+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Mis à jour le 14/09/2023

S'inscrire

Formation Reconversion Cybersécurité : DevSecOps

10 jours (70 heures)

PRÉSENTATION

Les métiers de la cybersécurité comme les métiers du DevOps font face à une pénurie de talents. Le DevSecOps permet aux entreprises d'accélérer leur rythme de développement en intégrant les bonnes pratiques de sécurité. Cette approche utilise des outils d'intégration continue et d'automatisation pour améliorer la délivrabilité dé et la sécurité des applications.

Cette formation reconversion à la cyber sécurité vous formera aux bonnes pratiques dans le domaine de la sécurisation des développements. Nous nous baserons sur les meilleurs référentiels et pratiques du marché et les diffuserons de manière homogène auprès de vos équipes de développement.

Notre formation reconversion cybersécurité alternera théorie et cas pratiques afin de garantir une mise en pratique opérationnelle suite à la formation. Une évaluation des acquis devra être validée pour mesurer l'assimilation des concepts dispensés.

OBJECTIFS

- Comprendre les principaux concepts, principes et normes pour la sécurité des développements
- Pouvoir développer et tester des applications répondant aux exigences de conformité et de certification

PUBLIC VISÉ

- Développeurs
- Architectes techniques
- Chefs de projets

Pré-requis

- Bonnes connaissances en Windows et Linux/UNIX
- Bonnes connaissances en TCP/IP
- Bonnes connaissances en HTTP, JavaScript et développement

PROGRAMME DE NOTRE FORMATION reconversion cybersécurité : DEVSECOPS

Avant de commencer

- Fondamentaux de la sécurité Web
  - Sécurité Web : Vue d'ensemble
  - Sécurité des serveurs Web
  - Applications Web et sécurité
  - Réduction des risques
  - Les applications Web, des cibles courantes
  - Qui sont les pirates ?
  - Concevoir des applications sécurisées
  - Tester les applications
  - OWASP (Open Web Application Security Project)
- Règles de Secure Coding .NET & C#
- Règles de Secure Coding Java avec CERT Oracle Java Secure Coding
- Exercice 1.1
  - Configuration du cours
  - Application Web du cours
  - Microsoft IIS and Apache Tomcat
  - Application .NET Altovo Mutual
  - Application Java OWASP WebGoat
  - Exercice 1.1 : Examiner l'application Web du cours
  - Résumé du chapitre
  - Questions de synthèse

Fondamentaux de la sécurité web

- Besoins des applications en matière de sécurité
  - Nécessités de sécurité
  - Répondre aux nécessités de sécurité
  - Codage : Bonnes pratiques
  - Responsabilité du développeur

- