Arrow ECS Finland Oy - Education Services

TRAINING OFFERING

You can reach us at:

Arrow ECS Finland Oy, Lars Sonckin kaari 16, 02600 Espoo, Finland

Email: education.ecs.fi@arrow.com
Phone: 0870 251 1000

CSSLP - Certified Secure Software Lifecycle Professional

CODE:	LENGTH:	PRICE:
ISC_CSSLP	40 Hours (5 days)	Request Price

Description

Official ISC2 CBK Training for the CSSLP

Official ISC2® Training Seminar for the Certified Secure Software Lifecycle Professional (CSSLP®) provides a comprehensive review of the knowledge required to incorporate security practices – 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 students 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 ISC2 CSSLP Common Body of Knowledge (CBK®).

Official courseware is developed by ISC2– 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 ISC2 content.

Training features:

- Instruction from an ISC2 Authorized Instructor
- Official ISC2 Student Training Guide
- Chapter quizzes
- Interactive flash cards to reinforce learning
- 17 applied scenarios with corresponding activities teaching you how to apply security through the SDLC process.
- 24 discussions encouraging peer to peer interaction around key topics
- 7 content specific activities, including 6 case studies
- 8 end of chapter quizzes with answer explanation to assess comprehension
- 160 question post course assessment with answer explanation highlighting areas for further study

Course Domains

- Domain 1: Secure Software Concepts
- Domain 2: Secure Software Requirements
- Domain 3: Secure Software Architecture and Design
- Domain 4: Secure Software Implementation
- Domain 5: Secure Software Testing
- Domain 6: Secure Software Lifecycle Management
- Domain 7: Secure Software Deployment, Operations, Maintenance
- Domain 8: Secure Software Supply Chain

Objectives

After completing this course, the student will be able to:

- Discuss the core concepts of software security and the foundational principles that drive construction of resilient software.
- Discuss the security design principles as essential elements for building secure software.
- Discuss software security standards and frameworks, roadmaps and strategies and risk management.
- Explain security in software development methodologies, security metrics and security culture in software development.
- Identify and analyze software requirements pertaining to data privacy, security and compliance with laws and regulations.
- Describe requirement specification and tractability, misuse and abuse cases and flow down of security requirements to supplier.
- Explain secure architecture and design elements and patterns, architectural risk assessment, threat modeling, threat intelligence and attack surface evaluation.
- Explain security architecture and control identification, prioritization and positioning.
- Apply secure coding practices, analyze code for security risks and implement security controls.

- Discuss third-party code and libraries, software composition analysis and security of the build process.
- Discuss security testing strategy plan and analyze security testing methods.
- Discuss validation and verification, security test results and tracking security errors.
- Describe secure software integration and deployment, security data and post-deployment security testing.
- Recognize various security-relevant maintenance activities and discuss planning for the continuity of operations.
- Discuss software supply chain risks and analyze security of third-party software.
- Explain supplier security requirements in the acquisition process and support for contractual requirements.

Audience

This training course is intended for professionals who have at least four years of cumulative, paid work experience as a software development lifecycle professional in one or more of the eight domains of the ISC2 CSSLP CBK. The course builds on and brings together the holistic view of the topics covered in the everyday environment of an information assurance professional. Experience in the following professions will greatly enhance the learning environment:

- Software developers
- Engineers and architects
- Product managers
- Project managers
- Software QA
- QA testers
- Business analysts
- Professionals who manage these stakeholders

Programme

Domains/Modules/Chapters
This course covers the following chapters and learning objectives:

Chapter 1: Secure Software Concepts Domain

- Define core security objectives for software development.
- Describe the information security triad and explain the main mechanisms of confidentiality, integrity and availability of information.
- Characterize the relationship between information security and data privacy.
- Describe accountability, auditing and logging in the context of software security.
- Explain non-repudiation, digital signatures, benefits of code signing and blockchain.
- Understand the foundational concepts behind security design principles with respect to secure software development.

Chapter 2: Secure Software Lifecycle and Risk Management Domain

- Understand and describe OWASP’s Software Assurance Maturity Model (OpenSAMM) and Building Security In Maturity Model (BSIMM).
- Define and recognize security configuration standards and benchmarks.
- Understand and describe security-focused configuration management processes.
- Recognize security milestones.
- Explain and illustrate incorporation of software security practices into the SDLC processes.
- Discuss security in predictive and adaptive planning for software development.
- Describe DevOps and DevSecOps.
- Describe System Security Plan.
- Recognize security-relevant documentation.
- Evaluate metrics in software development.
- Recognize attack surface evaluation for measuring security in software.
- Describe software decommissioning, end-of-life policy and processes.
- Discuss data disposition.
- Explain information system continuous monitoring (ISCM).
- Describe security information event management (SIEM).
- Recognize risk management terminology and describe the risk management process.
- Explain regulations and legal aspects pertaining to intellectual properties and security breaches.
- Discuss architectural risk assessment.
- Describe operational risks relevant to integration and deployment environment.
- Recognize the importance of personnel training.
- Describe security champions and discuss the importance of security education and guidance.
- Explain retrospectives and continuous improvement in Agile development environments.
- Discuss lessons learned with respect to the processes used to build software.

