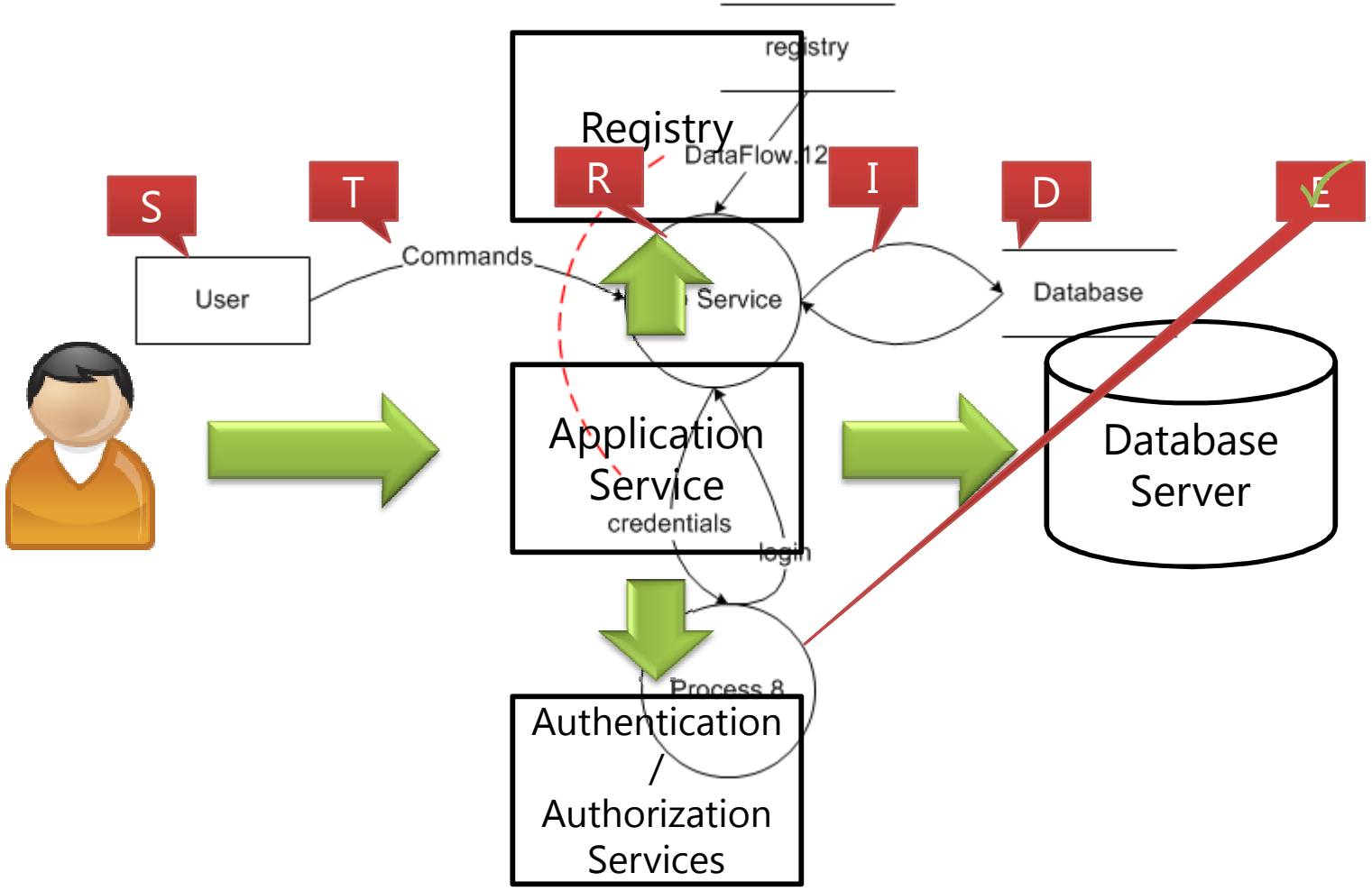
Ongoing Process Improvements 6 month cycle

