Software Security Engineering
Lecture 11

Nancy R. Mead, SEI
nrm@sei.cmu.edu

Topics

- Implementation strategy
- Course summary

Implementation Strategy

Areas of Practice (prioritized within each area)

- SSE practices that span the SDLC
- Requirements engineering practices
- Architecture and design practices
- Coding and testing practices
- Security analysis for system complexity and scale: mitigations
- Governance and management practices

SSE Practices That Span the SDLC

- Properties of secure software
- Attack patterns
- Assurance case

Requirements Engineering Practices

- Standard security requirements engineering process
- Security risk assessment
- Threat identification
- Security requirements elicitation
- Security requirements categorization and prioritization
- Security requirements inspection

Architecture and Design Practices

- Security principles
- Attack patterns
- Architectural risk analysis
- Security guidelines

Coding and Testing Practices

- Secure coding practices
- Source code review for security vulnerabilities
- Unique aspects of software security testing
- Functional test cases for security
- Risk-based test cases for security
- Test cases using a range of security test strategies

Security Analysis for System Complexity and Scale: Mitigations

- Tackle known interface vulnerabilities first
- Conduct end-to-end analysis of cross-system work processes
- Attend to containing and recovering from failures
- Explore failure analysis and mitigation to deal with complexity
- Coordinate security efforts across organizational groups

Governance and Management Practices

- Risk-based definition of adequate security
- Continuous risk management framework
- Software security practices integrated with SDLC
- Software security as a cultural norm
- Characteristics of software security at the governance/management level
- Enterprise software security risk framework
- Software security included in software measurement process

Course Summary

Course Topics – What We Said We Would Cover

- Security models and methods in the areas of:
  - lifecycle process models
  - risk management
  - requirements engineering
  - architecture and design
  - coding and testing
  - governance and management
- If time permits, acquisition of newly developed and COTS software will also be discussed.

What We Covered

Lecture 1

About this course

Software assurance challenges

Foundations for software assurance

Software assurance guiding principles

Lecture 2

Software assurance practices

Software assurance lifecycle models

Software assurance maturity models

Lecture 3

- An Assurance Ecosystem (carried over from Lecture 2)
- Requirements Engineering
- Introduction to SQUARE

Lecture 4

Background

- The Need for SQUARE
- Recap of the SQUARE process
- Three Cases for Square for Acquisition (A-SQUARE)
  - A. introduction
  - B. workflow
  - C. important points
- Conclusion and further work

Lecture 5

- What does mission failure look like?
  - Example: 2003 Power Grid failure
- Overview of Mission Thread Analysis
- Examples using Mission Thread Analysis
- Experience to-date

Lecture 6

- Industry Case Study in Threat Modeling
- Introduction to Threat Modeling
- Use of Threat Modeling in Prioritization of Security Requirements
- Conclusion

Lecture 7

- Risk Management Overview
- Two Approaches for Analyzing Risk
- Mission Risk Diagnostic (MRD)
- Standard Driver Sets
- Risk-Based Measurement and Analysis
- Summary

Lecture 8

- How to Threat Model
- The STRIDE per Element Approach to Threat Modeling
- Diagram Validation Rules of Thumb
- Exercise
- Demo Video

Lecture 9

- Attack surface
- Measurement
- Inspecting for security

Lecture 10

Secure Coding - Strings

- Common errors using NTBS
- Common errors using **`basic_string`**
- String Vulnerabilities
- Mitigation Strategies
- Summary

Some Areas That We Didn't Cover or Just Touched On

Development

- Assurance cases
- Architectural Risk Analysis
- Coding practices
- Security Testing

Operational and Analytical Considerations

- System administration
- Recognition and response
- Thinking like an attacker
- Analysis/forensics
- Legal/policy issues

Acquisition

- Only touched on selected topics

Reference

- Software Security Engineering book, Chapter 8

Comments and Suggestions?

NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.