THE UNIVERSITY OF WINNIPEG

ACS-2821-001

Information Security in Business

Software Development Security

DISCOVER • ACHIEVE • BELONG

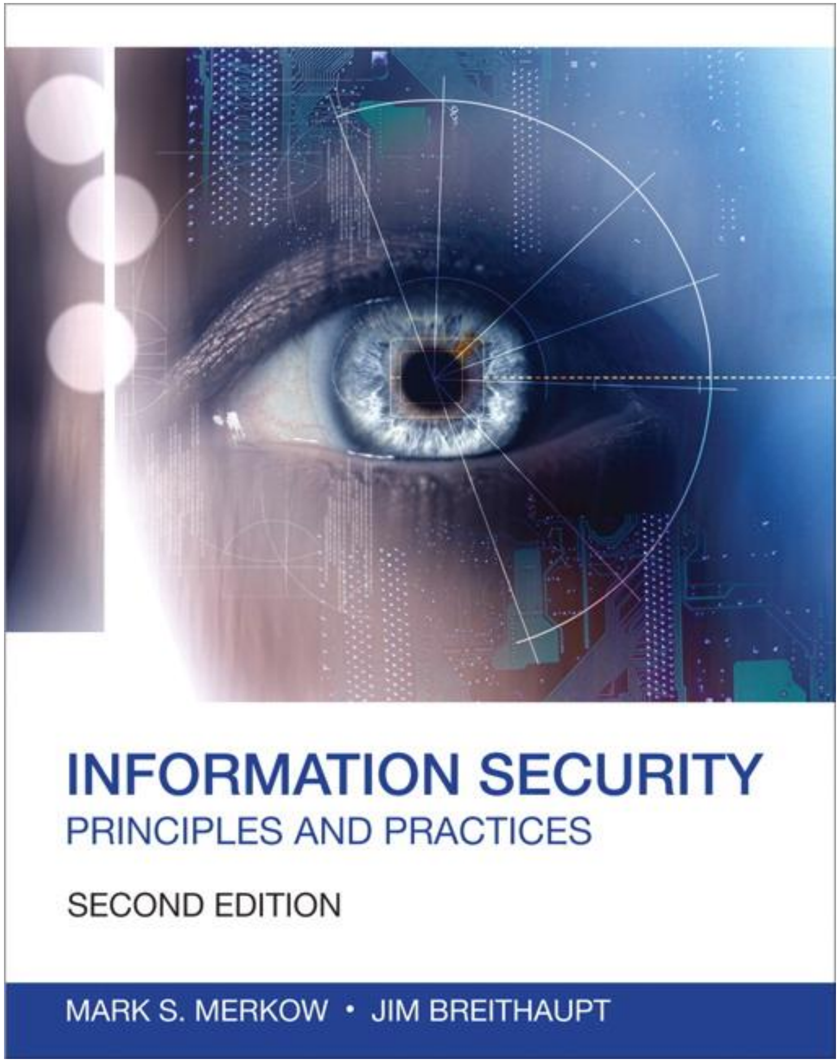
ACS-2821-001 – Slides Used In The Course

A note on the use of these slides:

These slides has been adopted and/or modified from the original for the use in this course. The author of the text have make these slides available to all (faculty, students, readers) and they obviously represent a *lot* of work on their part.

In return for use, please:

- If slides are being used (e.g., in a class) that the source be mentioned (after all, the author like people to use our book!)
- If any slides are being posted on a www site, note that they are adapted from (or perhaps identical to) the author original slides, and note their copyright of this material.



INFORMATION SECURITY
PRINCIPLES AND PRACTICES
SECOND EDITION
MARK S. MERKOW • JIM BREITHAUPT

Objectives

- Describe the importance of security activities throughout the system development life cycle (SDLC) to implement secure systems
- Describe the tasks and activities within each phase of the SDLC needed for an overall secure software program
- Understand the major industry models for measuring the maturity of a secure software development program

Overview

- Lessons
  - “The best offense is a good defense”
  - “An ounce of prevention is worth a pound of cure”

- As the software development process continues to mature, software designers build more safeguards into their applications
  - To prevent intrusion attacks instead of relying on security administrators to react to attacks after they occur

