

SC205 ISC2 CSSLP Vorbereitung

Kurzbeschreibung:

Softwareentwickler und Sicherheitsexperten erlernen Best Practices für alle Phasen des Software Development Lifecycle (SDLC) – vom Design über Implementierung und Test bis zur Bereitstellung. Der Kurs bereitet gezielt auf die CSSLP-Zertifizierung von ISC2 vor und deckt das gesamte Themenspektrum des Common Body of Knowledge (CBK®) ab, um praxisnahes und umfassendes Sicherheitswissen zu vermitteln.

Zielgruppe:

Der Kurs **SC205 ISC2 CSSLP Vorbereitung** richtet sich an:

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

Voraussetzungen:

Um dem Lerntempo und Inhalten des Kurses **SC205 ISC2 CSSLP Vorbereitung** gut folgen zu können und um sich für die Zertifizierung zu qualifizieren, müssen Sie mindestens folgende Voraussetzungen zu erfüllen:

- Mindestens vier Jahre Berufserfahrung im Bereich SDLC
- Erfahrung in einem oder mehreren der acht Domänen des ISC2-CSSLP-Leitfadens oder drei Jahre Berufserfahrung in einem oder mehreren der acht Domänen des CSSLP-Leitfadens
- Abschluss in Informatik, Informationstechnologie (IT) oder verwandten Bereichen

Sonstiges:

Dauer: 5 Tage

Preis: 3450 Euro plus Mwst.

Ziele:

Ziel des Kurses **SC205 ISC2 CSSLP Vorbereitung** ist es, Ihnen ein fundiertes Wissen über sicherheitskritische Aspekte im gesamten Softwareentwicklungs-Lebenszyklus zu vermitteln. Sie lernen, wie Sie Sicherheit von Anfang an in den Softwareentwicklungsprozess einbinden und so die Entwicklung sicherer Anwendungen sicherstellen können.

Inhalte/Agenda:

- ♦ **Domain 1: Secure Software Concepts**
 - ♦ 1.1 Understand core concepts
 - ♦ 1.2 Understand security design principles
- ♦
- ♦ **Domain 2: Secure Software Lifecycle Management**
 - ♦ 2.1 Manage security within a software development methodology (e.g., Agile, waterfall)
 - ♦ 2.2 Identify and adopt security standards (e.g., implementing security frameworks, promoting security awareness)
 - ♦ 2.3 Outline strategy and roadmap
 - ♦ 2.4 Define and develop security documentation
 - ♦ 2.5 Define security metrics (e.g., criticality level, average remediation time, complexity, Key Performance Indicators (KPI), objectives and key results)
 - ♦ 2.6 Decommission applications
 - ♦ 2.7 Create security reporting mechanisms (e.g., reports, dashboards, feedback loops)
 - ♦ 2.8 Incorporate integrated risk management methods
 - ♦ 2.9 Implement secure operation practices
- ♦
- ♦ **Domain 3: Secure Software Requirements**
 - ♦ 3.1 Define software security requirements
 - ♦ 3.2 Identify compliance requirements
 - ♦ 3.3 Identify data classification requirements
 - ♦ 3.4 Identify privacy requirements
 - ♦ 3.5 Define data access provisioning
 - ♦ 3.6 Develop misuse and abuse cases
 - ♦ 3.7 Develop security requirement traceability matrix
 - ♦ 3.8 Define third-party vendor security requirements
- ♦
- ♦ **Domain 4: Secure Software Architecture and Design**
 - ♦ 4.1 Define the security architecture
 - ♦ 4.2 Perform secure interface design
 - ♦ 4.3 Evaluate and select reusable technologies
 - ♦ 4.4 Perform threat modeling
 - ♦ 4.5 Perform architectural risk assessment and design reviews
 - ♦ 4.6 Model (non-functional) security properties and constraints
 - ♦ 4.7 Define secure operational architecture (e.g., deployment topology, operational interfaces, Continuous Integration and Continuous Delivery (CI/CD))
- ♦
- ♦ **Domain 5: Secure Software Implementation**
 - ♦ 5.1 Adhere to relevant secure coding practices (e.g., standards, guidelines, regulations)
 - ♦ 5.2 Analyze code for security risks
 - ♦ 5.3 Implement security controls (e.g., watchdogs, file integrity monitoring, anti-malware)
 - ♦ 5.4 Address the identified security risks (e.g., risk strategy)
 - ♦ 5.5 Evaluate and integrate components
 - ♦ 5.6 Apply security during the build process
- ♦
- ♦ **Domain 6: Secure Software Testing**
 - ♦ 6.1 Develop security testing strategy & plan
 - ♦ 6.2 Develop security test cases
 - ♦ 6.3 Verify and validate documentation (e.g., installation and setup instructions, error messages, user guides, release notes)
 - ♦ 6.4 Identify undocumented functionality
 - ♦ 6.5 Analyze security implications of test results (e.g., impact on product management, prioritization, break/build criteria)
 - ♦ 6.6 Classify and track security errors
 - ♦ 6.7 Secure test data
 - ♦ 6.8 Perform verification and validation testing (e.g., independent/internal verification and validation, acceptance test)
- ♦
- ♦ **Domain 7: Secure Software Deployment, Operations, Maintenance**
 - ♦ 7.1 Perform operational risk analysis
 - ♦ 7.2 Secure configuration and version control
 - ♦ 7.3 Release software securely
- ♦

- ◊ 7.4 Store and manage security data
- ◊ 7.5 Ensure secure installation
- ◊ 7.6 Obtain security approval to operate (e.g., risk acceptance, sign-off at appropriate level)
- ◊ 7.7 Perform information security continuous monitoring
- ◊ 7.8 Execute the incident response plan
- ◊ 7.9 Perform patch management (e.g. secure release, testing)
- ◊ 7.10 Perform vulnerability management (e.g., tracking, triaging, Common Vulnerabilities and Exposures (CVE))
- ◊ 7.11 Incorporate runtime protection (e.g., Runtime Application Self Protection (RASP), web application firewall (WAF), Address Space Layout Randomization (ASLR), dynamic execution prevention)
- ◊ 7.12 Support continuity of operations
- ◊ 7.13 Integrate service level objectives and service-level agreements (SLA) (e.g., maintenance, performance, availability, qualified personnel)

-

-

-

-

- ◊ 8.1 Implement software supply chain risk management (e.g., International Organization for Standardization (ISO), National Institute of Standards and Technology (NIST))
- ◊ 8.2 Analyze security of third-party software
- ◊ 8.3 Verify pedigree and provenance
- ◊ 8.4 Ensure and verify supplier security requirements in the acquisition process
- ◊ 8.5 Support contractual requirements (e.g., intellectual property ownership, code escrow, liability, warranty, End-User License Agreement (EULA), service-level agreements (SLA))

-

-

-