SECURE SOFTWARE ENGINEERING

Subject Code: CS34323CS	Subject Type: Elective
Evaluation Scheme: MSE(30), TA(20), ESE(50)	L-T-P: 3-0-0

Pre-Requisites: Software Engineering, Information Security, and Programming.

Course Objective: To explore the depths of various digital incident domains, gain practical skills in detecting cyberattacks and incidents using commercial, open-source, and freeware tools, and develop case-oriented procedures for emerging scenarios.

Course Outcomes: At the end of the course, the student will be able to

CO-1	Understand secure software engineering principles and SDLC models
CO-2	Identify and analyze software vulnerabilities and threats.
CO-3	Apply secure coding and testing techniques to develop secure applications.
CO-4	Evaluate tools and methodologies for secure software deployment and maintenance.
CO-5	Interpret security policies, compliance, and ethical issues in software development.

Course Articulation Matrix:

CO/PO	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6
CO-1	2	3	2	2	3	-
CO-2	3	3	3	-	3	1
CO-3	3	2	3	1	3	2
CO-4	1	3	3	2	3	1
CO-5	3	3	3	3	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

Syllabus:

Unit-1: Foundations of Secure Software Engineering

Introduction to Secure Software Engineering, Software security vs information security, Secure SDLC models (Waterfall, Agile, DevSecOps), Security principles: Least privilege, defense in depth, fail-safe defaults, Threat modeling basics (STRIDE, attack surface analysis)

Unit-2: Software Vulnerabilities & Secure Design

Common vulnerabilities (buffer overflow, injection, XSS, CSRF), OWASP Top 10 vulnerabilities, Secure design principles and patterns, Input validation and error handling Authentication and authorization mechanisms

Unit-3: Secure Coding & Application Security Testing

Secure coding practices (C, Java, Python), Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), Software composition analysis (SCA), Code review techniques and tools

Unit-4: Secure Deployment, DevSecOps & Compliance

Secure deployment practices (CI/CD security), DevSecOps concepts and pipeline security, Vulnerability management and patching, Incident response in software systems, Legal, ethical, and compliance issues

Learning Resources:

Text and Reference Books:

1. Seacord, Robert C. Secure Coding in C and C++. Addison-Wesley, 2013.
2. Dowd, Mark, John McDonald, and Justin Schuh. The art of software security assessment: Identifying and preventing software vulnerabilities. Pearson Education, 2006.
3. Adkins, Heather, et al. Building secure and reliable systems: best practices for designing, implementing, and maintaining systems. " O'Reilly Media, Inc.", 2020.McGraw, Gary. "Software security: Building security in." Datenschutz und Datensicherheit-DuD 36.9 (2012): 662-665.