INFOSECTRAIN

CSSLP

Certified Secure Software Lifecycle Professional

Certification Exam Outline

Course Description

The CSSLP course from InfoSec Train is spread out and covers all eight domains of the CSSLP certification. With 40 hrs of expert training by certified and experienced trainers and access to recorded sessions, the CSSLP training from InfoSec Train easily stands out in the industry.

Why CSSLP course from Infosec Train?

The CSSLP training from InfoSec Train is the best in the industry. Here are some compelling reasons to enroll for Infosec Train's CSSLP Training Course:

- Chapters are structured in an easy and understandable way
- All training is provided with engaging content and recordings are provided thereafter
- Trainers are the best in the industry with the CSSLP certification along with other Information security certifications.
- Trainers have several years of experience in the Information security industry as well as in the training industry
- Excellent guidance for clearing the certification exam

Target Audience

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

Pre-Requisites

A candidate who is planning to take the CSSLP exam should have 4 or more years of SDLC (Software Development Lifecycle Experience) experience in one or more of the eight domains of the CSSLP CBK. They can also attempt the exam if they have 3 years of SDLC experience in one or more domains of the CSSLP CBK along with a 4-year Baccalaureate degree in Computer Science or related fields.

Exam Information

Duration	3hours
Number of questions	125 questions
Question format	Multiple choice
Pass score	700 out of 1000

Note:

- CSSLP® is a registered mark of The International Information Systems Security Certification Consortium ((ISC)2).
- We are not an authorized training partner of (ISC)2.

CSSLP Examination Weights

Domains	Weight
Secure Software Concepts	13%
Secure Software Requirements	14%
Secure Software Implementation/Programming	16%
Secure Software Testing	14%
Secure Lifecycle Management	10%
Software Deployment, Operations, and Maintenance	9%
Supply Chain and Software Acquisition	8%

CSSLP Course Content

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

- 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)
- 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))