Microsoft SDL Threat Modeling Overview

Microsoft SDL Threat Modeling: A process to understand security threats to a system, determine risks from those threats, and establish appropriate mitigations

- Key requirement for applications developed with the Microsoft SDL
- Microsoft SDL Threat Modeling
  1. Diagramming
  2. Threat Enumeration
  3. Mitigation
  4. Validation
- Can be performed by both security and non-security experts

Microsoft SDL
Threat Modeling Illustrated



When to Threat Model

- Best performed during the application design phase
  - Easiest to make application changes
  - Less costly than adding mitigations and testing them after code has been implemented and onwards

Who Performs/Drives Threat Modeling?

The Microsoft SDL Threat Modeling process can be performed by:

- Security experts
- Non-security experts

The threat modeling process should be driven by:

- Application designers; however, developers and testers should be involved

What To Threat Model

- The application as a whole
- Security and privacy features
- Features whose failures have security or privacy implications
- Features that cross trust boundaries

Top Microsoft SDL Threat Modeling Advantages and Disadvantages

Advantages

- Can be used to find threats to a system early in the development process
- Can be used by both security experts and non-security experts
- Can be used to guide other security assessment activities

Disadvantages

- Upfront costs (training, software, and setup)

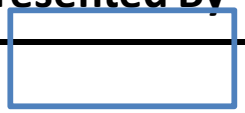
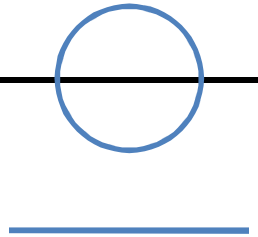
The Microsoft SDL Threat Modeling Process

Step 1: Diagramming

Step Objective: To model an application design as a data flow diagram to drive threat analysis

- Data flow diagrams (DFDs)
  - Widely used and easily understood graphical representation
  - Most attacks based on data flowing through an application or system
- Trust boundaries

Data Flow Diagrams (DFDs) Elements

Element	Represented By	Description
External Entity		Any entity not within the control of the application, such as people and external systems
Process		Code, such as native code executables and .NET assemblies
Data Store		Data at rest, such as registry keys and databases
Data Flow		How data flows between elements, such as function calls and network data

Additional Element: Trust Boundaries

Element	Represented By	Description
Trust Boundary		A point within an application where data flows from one privilege level to another, such as network sockets, external entities and processes with different trust levels

Step 2: Threat Enumeration

Step Objective: To identify threats for each data flow diagram element in the threat model

- **Experts:** Brainstorming and other informal methods
- **Experts and Non-Experts:** STRIDE threat types
  - Based on Microsoft Security Response Center (MSRC) issues and Common Vulnerability and Exposures (CVE) (see http://cve.mitre.org for more information)

STRIDE Threat Types

Desired Property	Threat	Definition
Authentication	**S**poofing	Impersonating something or someone else
Integrity	**T**ampering	Modifying code or data without authorization
Non-repudiation	**R**epudiation	The ability to claim to have not performed some action against an application
Confidentiality	**I**nformation Disclosure	The exposure of information to unauthorized users
Availability	**D**enial of Service	The ability to deny or degrade a service to legitimate users
Authorization	**E**levation of Privilege	The ability of a user to elevate their privileges with an application without authorization

Identifying STRIDE Threats by Data Flow Diagram Element Type

Element	S	T	R	I	D	E
External entity	✓		✓			
Process	✓	✓	✓	✓	✓	✓
Data Store		✓	✓	✓	✓	
Data Flow		✓		✓	✓	

Step 3: Mitigation

Step Objective: To address identified threats to an application design

Approaches to threat mitigation (in order of preference):

1. Redesign
2. Use standard mitigations
3. Use unique mitigations
4. Accept risk in accordance with policies

Examples of Standard Mitigations

Threat	Example Standard Mitigations
Spoofing	IPsec Digital signatures Message authentication codes Hashes
Tampering	ACLs Digital signatures Message Authentication Codes
Repudiation	Strong Authentication Secure logging and auditing
Information Disclosure	Encryption ACLs
Denial of Service	ACLs Quotas High availability designs
Elevation of Privilege	ACLs Group or role membership Input validation

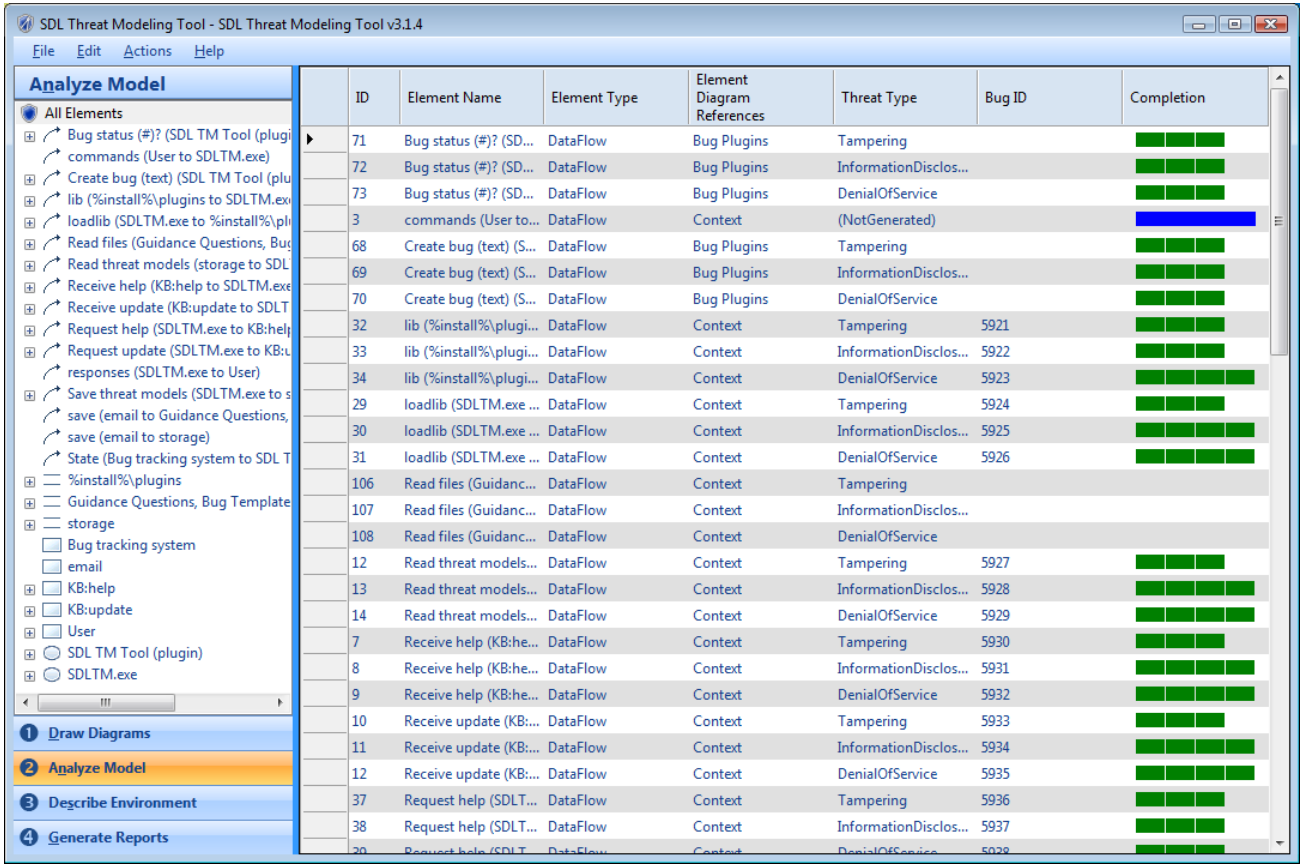
- Refer to Chapter 9 of the Microsoft SDL for a more complete listing
  - http://www.microsoft.com/learning/en/us/books/8753.aspx

