ISC2 Certified Secure Software Lifecycle Professional Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

The Official (ISC)²® Certified Secure Software Lifecycle Professional (CSSLP®) training provides a comprehensive review of the knowledge required to incorporate security practices, including authentication, authorization and auditing, into each phase of the Software Development Lifecycle (SDLC), from software design and implementation to testing and deployment. This training course will help learners review and refresh their knowledge and identify areas they need to study for the CSSLP exam. Content aligns with and comprehensively covers the eight domains of the (ISC)² CSSLP Common Body of Knowledge (CBK®).

As an (ISC)2 Official Training Provider, we use courseware developed by (ISC) creator of the CSSLP CBK to ensure your training is relevant and up-to-date. Our instructors are verified security experts who hold the CSSLP and have completed intensive training to teach (ISC)² content.

Please Note: An Exam voucher is included with this course.

Prerequisites

The CSSLP is ideal for those working in roles such as:

- Software Architect
- Software Engineer
- Software Developer
- Application Security Specialist/Manager/Architect
- Software Program Manager
- Quality Assurance Tester
- Penetration Tester/Testing Manager
- Software Procurement Analyst
- Project Manager
- Security Manager

Candidates must have a minimum of 4 years' cumulative work experience in 1 or more of the 8 domains of the CSSLP Common Body of Knowledge (CBK), or 3 years of experience as an SDLC professional with a 4-year degree or regional equivalent in Computer Science, Information Technology or a related field.

A candidate who doesn't have the required experience may become an Associate of (ISC)² by successfully passing the CSSLP examination. The Associate of (ISC)² will then have 5 years to earn the 4 years of required experience.

ISC2 Certified Secure Software Lifecycle Professional Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

- Understand the core concepts of software security and the foundational principles that drive construction of resilient software.
- Recognize the importance of security requirements and understand the techniques for elicitation and specification of software security requirements.
- Recognize privacy requirements and their impact on the selection of safeguards and countermeasures.
- Understand threat modeling, attack surface evaluation, and architectural risk assessment.
- Recognize secure design principles and patterns.
- Understand secure coding practices, common application vulnerabilities and their mitigation strategies.
- Understand various code analysis techniques using automated and manual techniques.
- Recognize risks of third-party software components and libraries, malicious code and mitigation strategies.
- Describe security testing strategy and techniques and identify functional and non-functional testing methods.
- Describe defect tracking and risk scoring methods.
- Identify secure software methodologies, standards and frameworks.
- Understand Governance, Risk, and Compliance and recognize regulations and compliance requirements, Inc. NIST 800-218
- Describe risks during deployment and understand security relevant issues during the operations and maintenance phase of the lifecycle.
- Understand vulnerability management, security monitoring, incident response, and root cause analysis.
- Recognize software supply chain risks and attacks.

Outline

Domain 1: Secure Software Concepts

- Core Concepts
- Security Design Principles

Domain 2: Secure Software Requirements

- Define Software Security Requirements
- Identity and Analyze Compliance Requirements
- Identify and Analyze Data Classification Requirements
- Identify and Analyze Privacy Requirements
- Develop Misuse and Abuse Cases
- Develop Security Requirement Traceability Matrix (STRM)
- Ensure Security Requirements Flow Down to Suppliers/Providers

Domain 3: Secure Software Architecture and Design

- Define the Security Architecture
- Performing Secure Interface Design
- Performing Architectural Risk Assessment
- Model (Non-Functional) Security Properties and Constraints

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Model and Classify Data
- Evaluate and Select Reusable Secure Design
- Perform Security Architecture and Design Review
- Define Secure Operational Architecture (e.g., deployment topology, operational interfaces)
- Use Secure Architecture and Design Principles, Patterns, and Tools

Domain 4: Secure Software Implementation

- Adhere to Relevant Secure Coding Practices (e.g., standards, guidelines and regulations)
- Analyze Code for Security Risks
- Implement Security Controls (e.g., watchdogs, File Integrity Monitoring (FIM), anti-malware)
- Address Security Risks (e.g. remediation, mitigation, transfer, accept)
- Securely Reuse Third-Party Code or Libraries (e.g., Software Composition Analysis (SCA))
- Securely Integrate Components
- Apply Security During the Build Process

Domain 5: Secure Software Testing

- Develop Security Test Cases
- Develop Security Testing Strategy and Plan
- Verify and Validate Documentation (e.g., installation and setup instructions, error messages, user guides, release notes)
- Identify Undocumented Functionality
- Analyze Security Implications of Test Results (e.g., impact on product management, prioritization, break build criteria)
- Classify and Track Security Errors
- Secure Test Data
- Perform Verification and Validation Testing

Domain 6: Secure Lifecycle Management

- Secure Configuration and Version Control (e.g., hardware, software, documentation, interfaces, patching)
- Define Strategy and Roadmap
- Manage Security Within a Software Development Methodology
- Identify Security Standards and Frameworks
- Define and Develop Security Documentation
- Develop Security Metrics (e.g., defects per line of code, criticality level, average remediation time, complexity
- Decommission Software
- Report Security Status (e.g., reports, dashboards, feedback loops)
- Incorporate Integrated Risk Management (IRM)
- Promote Security Culture in Software Development
- Implement Continuous Improvement (e.g., retrospective, lessons learned)

Domain 7: Software Deployment, Operations and Maintenance

- Perform Operational Risk Analysis
- Release Software Securely
- Securely Store and Manage Security Data
- Ensure Secure Installation
- Perform Post-Deployment Security Testing
- Obtain Security Approval to Operate (e.g., risk acceptance, sign-off at appropriate level)

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Perform Information Security Continuous Monitoring (ISCM)
- Support Incident Response
- Perform Patch Management (e.g. secure release, testing)
- Perform Vulnerability Management (e.g., scanning, tracking, triaging)
- Runtime Protection (e.g., Runtime Application Self-Protection (RASP), Web Application Firewall (WAF), Address Space Layout Randomization (ASLR))
- Support Continuity of Operations
- Integrate Service Level Objectives (SLO) and Service Level Agreements (SLA) (e.g., maintenance, performance, availability, qualified personnel)

Domain 8: Supply Chain

- Implement Software Supply Chain Risk Management
- Analyze Security of Third-Party Software
- Verify Pedigree and Provenance
- Ensure Supplier Security Requirements in the Acquisition Process
- Support contractual requirements (e.g., Intellectual Property (IP) ownership, code escrow, liability, warranty, End-User License Agreement (EULA), Service Level Agreements (SLA))