"CSSLP Exam Prep: Master Secure Software Lifecycle Practices"

Course Introduction:

The "CSSLP: Certified Secure Software Lifecycle Professional Exam Prep" course is designed to equip learners with the necessary knowledge and skills to successfully pass the CSSLP certification exam. This certification is pivotal for professionals involved in application security, as it validates their expertise in integrating security practices throughout the software development lifecycle (SDLC). Over the course of 5 days, participants will engage in a structured learning experience that covers the core domain areas of the CSSLP exam, enhancing their ability to implement secure software practices effectively.

Day 1: Introduction to Secure Software Concepts and Requirements

• Overview of Secure Software Development: Understand the importance of security in software development and the role of CSSLP certified professionals.

• Fundamental Security Concepts: Learn about confidentiality, integrity, and availability, which are the cornerstones of secure software.

• Software Lifecycle Overview: Explore the stages of the software lifecycle and how security is integrated into each phase.

• Understanding Security Requirements: Identify and gather security requirements to ensure software meets security standards from the outset.

• Risk Management Frameworks: Introduction to risk assessment methodologies and how they impact software security.

• Legal and Regulatory Compliance: Review key legal and regulatory requirements that influence secure software development, such as GDPR and HIPAA.

Day 2: Secure Software Design and Architecture

• Secure Design Principles: Delve into principles such as least privilege, defense in depth, and secure defaults to build robust software architectures.

• Threat Modeling: Learn to identify and mitigate potential threats through structured threat modeling processes.

• Design Review and Evaluation: Understand the importance of design reviews and how to

evaluate software architecture for security vulnerabilities.

• Secure Architecture Patterns: Explore common secure architecture patterns and how they can be applied to enhance security.

• Data Classification and Protection: Learn techniques for classifying and protecting data within software applications.

• Case Studies: Analyze real-world examples of secure and insecure software designs to reinforce learning.

Day 3: Secure Software Implementation and Testing

• Secure Coding Practices: Discover best practices for writing secure code, including input validation and error handling.

• Common Software Vulnerabilities: Study prevalent vulnerabilities such as SQL injection and cross-site scripting, and learn how to prevent them.

• Secure Testing Methodologies: Introduce various testing methodologies, including static and dynamic analysis, to identify security flaws.

• Penetration Testing and Ethical Hacking: Learn the basics of penetration testing and how ethical hacking can uncover vulnerabilities.

• Code Review Processes: Understand the importance of code reviews in the software development process and how to conduct them effectively.

• Automated Testing Tools: Familiarize with tools that automate security testing and integrate into continuous integration pipelines.

Day 4: Secure Software Deployment, Operations, and Maintenance

• Secure Deployment Strategies: Explore methods for deploying software securely, including the use of trusted platforms and environments.

• Change Management and Configuration: Learn about managing changes securely and maintaining configuration integrity.

• Incident Response and Recovery: Understand how to respond to security incidents and recover applications to a secure state.

• Monitoring and Logging: Discover techniques for monitoring software for security breaches and maintaining logs for analysis.

• Patching and Vulnerability Management: Learn the importance of regular patching and vulnerability assessments in maintaining software security.

• Secure Software Maintenance: Explore strategies for maintaining security throughout the software's operational life.

Day 5: Software Supply Chain and Exam Preparation

• Securing the Software Supply Chain: Understand the vulnerabilities associated with the software supply chain and how to mitigate them.

• Third-Party Software Evaluation: Learn methods for evaluating the security of third-party components and libraries.

• Continuous Improvement in Security Practices: Discover how to foster a culture of continuous security improvement in software teams.

• Exam Strategies and Techniques: Gain insights into effective study methods and strategies for tackling the CSSLP exam.

• Practice Exam and Review: Engage in a practice exam to assess your understanding and readiness for the official CSSLP certification.

• Final Q&A and Wrap-Up: Address any remaining questions and summarize key takeaways from the course to ensure confidence moving forward.

This comprehensive 5-day curriculum aims to provide a thorough understanding of secure software lifecycle principles, preparing participants for the CSSLP exam and enhancing their professional capabilities in secure software development.