The Practice of Software Engineering

- In the early days of software development, software security was little more than a system ID, a password, and a set of rules determining the data access rights of users on the machine
- There is a need to discuss the risks inherent in making software systems available to a theoretically unlimited and largely anonymous audience
- Security in software is no longer an “add-on” but a requirement that software engineers must address during each phase of the SDLC
- Software engineers must build defensive mechanisms into their computer systems to anticipate, monitor, and prevent attacks on their software systems

Software Development Life Cycle

- Fundamental tasks
  - Understand the requirements of the system
  - Analyze the requirements in detail
  - Determine the appropriate technology for the system based on its purpose and use
  - Identify and design program functions
  - Code the programs
  - Test the programs, individually and collectively
  - Install the system into a secure “production” environment

Software Development Life Cycle

- Models
  - Simple SDLC
  - Waterfall model
  - Scrum Model
  - Agile Model

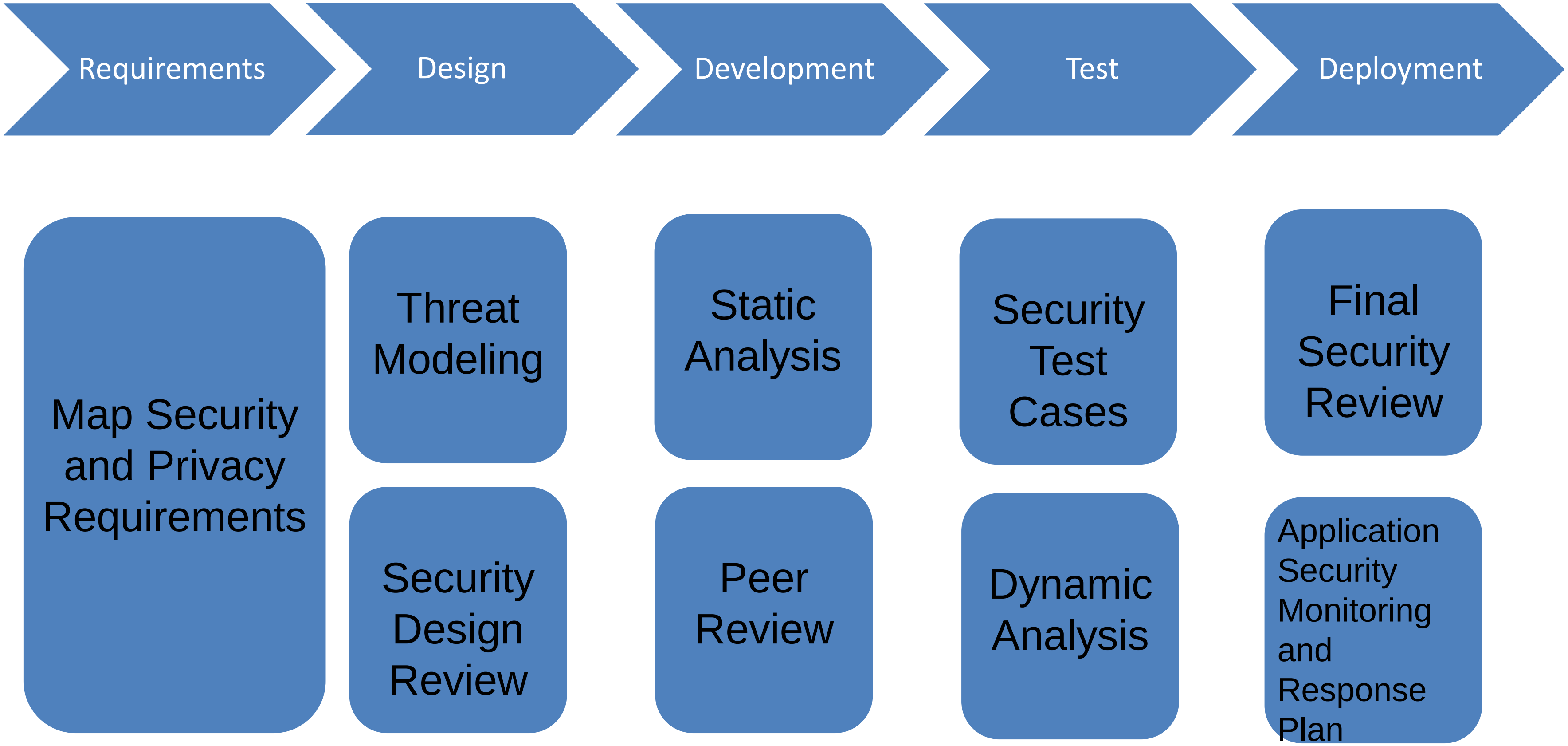
Software Development Life Cycle

- General approaches to software design
  - Emphasize the data (Data Centric)
    - The data model takes precedence over all else, for example, data flow diagramming
  - Emphasize the user’s interaction with the system (User Centric)
    - Rational unified process/use cases
  - Regardless of the approach used security should be considered

Software Development Life Cycle

- Phases of SDLC
  - Phase zero (project inception)
  - System requirements
  - System design
  - Development
  - Test
  - Deployment
- To make software secure, security must be built into the development life cycle
- The earlier in the development life cycle security is implemented, the cheaper software development will be

Software Development Life Cycle



Requirements

Map Security and Privacy Requirements

Design

Threat Modeling

Security Design Review

Development

Static Analysis

Peer Review

Test

Security Test Cases

Dynamic Analysis

Deployment

Final Security Review

Application Security Monitoring and Response Plan

Requirements Gathering and Analysis

- The first step in the SDLC
- Key activities
  - Map out and document non-functional requirements (NRFs)
  - Map security and privacy requirements
- Business system analysis should be familiar with
  - Organizational security policies and standards
  - Organizational privacy policy
  - Regulatory requirement (HIPAA, Sarbanes-Oxley)
  - Relevant industry standards (PCI DSS, ANSI-X9)

System Design and Detailed Design

- Major processes during the design phase
  - Threat modeling
    - Used to determine the technical security posture of the application being developed