Chapter 3: Secure Software Requirements Domain

- Discuss requirements management and identify sources for software security requirements.
- Recognize functional and nonfunctional requirements and explain the importance of security- focused stories in SCRUM/SCRUM-like methodologies.
- Analyze misuse/abuse cases and recognize their relevance to known attack patterns.
- Describe Security Requirements Traceability Matrix (STRM) and discuss how security requirements flow down to suppliers/providers.
- Analyze security policies and their supporting elements as internal sources for security requirements.
- Explain compliance requirements and recognize laws, regulations and industry standards as external sources for security requirements.
- Discuss security standards and frameworks.
- Describe data governance, explain data ownership, and recognize relevant roles and responsibilities.
- Describe data classification and explain security labeling and marking.
- Recognize data types, structured and unstructured.
- Describe the data lifecycle and explain the process for secure data retention and destruction.
- Discuss privacy risk, recognize privacy laws and regulations, and explain the requirements for safeguarding personal information.
- Discuss data anonymization and enumerate various approaches for anonymization.
- Explain user consent, data retention and data disposition in the context of privacy.
- Recognize implications of cross-border data transfer and restrictions for the transfer of personal data.

Chapter 4: Secure Software Architecture and Design Domain

- Understand common threats; describe the threat modeling process, tools and methodologies and explain the process of attack surface evaluation and management.
- Discuss threat intelligence and describe the sources for cyber threat information.
- Discuss the process of identification and prioritization of security controls and describe security properties and constraints on the design and constraints imposed by the deployment environment.
- Describe various architectures and discuss their security-relevant aspects.
- Describe pervasive computing and IoT, discuss various contactless technologies and discuss their security and privacy aspects.
- Explain embedded software and discuss the update challenge and discuss Field-Programmable Gate Array (FPGA) and microcontroller security.
- Explain cloud computing, service models and deployment models, and describe the shared security responsibility model. Discuss mobile applications security.
- Discuss hardware platform concerns, side channel mitigation, speculative execution mitigation, and Hardware Security Modules (HSM).
- Explain cognitive computing, machine learning and artificial intelligence.
- Discuss control systems and their applications in various areas and safety criticality aspects.
- Evaluate security criteria of interfaces, out-of-band management and log interfaces.
- Understand upstream and downstream dependencies, protocol design choices and their security ramifications.
- Describe various authentication and authorization mechanisms; explain credential management and the digital certificate standard.
- Discuss flow controls and data loss prevention; compare and contrast virtual machines and containers.
- Explain the trusted computing base (TCB) and the trusted platform module (TPM).
- Discuss database security, programming language environment, and operating system controls and services.
- Discuss secure architecture and secure design principles, and explain secure design patterns.
- Explain verification of the design, formal and informal secure code reviews and the code inspection process.

Chapter 5: Secure Software Implementation Domain

- Explain the need for establishing and enforcing secure coding standards.
- Describe different approaches for implementing security in managed applications.
- Describe common flaws in software and corresponding mitigation strategies.
- Discuss input validation, output encoding, authentication, session management, access control, cryptographic practices, error and exception management practices and logging.
- Explain type safety, memory management and isolation
- Discuss cryptography, applications to transit and storage, cryptographic agility, cryptographic libraries and encryption algorithm selection.
- Explain access control, trust zones and function permissions.
- Explain vulnerability databases and lists.
- Discuss Common Vulnerabilities and Exposures (CVE), Common Weakness Enumerations (CWE) and Common Attack Pattern Enumeration and Classification (CAPEC).
- Enumerate OWASP Top 10 Web Application Security Risks.
- Describe categorization of controls by type and by function.
- Describe controls to prevent common web application vulnerabilities
- Describe OWASP Proactive Controls and critical focus areas around building secure software.
- Evaluate the risks associated with using third-party and open-source components and libraries.
- Describe Software Composition Analysis (SCA) and open source management.
- Discuss OWASP Dependency Check and Dependency Track.
- Discuss API integration and evaluate the security aspects.
- Describe system-of-systems.

