

PHÁT TRIỂN PHẦN MỀM AN TOÀN (SECURE SOFTWARE DEVELOPMENT)

Đề cương chi tiết (Course Syllabus)

1. General Information

Course name: Phát triển phần mềm an toàn (Secure Software Development)
Course code: SEC1418_CLC
Course type: Selective
Number of credits: 2

2. Objectives

Knowledge:

This course provides basic knowledge on collecting security requirements in software development. Also, the course introduces design and development processes to meet security requirements, and integrating secure software development (SSD) processes with conventional software development life cycle (SDLC) processes.

Skills:

On successful completion of this course a student will be able to:

- Identify and prioritize software security requirements.
- Apply appropriate security tools and techniques to different phases of security enhanced software lifecycle models.

Attitude:

Students are required to attend the classes and complete assignments/projects.

3. Abstracts

The course is to educate learners on the fundamentals of the secure software lifecycle, enabling them to apply principles of secure software development. The course also includes practical applications and tools, allowing learners to experience the secure software lifecycle process by developing tangible artifacts.

4. Teaching and learning methods

Lectures: 20h
Exercises: 4h
Projects: 4h
Labs: 2h
Individual reading: 0h

5. Prerequisites: Fundamentals of Information Security - SEC1401_CLC

6. Learning outcomes

After completing this course, the student is able to:

[CLO1] Identify and prioritize security requirements in design and development of computer software.

[CLO2] Describe and apply appropriate security techniques and tools to different phases of security enhanced software lifecycle models.

7. Assignment criteria

Learning outcomes	Assignment criteria
[CLO1] Identify and prioritize security requirements in design and development of computer software.	Chapter 1, Chapter 2, Chapter 3
[CLO2] Describe and apply appropriate security techniques and tools to different phases of security enhanced software lifecycle models.	Chapter 2, Chapter 3, Chapter 4,

8. Outlines

Chapter 1 Introduction

- 1.1. Software vulnerabilities
- 1.2. Software assurance definition
- 1.3. Properties of secure software

Chapter 2 Security in design

- 2.1. Software security requirements
- 2.2. Threat models
- 2.3. SDLC models
- 2.4. Maturity models

Chapter 3 Security in development

- 3.1. Principles of secure coding
- 3.2. Static analysis
- 3.3. Defensive coding
- 3.4. Secure coding standards
- 3.5. Secure frameworks for web and mobile development

Chapter 4 Software security testing and integration

- 4.1. Types of testing
- 4.2. SecOps
- 4.3. Continuous integration
- 4.4. Challenges and Future Trends

9. Required Textbooks

- [1] Loren Kohnfelder, Designing Secure Software: A Guide for Developers, No Starch Press, Dec 21, 2021, 312 pages

10. Suggested Textbooks

- [2] Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, and Nancy R. Mead, Software Security Engineering: A Guide for Project Managers, Addison Wesley Publishers, 2008
- [3] Secure Software Development Framework (SSDF), <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf>

11. Schedule

Main contents	Duration	Specific contents
Chapter 1 Introduction	3h lecture	<ul style="list-style-type: none">– Software vulnerabilities– Software assurance definition– Properties of secure software
Chapter 2 Security in design	6h lecture 1h exercise	<ul style="list-style-type: none">– Software security requirements– Threat models– SDLC models– Maturity models
Chapter 3 Security in development	6h lecture 1h exercise 2h project 2h lab	<ul style="list-style-type: none">– Principles of secure coding– Static analysis– Defensive coding– Secure coding standards– Secure frameworks for web and mobile development
Chapter 4 Software security testing and intergration	5h lecture 2h project	<ul style="list-style-type: none">– Types of testing– SecOps– Continuous integration– Challenges and Future Trends

12. Grading Policy

Attendance:	10%
Mid-term exam/exercises:	10%
Course projects:	30%
Final examination:	50%