EC-Council Certified Application Security Engineer (CASE) Java + Exam voucher

Duration: 3 Days **Course Code: CASE-JAVA** **Version: 1.0**

Overview:

The Certified Application Security Engineer (CASE) credential is developed in partnership with large application and software development experts globally. The CASE credential tests the critical security skills and knowledge required throughout a typical software development life cycle (SDLC), focusing on the importance of the implementation of secure methodologies and practices in today's insecure operating environment.
The CASE certified training program is developed concurrently to prepare software professionals with the necessary capabilities that are expected by employers and academia globally.It is designed to be a hands-on, comprehensive application security course that will help software professionals create secure applications. The training program encompasses security activities involved in all phases of the Software Development Lifecycle (SDLC): planning, creating, testing, and deploying an application.
Unlike other application security trainings, CASE goes beyond just the guidelines on secure coding practices and includes secure requirement gathering, robust application design, and handling security issues in post development phases of application development. This makes CASE one of the most comprehensive certifications on the market today. It is desired by software application engineers, analysts, testers globally, and respected by hiring authorities.

Target Audience:

Individuals involved in the role of developing, testing, managing, or protecting a wide area of applications or individuals hoping to become application security engineers/analysts/testers

Objectives:

- **After completing this course you should be able to:**
- Understand secure SDLC and secure SDLC models in-depth
- Apply the knowledge of OWASP Top 10, threat modelling, SAST and DAST
- Capture security requirements of an application in development
- Define, maintain and enforce application security best practices
- Perform manual and automated code review of application
- Conduct application security testing for web applications to assess the vulnerabilities
- Drive the development of a holistic application security program
- Rate the severity of defects and publishing comprehensive reports detailing associated risks and mitigations
- Work in teams to improve security posture
- Use Application security scanning technologies such as AppScan, Fortify, WebInspect, static application security testing (SAST), dynamic application security testing (DAST), single sign-on, and encryption
- Follow secure coding standards that are based on industry-accepted best practices such as OWASP Guide, or CERT Secure Coding to address common coding vulnerabilities.
- Create a software source code review process that is a part of the development cycles (SDLC, Agile, CI/CD)

Prerequisites:

To be eligible to apply to sit for the CASE exam the candidate must either:

- Attend the official EC-Council CASE training through an accredited EC-Council Partner (Accredited Training Centre/ iWeek/ iLearn) (All candidates are required to pay the USD100 application fee unless your training fee already includes this) or
- Be an ECSP (.NET/ Java) member in good standing or
- Have a minimum of 2 years working experience in InfoSec/ Software domain or

Testing and Certification

Recommended as preparation for the following exams:

- EC-Council Certified Application Security Engineer Exam - Candidates for this exam must have attended CASE training through an accredited EC-Council Partner and meet the other EC-Council Eligilibity Criteria.

- Have any other industry equivalent certifications such as GSSP .NET/Java

Content:

Understanding Application Security, Threats and Attacks

- What is a Secure Application
- Need for Application Security
- Most Common Application Level Attacks
- Why Applications become Vulnerable to Attacks
- What Consistutes Comprehensive Application Security
- Insecure Application: A Software Development Problem
- Software Security Standards, Models and Frameworks

Security Requirements Gathering

- Importance of Gathering Security Requirements
- Security Requirement Engineering (SRE)
- Abuse Case and Security Use Case Modeling
- Abuser amd Security Stories
- Security Quality Requirements Engneering (SQUARE)
- Operationally Critical Threat, Asset and Vulnerability Evaluation (OCTAVE)

Secure Application Design and Architecture

- Relative Cost of Fixing Vulnerabilities at Different Phases of SDLC
- Secure Application Design and Architecture
- Goal of Secure Design Process
- Secure Design Actions
- Secure Design Principles
- Threat Modeling
- Decompose Application
- Secure Application Architecture

Secure Coding Practices for Input Validation

- Input Validation Pattern
- Validation and Security Issues
- Impact of Invalid Data Input
- Data Validation Techniques
- Input Validation using Frameworks and APIs
- Open Source Validation Framework for Java
- Servlet Filters Validation Filters for Servlet
- Data Validation using OWASP ESAPI
- Data Validation: Struts Framework
- Data Validation: Spring Framework
- Input Validation Errors
- Common Secure Coding Practices

Secure Coding Practices for Authentication and Authorization

- Introduction to Authentication
- Types of Authentication
- Authentication Weaknesses and Prevention
- Introduction to Authorization
- Access Control Model
- EJB Authorization
- Java Authentication and Authorization (JAAS)
- Java EE Security
- Authorization Common Mistakes and Countermeasures
- Authentication and Authorization in Spring Security Framework
- Defensive Coding Practices against Broken Authentication and Authorization
- Secure Development Checklists: Broken Authentication and Session Management

Secure Coding Practices for Cryptography

- Java Cryptographic
- Encryption and Secret Keys
- Cipher Class
- Digital Signatures
- Secure Socket Layer (SSL)
- Key Management
- Digital Signatures
- Signed Code Sources
- Hashing
- Java Card Cryptography
- Spring Security: Crypto Module
- Do's and Dont's in Java Cryptography
- Best Practices for Java Cryptography

Secure Coding Practices for Session Management

- Session Management
- Session Tracking
- Session Management in Spring Security
- Session Vulnerabilities and their Mitigation Techniques
- Best Practices and Guidelines for Secured Sessions Management
- Checklist to Secure Credentials and Session ID's
- Guidelines for Secured Session Management

Secure Coding Practices for Error Handling

- Introduction to exceptions
- Erroneous Exceptional Behaviors
- Dos and Don'ts in Error Handling
- Spring MVC Error Handling
- Exception Handling in Struts 2
- Best Practices for Error Handling
- Introduction to Logging
- Logging using Log4j

Static and Dynamic Application Security Testing (SAST and DAST)

- Static Application Security Testing
- Manual Secure Code Review for Most Common Vulnerabilities
- Code Review: Check List Approach
- SAST Finding
- SAST Report
- Dynamic Application Security Testing
- Automated Application Vulnerability Scanning Tools
- Proxy-based Security Testing Tools
- Choosing between SAST and DAST

Secure Deployment and Maintenance

- Secure Deployment
- Prior Deployment Activity
- Deployment Activities: Ensuring Security at Various Levels
- Ensuring Security at Host Level
- Ensuring Security at Network Level
- Ensuring Security at Application Level
- Ensuring Security at Web Container Level (Tomcat)
- Ensuring Security in Orcale
- Security Maintenance and Monitoring

Further Information:

For More information, or to book your course, please call us on 00 966 92000 9278

training@globalknowledge.com.sa

www.globalknowledge.com/en-sa/

Global Knowledge - KSA, 393 Al-Uroubah Road, Al Worood, Riyadh 3140, Saudi Arabia