- Describe the build process, version control, and safeguards used to ensure integrity.
- Discuss anti-tampering techniques as part of software assurance.
- Explain the relation of compiler switches and warnings to the enhancement of security.

Chapter 6: Secure Software Testing Domain

- Explain functional and nonfunctional security testing, purpose and the phases in penetration testing fuzzing and its variations and limitations.
- Explain vulnerability scanning and content scanning.
- Discuss simulation, understand configuration drifts in development environments and describe real user monitoring and synthetic monitoring.
- Describe fault injection, stress testing and break testing.
- Describe various types of functional testing, including unit testing, integration testing and regression testing.
- Describe various types of nonfunctional testing, including scalability, interoperability and performance testing.
- Describe cryptographic validation and explain Pseudo-Random Number Generators and entropy.
- Explain test strategy and describe functional and nonfunctional testing.
- Explain the relationship between use cases and misuse and abuse cases and the importance of creating misuse and abuse cases.
- Explain test strategy and describe functional and nonfunctional testing.
- Describe test cases and test harness.
- Explain black-box and white-box testing, objectives and code coverage.
- Discuss application security testing (AST) methods and explain their benefits and limitations.
- Discuss manual code reviews and describe searching for embedded malicious code.
- Recognize software security-relevant standards, explain crowdsourcing benefits and concerns and discuss bug bounty.
- Explain the security implications of test results on product management and prioritization of remediation efforts.
- Explain break-build criteria.
- Describe the process of tracking security defects.
- Explain risk scoring, and the Common Vulnerability Scoring System (CVSS).
- Explain generation of test data, security of test data, ramifications of using production data in the test environment and database referential integrity and constraints.
- Describe the process of verification and validation testing and explain acceptance testing.
- List various software documentation and explain undocumented functionality.
- Describe OWASP's Application Security Verification Standard (ASVS), its structure and its goals.

Chapter 7: Secure Software Deployment, Operations and Maintenance Domain Learning Objectives

- Explain secure integration, build and deployment.
- Describe the secure software toolchain.
- Describe build artifacts and discuss mobile application and platform security.
- Describe security data, including credentials, keys and certificates and discuss ramifications of failing to protect them in production.
- Describe vaults used to manage secrets and discuss key vault considerations.
- Describe the secure bootstrapping process, hardening and the least privilege principle with respect to secure software installation.
- Explain secure software activation methods and security policy implementation with respect to secure software installation.
- Describe the Authorization to Operate (ATO) process and the steps involved.
- Explain risk acceptance.
- Explain post-deployment verification, issue tracking and testing constraints.
- Describe security testing automation.
- Describe the benefits of information security continuous monitoring(ISCM)and list some considerations for its implementation.
- Describe events, logs and threat intelligence.
- Explain computer security incidents, incident response and forensics.
- Describe incident precursors and indicators, monitoring logs and alerts and root-cause analysis.
- Describe security patch management and explain the timing, prioritization and testing aspects of security patches.
- Describe vulnerability management and vulnerability scan tools.
- Explain the operations of web application firewalls.
- Explain locality of reference, address space layout randomization and data execution prevention.
- Explain continuity of operations, business impact analysis, data backup and restore and data archiving.
- Discuss disaster recovery (DR), data residency requirement aspect of DR, resiliency and erasure code.

Chapter 8: Secure Software Supply Chain Domain

- Describe the software supply chain.
- Recognize participants in the supply chain.
- Explain software supply chain risk management.
- Explain security risks associated with third party/open source code and recognize OWASP's Software Component Verification Standard (SCVS).
- Describe software supply chain attacks.

- Explain the risks associated with peer-to-peer applications and file sharing.
- Explain code repository and build environment security.
- Explain cryptographically hashed, digitally signed components.
- Describe security in the acquisition process and audit of security policy compliance.
- Explain third-party vulnerability/incident notification and reporting and maintenance and support structure.
- Explain commercial and open-source software licenses.
- Explain vendor/supplier security track record in acquisition and the right-to-audit clause in contracts.
- Explain contractual requirements for intellectual property(IP) ownership inout sourcing relationships, code escrow, liability, warranty and service-level agreements (SLAs).

Chapter 9: Applied Scenario Activities Learning Objectives

- Apply security through the SDLC via animated video-based scenarios and corresponding activities.

Note: Throughout this course, exam domains may be covered in several chapters. Included in the course is a table indicating where the exam outline objectives are covered in this course. Unique icons are also used through the course materials to identify exam outline objectives.

Session Dates

Aikataulutamme kiinnostuksen mukaan. Ota yhteyttä

Additional Information

This training is also available as onsite training. Please contact us to find out more.