Step 4: Validation

Step Objective: To help ensure that threat models accurately reflect application design and potential threats

- Validation of the model
- Validation of enumerated threats
- Validation of mitigations
- Validation of assumptions

The Microsoft SDL Threat Modeling Tool



- Refer to the following link for more information regarding this tool
  - http://msdn.microsoft.com/en-us/security/dd206731.aspx

Microsoft SDL
Threat Modeling Requirements

Threat models must:

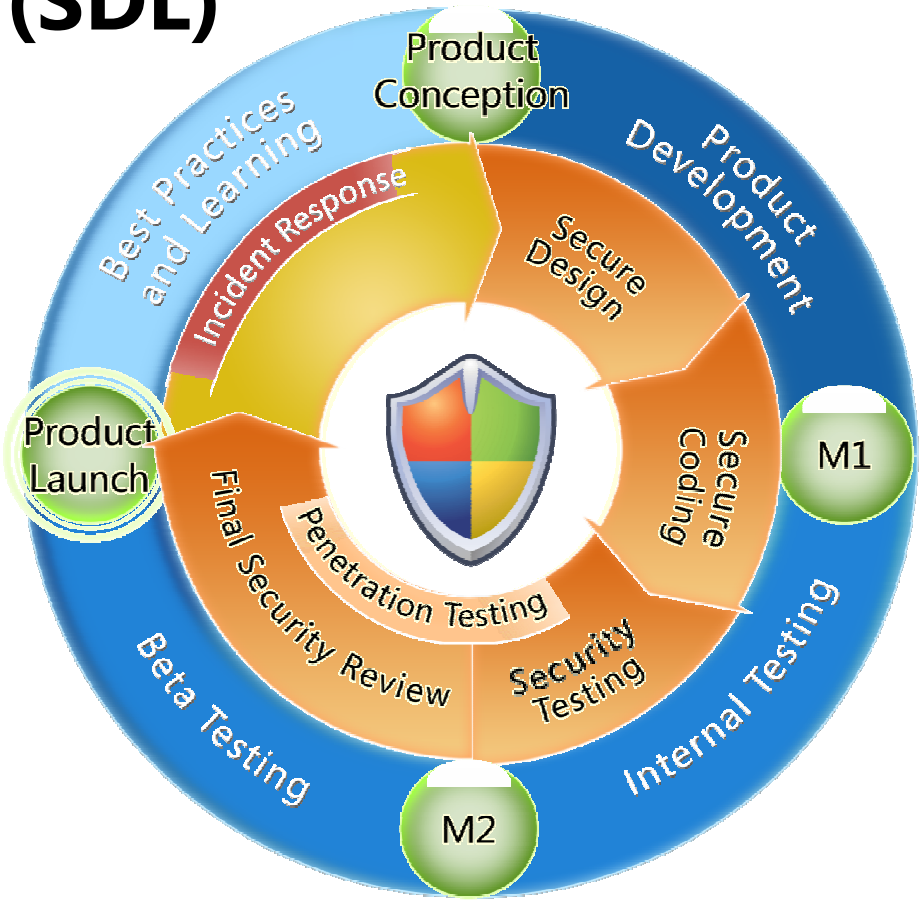
- Be created for all functionality identified during the cost analysis phase (stage 2) of the Microsoft SDL
- Meet minimal quality requirements (refer to chapter 9 of the Microsoft SDL book, http://www.microsoft.com/learning/en/us/books/8753.aspx)
- Be reviewed and approved by at least one developer, one tester and one program manager
- Be stored using document control systems

