Uka Tarsadia University

B. Tech.

CSE / CSE (CC) / CE (SE)

Semester VII

Program Elective - V
SOFTWARE DEVELOPMENT SECURITY
CE6013

EFFECTIVE FROM July-2024

Syllabus version: 1.00

Subject Code	Subject Title
CE6013	**Software Development Security**

Teaching Scheme				Examination Scheme				
Hours		**Credits**		**Theory Marks**		**Practical Marks**		**Total Marks**
Theory	Practical	Theory	Practical	Internal	External	Internal	External	
3	2	3	1	40	60	20	30	150

Objectives of the course:

- To understand the software security, encompassing assurance, threats, insecurities, and the importance of early defect detection.
- To learn security assurance effectively within software development processes to establish and uphold software security standards.

Course outcomes:

Upon completion of the course, the student shall be able to,

CO1: Understand security's software relevance, exploring assurance, threats, insecurities, and early defect benefits.
CO2: Define software security by integrating assurance into development.
CO3: Understand requirements, quality, security, SQUARE model, and elicitation for secure software.
CO4: Design and code securely by embracing architectural risk analysis, coding best practices, and security testing.
CO5: Analyze failures, attacker behavior, and system complexity.
CO6: Learn about governance, security project management, resilience, legal aspects, and security maturity in software engineering.

Sr. No.	Topics	Hours
	Unit – I	
1	**Why is Security a Software Issue?** Why is security a software issue? – Introduction, The problem, Software assurance and software security, Threats to software security, Sources of software insecurity, The benefits of detecting software security defects early, Managing secure software development.	5
	Unit – II	
2	**What Makes Software Secure?** Defining properties of secure software, How to influence the security properties of software? How to assert and specify desired security	8

	properties? – Building a security assurance case, A security assurance case example, Incorporating assurance cases into the SDLC, Related security assurance and compliance efforts, Maintaining and benefiting from assurance cases.	
Unit – III		
3	**Requirements Engineering for Secure Software:** The importance of requirements engineering, Quality requirements, Security requirements engineering, Misuse and abuse cases, The SQUARE process model, SQUARE sample outputs, Requirements elicitation, and Requirements prioritization. **Secure Software Architecture and Design:** The critical role of architecture and design, Issues and challenges, Software security practices for architecture and design: Architectural risk analysis, Software security knowledge for architecture and design: Security principles, Security guidelines, and Attack patterns.	10
Unit – IV		
4	**Considerations for Secure Coding and Testing:** Code analysis – Common software code vulnerabilities, Source code review, Coding practices, Software security testing – Contrasting software testing and software security testing, Functional testing, Risk-based testing, Security testing considerations throughout the SDLC – Unit testing, Testing libraries and executable files, Integration testing, System testing, Sources of additional information on software security testing.	6
Unit – V		
5	**Security and Complexity – System Assembly Challenges:** Introduction, Security failures – Categories of errors, Attacker behavior, Functional and attacker perspectives for security analysis: Two examples, System complexity drivers and security, Deep technical problem complexity.	8
Unit – VI		
6	**Governance and Managing for More Secure Software:** Governance and security, Adopting an enterprise software security framework, How much security is enough?, Security and project management, Maturity of practice – Protecting information, Audit's role, Operational resilience and convergence, A legal view, A software engineering view, Exemplars.	8

Sr. No.	Software Development Security(Practicals)	Hours
1	Identify common threats to software security, study sources of software insecurity and potential risks.	4
2	Study eliciting security requirements for software projects, prioritize security requirements and their importance.	2
3	Perform a risk analysis to identify security vulnerabilities in software architecture and develop strategies to mitigate identified risks.	4
4	Implement secure coding practices to address common software vulnerabilities, review code and apply security guidelines and principles.	4
5	Set up a lab environment for security testing, including code review and vulnerability scanning and perform security testing on sample applications and analyze results.	4
6	Analyze security failures and attacker behavior in complex systems and study strategies for managing security in the face of system complexity.	4
7	Study governance and security management and explore the role of governance in ensuring software security and compliance.	4
8	Conduct a security maturity assessment for a software development organization, Identify areas for improvement and develop a plan to enhance security maturity.	4

Text book:

1. Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw and Nancy R. Mead, “Software Security Engineering: A Guide for Project Managers”, Addison-Wesley Professional.

Reference books:

1. Jason Grembi, “Secure Software Development: A Security Programmer's Guide”, Cengage Learning.
2. Gray McGraw, “Software Security – Building Security In”, Addison Wesley.
3. James Ransome and Anmol Misra, “Core Software Security”, Auerbach Publications.

Course objectives and Course outcomes mapping:

- To understand the software security, encompassing assurance, threats, insecurities, and the importance of early defect detection: CO1, CO2, CO3.
- To learn security assurance effectively within software development processes to establish and uphold software security standards: CO4, CO5, and CO6.

Course units and Course outcomes mapping:

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Why is Security a Software Issue?	✓					

2	What Makes Software Secure?		✓				
3	Requirements Engineering for Secure Software, and Secure Software Architecture and Design			✓			
4	Considerations for Secure Coding and Testing				✓		
5	Security and Complexity – System Assembly Challenges					✓	
6	Governance, and Managing for More Secure Software						✓

Programme outcomes:

PO 1: Engineering knowledge: An ability to apply knowledge of mathematics, science, and engineering.

PO 2: Problem analysis: An ability to identify, formulates, and solves engineering problems.

PO 3: Design/development of solutions: An ability to design a system, component, or process to meet desired needs within realistic constraints.

PO 4: Conduct investigations of complex problems: An ability to use the techniques, skills, and modern engineering tools necessary for solving engineering problems.

PO 5: Modern tool usage: The broad education and understanding of new engineering techniques necessary to solve engineering problems.

PO 6: The engineer and society: Achieve professional success with an understanding and appreciation of ethical behavior, social responsibility, and diversity, both as individuals and in team environments.

PO 7: Environment and sustainability: Articulate a comprehensive world view that integrates diverse approaches to sustainability.

PO 8: Ethics: Identify and demonstrate knowledge of ethical values in non-classroom activities, such as service learning, internships, and field work.

PO 9: Individual and team work: An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give/receive clear instructions.

PO 11: Project management and finance: An ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: A recognition of the need for, and an ability to engage in life-long learning.

Programme outcomes and Course outcomes mapping:

Programme Outcomes	Course Outcomes					
	CO1	CO2	CO3	CO4	CO5	CO6
PO1	✓					
PO2				✓	✓	
PO3		✓				
PO4				✓		
PO5					✓	✓
PO6						
PO7						
PO8	✓					
PO9			✓			✓
PO10	✓					
PO11						✓
PO12						