CSSLP¬ Training and Certification

Course: **00069**
Filter: **Beginner**
Duration: **5 days**
Category:: **Gouvernance, Risk and Compliance**
Price: **2500,00 €**

About Course

Attend this official (ISC)² Certified Secure Software Lifecycle Professional (CSSLP) training and certification course and get prepared to achieve this premier secure software development certification. This course provides you with in-depth coverage on the skills and concepts in the eight domains of software security. This includes Software Concepts, Requirements, Design, Implementation, Testing, Lifecycle Management, among others. This course covers secure software development with the Certified Secure Software Lifecycle (CSSLP) and its domains. Topics include identifying security requirements, secure SDLC, manual testing, unit testing, functional testing, acceptance testing, and security testing, code review, and test automation. Students learn about security vulnerabilities, software testing, and source code. The course covers IAST (Interactive Application Security Testing tools, CI/CD (Continuous Integration/Continuous Delivery pipeline, and penetration testing to help prepare for the CSSLP exam. Passing the CSSLP Certification Exam meets U.S. DoD Directive 8140/8570.01 Information Assurance Security Architect/Engineer (IASAE) Level-I and Information Assurance Security Architect/Engineer (IASAE) Level-II requirements

What you'll learn

- Prepare for and pass the CSSLP Exam.
- Identify security software requirements.
- Follow secure coding practices.
- Develop a security testing strategy and plan.
- Choose a secure software methodology.
- Release software securely.

Pre-requisites

- This CSSLP course is for Software Developers, Engineers, Architects, Penetration Testers, and other IT (Information Technology) professionals who have a minimum of four years' experience in full-time Software Development Lifecycle (SDLC) in one or more of the eight domains covered in the CSSLP exam.

Curriculum

Module 1: Secure Software Concepts

- Core Concepts
- Security Design Principles

Module 2: Secure Software Requirements

- Define Software Security Requirements
- Identify 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

Module 3: Secure Software Architecture and Design

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

Module 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

Module 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

Module 6: Secure Software 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)

Module 7: Secure Software Deployment, Operations, 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 the appropriate level
- 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

Module 8: Secure Software Supply Chain

- Implement Software Supply Chain Risk Management
- Analyze the Security of Third-Party Software
- Verify Pedigree and Provenance
- Ensure Supplier Security Requirements in the Acquisition Process
- Support contractual requirements