Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

Fast Track to Java Security

Aborde en profondeur les aspects de la sécurité du développement sous différents axes

Développeur, vous souhaitez enrichir vos connaissances en sécurité de la programmation et ainsi renforcer la robustesse de vos développements.
Manager ou dirigeant, vous souhaitez sensibiliser vos collaborateurs à la sécurité et ainsi valoriser cet aspect critique du E-commerce auprès de vos clients.
Voici la formation qu'il vous faut.

Elle aborde en profondeur les aspects de la sécurité du développement sous différents axes :
Le point de vue de l'attaquant avec une introduction au hacking applicatif
Comment coder ? les règles, les bonnes recettes et raccourcis à ne pas prendre. Cette section couvre les principes et règles de développement sécurisé ainsi que la mise en place de la sécurité à l'aide de l'intégration continue.
La sécurité dans les cycles de développement – SDLC
HTML5 ses nouvelles features et le lien avec la sécurité
La sécurité du développement en lien avec les bases de données
La sécurisation des serveurs et plateformes

Détails

- Code : FTJSD
- Durée : 5 jours (35 heures)

Public

- Chefs de projets
- Développeurs
- Développeurs d'application
- Ingénieurs Logiciels

Pré-requis

- Maîtrise du langage de programmation Java.

Objectifs

- Cette formation propose une approche complète du sujet. C'est une formation pratique couvrant des fonctionnalités Java de sécurité, les règles, les forces et les faiblesses.
- Elle aide les développeurs à comprendre comment écrire des applications Java sûres et robustes et fournit des connaissances avancées dans divers aspects du développement Java sécurisé qui peut prévenir efficacement les codes erronés et dangereux.
- Il en résulte des pratiques de codage de sécurité Java qui permettront de gagner un temps précieux et peut-être sauveront la réputation des organisations utilisant ces applications.

Programme

HACK1

- Introduction et fondements
  - Vue d'ensemble de la sécurité applicative
  - Comment évaluer la sécurité (connaître l'ennemi, Préparer ses défenses, contrôler son travail)
  - Definitions
- Les tests d'intrusion
  - Méthodologie et outils
  - Classification des risques
  - Exploitation
  - Guide de test OWASP
- Notions connexes (whitebox / blackbox) et analyse de risque.

HACK2

- La pratique – le laboratoire permettra à l'étudiant d'attaquer un serveur en combinant plusieurs vulnérabilités vues lors de la première session. Le but étant d'obtenir les droits administrateur du système d'exploitation du serveur
- Chaque étudiant aura sa cible et sa machine d'attaque

CODE1

- Introduction
- La sécurité des applications : Les attaques et les défenses
- Les règles à respecter pour développer de manière sécurisée
- Analyse des erreurs classiques
- Protection et classification des données par sensibilité
- Comment intégrer proprement une matrice d'autorisation dans un logiciel
- Avantages des outils modernes (Frameworks) dans la sécurité applicative actuelle

CODE2 et CODE3 (Labos)

- Apprentissage de l'identification les erreurs logicielles

Tél : +352 27 39 351
Fax : +352 27 39 52 68
Email : formation@oxiane.lu

classiques dans un code source
- Travaux pratiques avec les outils Jenkins, SonarQube et TFS pour construire un ensemble de métriques permettant de mesurer la qualité d'un code source
- Une deuxième phase de travaux pratiques visera à diminuer le temps de déploiement dans une optique DevOps et en évaluer les impacts sécurité.
- Le laboratoire sera orienté vers l'analyse statique et dynamique du code afin de couvrir les différentes branches de la sécurité avant une mise en production

HTML5

- Nouveautés HTML5
  - CORS
  - WebSocket
  - Canvas
- Nouveautés HTTP 2.0
  - SPDY
  - Pipelining
- Exemples pratiques :
  - Web browser fingerprint
  - Attaques cotés client : quels sont les risques de Javascript et les limites de la détection antivirus

SSDLC1

- En quoi consiste le Secure Software Development Life Cycle (SSDLC) ?
  - Raison d'être
  - Avantages
- Comment mettre en place un SSDLC ?
  - Vue d'ensemble de l'OWASP Open SAMM
  - Pourquoi utiliser ce référentiel pour votre SSDLC ?
  - Comment utiliser Open SAMM ?
- Exemples pratiques :
  - Choisir les bonnes pratiques de sécurités en fonction de "Security Practices" en fonction du contexte
  - Définir une roadmap

DB1 : Sécurisation des bases de données

- Chiffrement
  - Système de chiffrement
  - Gestion des clés
  - Accès en RAM
  - Sécurité du système d'exploitation
- Accès
  - Database firewall
  - Logs d'accès/Audit
  - Injections SQL/NoSQL
  - Droits du DBMS
  - Droits dans le DBMS

HOST1

- Introduction
  - Contexte
  - Définitions
  - Méthodologie
- Hardening Guide
  - Scenarii
  - Environnement Java
  - Environnement Tomcat
  - Référentiel de test OWASP
- Approches de sécurisation : surface d'attaque et exploitabilité.
- Validation des différentes phases de sécurisation
- Intégration du serveur dans une démarche de sécurisation globale.
- Différences Linux/Unix/Windows

HOST2 (Labos)

- L'étudiant validera les impacts du hardening sur un serveur d'application
- En tant qu'attaquant, les différentes limitations induitent par la sécurisation avancée, impliquent une limitation des possibilités d'exploitation, l'étudiant en verra les différents effets dans l'architecture Java

Modalités

- Type d'action :Acquisition des connaissances
- Moyens de la formation :Formation présentielle – 1 poste par stagiaire – 1 vidéo projecteur – Support de cours fourni à chaque stagiaire
- Modalités pédagogiques :Exposés – Cas pratiques – Synthèse
- Validation :Exercices de validation – Attestation de stages