# Java Certified Application Security Engineer (JAVA CASE)

## Description

### Devenez un expert en sécurité des applications Java

La sécurité des applications est une priorité dans le développement moderne. Avec la certification Java Certified Application Security Engineer (JAVA CASE), vous apprendrez à concevoir, développer et maintenir des applications sécurisées, tout en répondant aux exigences croissantes en matière de cybersécurité. Cette formation couvre toutes les phases du cycle de développement des logiciels (SDLC), en mettant l'accent sur la sécurisation des applications dès la conception.

Que vous soyez développeur, analyste ou architecte logiciel, cette formation vous donnera les compétences nécessaires pour sécuriser les applications dans des environnements complexes. Vous apprendrez à identifier les failles de sécurité les plus courantes, à appliquer les meilleures pratiques de codage sécurisé et à utiliser des outils de test de sécurité. Ne manquez pas cette opportunité pour devenir un leader dans le domaine de la sécurité des applications.

**Contenu du cours**
**Module 1 : Comprendre la sécurité des applications, menaces et attaques**

- Qu'est-ce qu'une application sécurisée
- Besoin de la sécurité des applications
- Attaques courantes au niveau des applications
- Pourquoi les applications deviennent vulnérables
- Constitution d'une sécurité complète des applications
- Application non sécurisée : un problème de développement logiciel
- Normes, modèles et cadres de sécurité logicielle

**Module 2 : Collecte des exigences de sécurité**


© Copyright ITTA – Tous les contenus sont protégés par le droit d'auteur.

- Importance de la collecte des exigences de sécurité
- Ingénierie des exigences de sécurité (SRE)
- Modélisation des abus et des cas d'utilisation de sécurité
- Histoires de sécurité et d'abus
- SQUARE : Ingénierie des exigences de qualité en matière de sécurité
- Évaluation des menaces, actifs et vulnérabilités critiques (OCTAVE)

**Module 3 : Conception et architecture d'applications sécurisées**

- Coût relatif de la correction des vulnérabilités selon les phases du SDLC
- Objectif du processus de conception sécurisée
- Actions pour une conception sécurisée
- Principes de conception sécurisée
- Modélisation des menaces
- Décomposition des applications
- Architecture sécurisée des applications

**Module 4 : Pratiques de codage sécurisé pour la validation des entrées**

- Modèle de validation des entrées
- Problèmes de validation et de sécurité
- Impact des données non valides
- Techniques de validation des données
- Validation des entrées avec des frameworks et des API
- Utilisation de frameworks open source pour la validation en Java
- Filtres de validation pour les servlets
- Validation avec OWASP ESAPI
- Validation avec Struts et Spring Frameworks

**Module 5 : Pratiques de codage sécurisé pour l'authentification et l'autorisation**

- Introduction à l'authentification
- Types d'authentification
- Faiblesses de l'authentification et prévention
- Introduction à l'autorisation
- Modèles de contrôle d'accès
- Authentification et autorisation en Java EE
- Erreurs communes en authentification/autorisation et contre-mesures
- Pratiques contre l'authentification/session cassée

**Module 6 : Pratiques de codage sécurisé pour la cryptographie**

- Cryptographie en Java
- Chiffrement et gestion des clés
- Utilisation de la classe Cipher
- Signatures numériques et SSL
- Meilleures pratiques en cryptographie Java

**Module 7 : Gestion sécurisée des sessions**

- Gestion des sessions et suivi
- Vulnérabilités des sessions et techniques de mitigation


© Copyright ITTA – Tous les contenus sont protégés par le droit d’auteur.

- Meilleures pratiques de gestion sécurisée des sessions

**Module 8 : Pratiques de gestion des erreurs**

- Introduction à la gestion des exceptions
- Comportements erronés des exceptions
- Pratiques de gestion des erreurs
- Gestion des erreurs dans Spring MVC et Struts 2

**Module 9 : Tests de sécurité des applications (SAST et DAST)**

- Tests de sécurité statiques (SAST)
- Tests dynamiques (DAST)
- Outils de test automatisés et manuels

**Module 10 : Déploiement sécurisé et maintenance**

- Déploiement sécurisé à différents niveaux
- Sécurisation de l'hôte, du réseau et de l'application
- Maintenance et suivi de la sécurité

**Documentation**

- Support de cours numérique inclus

**Examen**

- Ce cours prépare à la certification Certified Application Security Engineer.
- Pour obtenir la certification Certified Application Security Engineer, les candidats doivent réussir l'examen : 312-96.
- Nombre de questions : 50
- Durée : 2 heures
- Si vous souhaitez passer cet examen, veuillez contacter notre secrétariat qui vous informera du coût des examens et prendra en charge toutes les démarches administratives nécessaires pour vous.

**Profils des participants**

- Développeurs Java
- Ingénieurs en sécurité des applications
- Architectes logiciels
- Analystes en sécurité
- Testeurs de sécurité des applications

**Connaissances Préalables**

- Connaissance de base en développement Java
- Compréhension des concepts de sécurité informatique
- Expérience pratique avec le cycle de développement logiciel (SDLC)
- Connaissances des outils de test de sécurité (SAST, DAST)
- Expérience en architecture logicielle

**Objectifs**

- Comprendre les concepts de sécurité du SDLC
- Appliquer les meilleures pratiques de sécurité des applications


© Copyright ITTA – Tous les contenus sont protégés par le droit d'auteur.

- Utiliser des outils de test de sécurité comme SAST et DAST
- Identifier et corriger les vulnérabilités courantes dans le code
- Concevoir des applications sécurisées dès la phase de conception
- Suivre les standards de sécurité des applications Java
- Mettre en œuvre des techniques de cryptographie en Java

**Description**
Formation Java Certified Application Security Engineer (JAVA CASE)
**Niveau**
Intermédiaire
**Prix de l'inscription en Présentiel (CHF)**
3950
**Prix de l'inscription en Virtuel (CHF)**
3800
**Durée (Nombre de Jours)**
3
**Reference**
CASE


© Copyright ITTA – Tous les contenus sont protégés par le droit d'auteur.