Certified Secure Software Lifecycle Professional (CSSLP)

Length: 5 Days

Summary: Security should not be an after-thought when it comes to application development. Throughout the software development lifecycle, developers and stakeholders need to be knowledgeable and active in carrying out the latest security practices to protect an organization against vulnerabilities and attacks to their most sensitive data. This course provides you with in-depth coverage on the skills and concepts in the eight domains of software security including Software Concepts, Requirements, Design, Implementation, Testing, and Lifecycle Management among others.

Course Objectives: Upon completion of this course, students will have learned how to:

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

Who Should Attend: The intended audience for the CSSLP training program is professionals who are involved in any phase of the software development life-cycle and those who are responsible for application security practices. Typically speaking, CSSLP is ideal for those working in roles such as, but not limited to:

- 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

COURSE CONTENT

1: SECURE SOFTWARE CONCEPTS

- Core concepts
- Security design principles

2: SECURE SOFTWARE REQUIREMENTS

- Identify security requirements
- Interpret data classification requirements
- Identify privacy requirements

3: SECURE SOFTWARE DESIGN

- Perform threat modeling
- Define the security architecture
- Model (non-functional) security properties and constraints
- Evaluate and select reusable secure design
- Use security enhancing architecture and design tools
- Use secure design principles and patterns

4: SECURE SOFTWARE IMPLEMENTATION/PROGRAMMING

- Follow secure coding practices
- Analyze code for security vulnerabilities
- Implement security controls
- Fix security vulnerabilities
- Look for malicious code
- Securely reuse third party code or libraries
- Securely integrate components
- Apply security during the build process
- Debug security errors

5: SECURE SOFTWARE TESTING

- Develop security test cases
- Develop security testing strategy and plan
- Identify undocumented functionality
- Interpret security implications of test results
- Classify and track security errors
- Secure test data
- Develop or obtain security test data
- Perform verification and validation testing (e.g., IV&V)

6: SOFTWARE LIFECYCLE MANAGEMENT

- Secure configuration and version control
- Establish security milestones
- Choose a secure software methodology
- Identify security standards and frameworks
- Create security documentation
- Develop security metrics
- Decommission software
- Report security status
- Support governance, risk and compliance (GRC)

7: SOFTWARE DEPLOYMENT, OPERATIONS AND MAINTENANCE

- Perform implementation risk analysis
- Release software securely
- Securely store and manage security data
- Ensure secure installation
- Perform post-deployment security testing
- Obtain security approval to operate
- Perform security monitoring (e.g., managing error logs, audits, meeting SLAs, CIA metrics)
- Support incident response
- Support patch and vulnerability management
- Support continuity of operations

8: SUPPLY CHAIN AND SOFTWARE ACQUISITION

- Analyze security of third party software
- Verify pedigree and provenance
- Provide security support to the acquisition process