    - Four key steps
      - Functional decomposition
      - Categorizing threats
      - Ranking threats
      - Mitigation planning
  - Design reviews
    - Carried out by a security subject matter expert
    - Typically iterative in nature

Development (Coding) Phase

- The activities within this phase generate implementation-related vulnerabilities
- Key processes
  - Static analysis
    - Uses automated tools to find issues with source code
  - Peer review
    - Developers review each others code and provide feedback
    - Time consuming
  - Unit testing
    - Helps prevent bugs and flaws from reaching the testing phase

Testing

- Critical step for discovering vulnerabilities not found earlier
- Steps
  - Built security test cases
  - Tests are used during dynamic analysis
  - Software is loaded and operated in a test environment

Deployment

- The final phase of SDLC
- Software is installed and configured in production environment
- Key activities
  - Final security review
  - Creating application security monitoring and response plan
- Security training is a prerequisite for anyone involved in the software development

Measuring the Secure Development Program

- Two software security maturity measurement models
  - Open Software Assurance Maturity Model (OpenSAMM)
  - Building Security in Maturity Model (BSIMM)

Measuring the Secure Development Program

Open Software Assurance Maturity Model (OpenSAMM)

- Evaluating an organization’s existing software security practices
- Building a balanced software security assurance program in well-defined iterations
- Demonstrating concrete improvements to a security assurance program
- Defining and measuring security-related activities throughout an organization
- For more information - https://opensamm.org

Measuring the Secure Development Program

Building Security in Maturity Model (BSIMM)

- Has four domains
  - Governance - help organize, manage, and measure a software security initiative
  - Intelligence - Collections of corporate knowledge used in carrying out software security activities throughout the organization.
  - SSDL Touchpoints - Practices associated with analysis and assurance of particular software development artifacts and processes.
  - Deployment - Practices that interface with traditional network security and software maintenance organizations
  - For more information - https://www.bsimm.com

Summary

- It’s essential that security is built into all phases of the SDLC
- The SDLC consists of the following five phases: colleting requirements, design, development, testing, and deployment
- Secure applications do not come by accident but through careful planning and deliberate actions to incorporate security in all stages of the SDLC

ACS-2821-001 – Questions?

QUESTIONS

now