SARVAJANIK UNIVERSITY
INCLUSIVE | INTEGRATED | INNOVATIVE

SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Bachelor of Technology

Year: B. Tech IV (Semester VII)

Subject Name: Secure Software Engineering **Subject Code:** BTIT14707
Type of course: Professional Elective Course
Prerequisite (if any): -

Rationale: From social interaction, education, and communication to business, transportation, and government and everything in between, society depends on complex software systems. The consequences of a failure in a software system can be severe, and reliable functionality and security are critical. This course provides a foundation for building secure software by applying security principles to the software development lifecycle.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	
3	0	0	3	60	25	15	-	-	100

CA1: Continuous Assessment (assignments / projects / open book tests / closed book tests) CA2: Sincerity in attending classes / class tests / timely submissions of assignments / self-learning attitude / solving advanced problems TEE: Term End Examination TEP: Term End Practical Exam (Performance and viva on practical skills learned in course) CA3: Regular submission of Lab work / Quality of work submitted / Active participation in lab sessions / viva on practical skills learned in course.

Contents:

Sr. No.	Contents	Total Hours
1.	**Secure Software Life Cycle** Software Design, Software Implementation, Software Testing, Continuous Updates and Patches, Modern Software Engineering	04
2.	**Memory and Type Safety** Pointer Capabilities, Memory Safety, Spatial Memory Safety, Temporal Memory Safety, A Definition of Memory Safety, Practical Memory Safety, Type Safety	08
3.	**Attack Vectors** Denial of Service (DoS), Information Leakage, Confused Deputy, Privilege Escalation, Control-Flow Hijacking , Code Injection , Code Reuse	06
4.	**Buffer Overflow** Stack Overflows, Defending Against Buffer Overflows, Other Forms of Overflow	04

	Attacks	
5.	**Software Security** Software Security Issues, Handling Program Input, Writing Safe Program Code, Interacting with the Operating System and Other Programs, Handling Program Output	06
6.	**Operating System Security** Introduction to Operating System Security, System Security Planning, Operating Systems Hardening, Application Security, Security Maintenance, Linux/Unix Security, Windows Security, Virtualization Security	06
7.	**Case Studies** Web security, Protecting long running services, Browser security, Command injection, SQL injection, Cross Site Scripting (XSS), Cross Site Request Forgery (XSRF), Mobile security, Android system security, Android market, Permission model	05
8.	**Secure Programming OWASP Top 10 Proactive Controls** C1: Define Security Requirements, C2: Leverage Security Frameworks and Libraries C3: Secure Database Access, C4: Encode and Escape Data, C5: Validate All Inputs C6: Implement Digital Identity, C7: Enforce Access Controls, C8: Protect Data Everywhere, C9: Implement Security Logging and Monitoring, C10: Handle All Errors and Exceptions	06

Suggested Specification table with Marks (Theory): (For B. Tech only)

Distribution of Theory Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
20%	40%	30%	10%	-	-

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate C: Create (Revised Bloom's Taxonomy)

Reference Books:

Sr no	**Title of book /article**	**Author(s)**	**Publisher and details like ISBN**	**Year of publication**	**Publication Edition**
1	Software Security Principles, Policies, and Protection	Mathias Payer		July 2021	
2	Computer Security:	William	Pearson	2012	2nd Edition

	Principles and Practice	Stallings			
3	Software Engineering – A Practitioner's approach	Roger S. Pressman	Tata McGraw Hill	2009	
4	Software Security: Building Security	Gary McGraw	Addison Wesley Software Security Series	2006	2nd Edition
5	Secure Software Design	Theodor Richardson, Charles Thies	Jones and Bartlet Learning	2013	2nd Edition

Course Outcomes (CO):

Sr. No.	CO statements	Marks % weightage
CO-1	Understanding of fundamental concepts in secure software life cycle.	10%
CO-2	Describe and discuss type safety and various attacks.	30%
CO-3	Understating the impact of Software security including buffer overflow and operating system security.	40%
CO-4	Analyse secure coding practices to prevent and mitigate vulnerabilities such as injection attacks, cross-site scripting (XSS), and android system security.	10%
CO-5	Apply Secure Programming Proactive Controls in secure software development lifecycles (SSDLC).	10%

List of Open learning website:

- OWASP Proactive Controls available at: https://owasp.org/www-project-proactive-controls/