Cyberhygiène pour les développeurs

CYBERSÉCURITÉ

Objectifs	Sensibiliser les développeurs aux risques de sécurité applicative - Apprendre à intégrer la sécurité dès les premières étapes du développement logiciel - Maîtriser les outils et pratiques DevSecOps pour automatiser la détection des vulnérabilités - Mettre en oeuvre des contre-mesures concrètes dans les pipelines CI/CD et le code source.
Participants	Développeurs (front, back, fullstack) - DevOps - ingénieurs logiciels - architectes applicatifs - chefs de projet technique - étudiants en développement logiciel avec bases solides.
Prérequis	Connaissances en développement logiciel - bonnes pratiques de codage sécurisé - maîtrise des vulnérabilités courantes (OWASP Top 10) - sensibilisation aux tests et revues de code.
Moyens pédagogiques	1 poste par participant - 1 vidéoprojecteur - Support de cours fourni à chaque participant - Ateliers Individuels - Modalités d'évaluation : Ateliers (TP) pendant tout le long de la formation et évaluation des acquis tout au long de la formation
Méthodes pédagogiques	Exposés interactifs et démonstrations - Travaux pratiques individuels et en groupe - Échanges d'expériences et de bonnes pratiques
Type de formation	Formation présentielle ou distancielle, selon les besoins et les contraintes des participants
Tarif inter-entreprise	3750 € HT
Durée	5 jour(s) - 35 heure(s)

Code : NCI_WXA9QUCC

Programme :

Sécurité par conception & erreurs classiques
Rappels OWASP Top 10 (API, Web, Mobile)
Menaces liées au code source : injections, XSS, SSRF, RCE, IDOR
Sécurité by design : principes fondamentaux (least privilege, fail securely)
Gestion sécurisée de la configuration (env, secrets, variables)

TP
Analyse de code vulnérable (Node.js, PHP ou Python)
Correction en live d'injections SQL / XSS
Mise en place de fichiers .env sécurisés et usage de dotenv

Contrôle de code et gestion des dépendances
Dépendances vulnérables (npm audit, pip-audit, etc.)
SAST (Static Application Security Testing) et linters de sécurité
Git & sécurité : secrets dans le repo, bonnes pratiques GitOps
Licences & paquets non maintenus

TP
Intégration de semgrep ou SonarQube dans un projet
Scans de dépendances avec Snyk, npm audit, Trivy
Détection et retrait de secrets dans Git (git-secrets, gitleaks)

Sécuriser les APIs & CI/CD
AuthN / AuthZ pour API : JWT, OAuth2, scopes, rate-limiting
Vulnérabilités API (exemple : Broken Object Level Authorization)
DevSecOps : Intégration sécurité dans GitHub Actions / GitLab CI
Secrets dans pipelines, validation, branches protégées

TP
Attaques sur API REST vulnérable (ex: FastAPI ou Express)
Correction et ajout d'authentification sécurisée (JWT, scopes)
Mise en place d'un pipeline CI avec scan SAST + tests de sécurité

Conteneurs, secrets et revue de code
Conteneurs : bonnes pratiques Docker (images minimales, user non-root)
Scans de conteneurs & durcissement (Trivy, DockerBench)
Gestion des secrets (Vault, AWS Secrets Manager, Doppler)
Politique de revue de code orientée sécurité

TP
Création d'un Dockerfile sécurisé
Scan de vulnérabilités sur une image applicative
Intégration de Vault dans une app Node ou Python
Revue de code d'une pull request avec checklist sécurité

Atelier DevSecOps complet
Déploiement complet d'un projet DevSecOps sécurisé
Setup CI/CD complet (GitHub Actions ou GitLab CI)
Scans automatisés (SAST, dépendances, conteneurs)
Authentification sécurisée sur une API
Dockerisation + configuration Vault pour secrets

Mise en production simulée avec logs de sécurité activés
Scénario bonus :
Une vulnérabilité est volontairement introduite
Le stagiaire doit la détecter dans la chaîne CI ou à l'exécution

WEDO 365 by NCI

Caen - Paris - Lyon - Rennes - Nantes - Le Havre - Nice - Montpellier - Lille - Toulon
Saint-Etienne - Toulouse - Montpellier - Bordeaux - Grenoble - Le Mans - Tours - Marseille
www.wedo365.fr - commercial@nci.fr - Tél : +33 (0)2 14 99 00 20