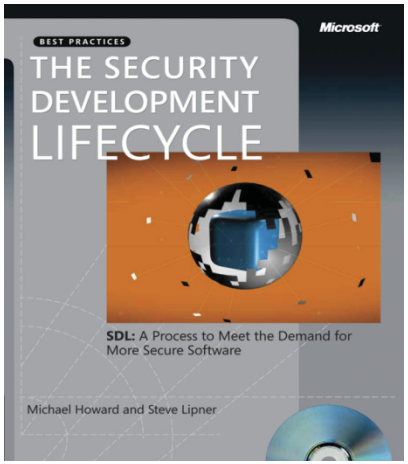
Conclusion

- Overview of the Microsoft SDL Threat Modeling process
- Advantages and disadvantages
- Steps of the Microsoft SDL Threat Modeling process
- Microsoft Threat Modeling Tool
- Microsoft SDL threat modeling requirements

Appendix

Microsoft Security Development Lifecycle (SDL)

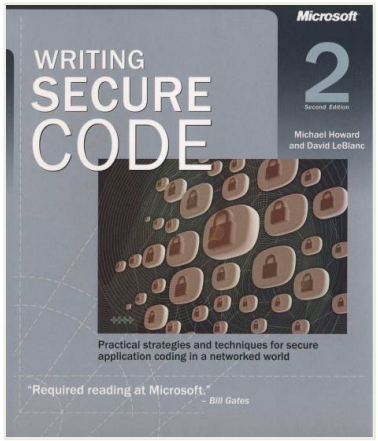




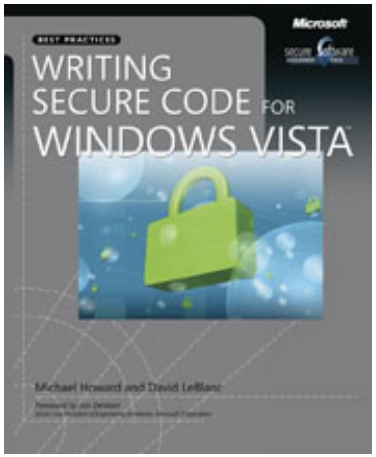
SDL Book:
http://www.microsoft.com/mspress/books/8753.aspx

Official SDL Web Site: http://www.microsoft.com/sdl

Microsoft Writing Secure Code Book Series



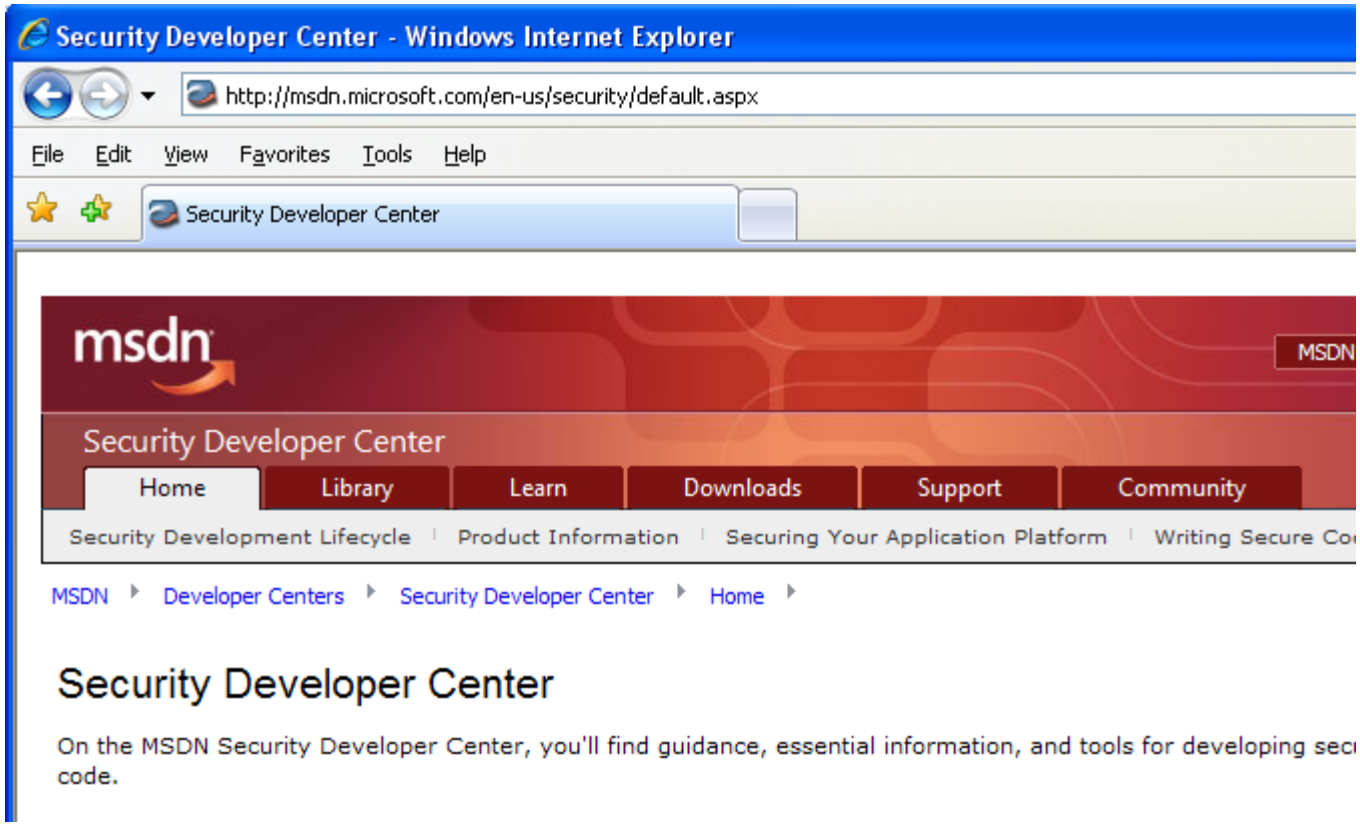
Writing Secure Code, 2nd Edition:
http://www.microsoft.com/mspress/books/5957.aspx



Writing Secure Code for Windows Vista:
http://www.microsoft.com/mspress/books/10723.aspx

Microsoft Developer Network (MSDN) Security Developer Center

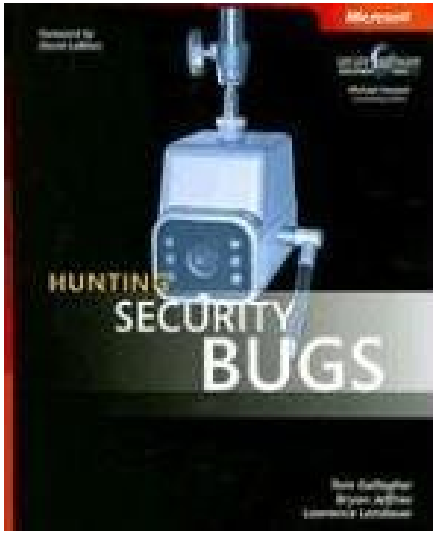
- Official Web site: http://msdn.microsoft.com/security



Secure Development Blogs

- The Microsoft Security Development Lifecycle (SDL) Blog: http://blogs.msdn.com/sdl
- Michael Howard's Blog: http://blogs.msdn.com/michael_howard

Microsoft Hunting Security Bugs



Hunting Security Bugs:

http://www.microsoft.com/mspress/books/8485.aspx

Additional SDL Training Content

- Secure Design Principles
- Secure Implementation Principles
- Secure Verification Principles
- SQL Injection Vulnerabilities
- Cross-Site Scripting Vulnerabilities
- Buffer Overflow Vulnerabilities