

SE4930 Secure Software Development

Dr. Walter Schilling

Winter, 2012-2013

For the test, you may bring one 8.5x 11 sheet of paper with, review midterm it. The exam will consist of took multiple choice, short answer, and small problems.

1. Week 1

(a) Lecture 1 Introduction to software security

- i. Comprehend the magnitude of the security problem.
- ii. Compare and contrast bugs and flaws
- iii. Compare and contrast secure software development, network security, and data security
- iv. Explain the trinity of trouble
- v. Justify why more code leads to more security bugs

(b) Lecture 2

- i. Define Confidentiality, Integrity, and Availability
- ii. Define authentication, authorization, and accountability
- iii. Explain the concept of a risk management framework
- iv. Explain the how the scope of a business may impact the security concerns.
- v. Explain the difference between bugs and flaws.

2. Week 2

(a) Lecture 1 Software Security Touchpoints

- i. Define and explain the relationship between Single Loss Expectancy, Annual Rate of Occurrence, and Annual Loss Expectancy.
- ii. Explain the concept of a Touchpoint
- iii. List the Software Security Touchpoints
- iv. Identify the most effective security practices to have within software development
- v. Explain the basic premise for CLASP
- vi. List the 7 best practices for security according to CLASP
- vii. Explain the difference between the reactive and proactive approaches to software security.
- viii. Recognize an example of a bug and a flaw within a software system
- ix. Explain the economic impact of various security activities.

(b) Lecture 2 Requirements Part 1

- i. Differentiate between security goals and security functions
- ii. Explain the concept of a security requirement
- iii. List the three characteristics of secure software
- iv. Explain the concept of a security profile
- v. Explain confidentiality requirements
- vi. Explain integrity requirements
- vii. Explain authentication requirements
- viii. Compare and contrast simple authentication, two factor authentication, and multifactor authentication.

3. Week 3

(a) Lecture 1 Threat Modeling

- i. Define the concept of an asset.

- ii. Define the concept of a threat.
- iii. Define the term attack vector.
- iv. Critique an architecture for potential risks.
- v. Explain the concepts of risk mitigation.

(b) Lecture 2 Abuse Cases

- i. Differentiate between security goals and security functions.
- ii. Explain the concept of a security requirement
- iii. Compare and contrast
- iv. Differentiate between security goals and security functions
- v. Explain the concept of a security requirement
- vi. Compare and contrast requirements and anti-requirements
- vii. Explain the concept of an abuse case and explain how one would create an abuse case
- viii. Explain what you need to do to think securely.
- ix. Explain the concept of a use case diagram
- x. Explain the concept of abuse cases as shown on a use case diagram
- xi. Construct examples of abuse cases

4. Week 4

(a) Lecture 1 Design Principles

- i. Explain why data and code commingling can cause problems
- ii. Compare and contrast asymmetric and symmetric cryptography design.
- iii. Explain the concept of a certificate.
- iv. Explain the usage of a hash code to verify the integrity of data.
- v. Compare and contrast a salted and unsalted hash key.

(b) Lecture 2 Design Principles

- i. Explain the design for an appropriate mechanism for encrypting and storing passwords.
- ii. List and explain the secure design principles.
 - A. Principle of Least Privilege
 - B. Separation of Duties
 - C. Defense in Depth
 - D. Fail Secure
 - E. Economy of Mechanisms
 - F. Complete Mediation
 - G. Open design
 - H. Least Common Mechanisms
 - I. Psychological Acceptability
 - J. Leveraging Existing Components
- iii. Describe the concept of trust domains and trust boundaries.
- iv. Critique architectures based on trust allocations.
- v. Critique a modern software application from a security standpoint.

5. Week 5

(a) Lecture 1

- i. Explain the concept of a trust relationship
- ii. Define the concept of a trust boundary
- iii. Given an architecture, construct a diagram showing trust boundaries and trust relationships.
- iv. Draw a data flow diagram, showing how information moves within a system.
- v. Explain the concept of threat modeling
- vi. Define associated terms related to threat modeling

(b) Lecture 2 Architecture

- i. Explain the relationship between client server and peer to peer architectures and the security risks associated with each system.

- ii. Explain the concept of a thin client.
- iii. Explain the concept of a fat client.
- iv. Define an n tier software architecture.
- v. Explain the problems with a 1 tier architecture.

6. Week 6

(a) Lecture 1 A Taxonomy of Coding Errors

- i. Recognize through code review simple programming mistakes.
- ii. Define the concept of a taxonomy
- iii. Explain the relationship between kingdoms and phyla
- iv. Explain the kingdoms for security vulnerabilities
 - A. Input Validation and Representation
 - B. API Abuse
 - C. Security Features
 - D. Time and State
 - E. Errors
 - F. Code Quality
 - G. Encapsulation
 - H. Environment
- v. Critique source code for examples of coding errors
- vi. Explain how simple coding errors might be exploited by an adversary

(b) Lecture 2 Midterm Exam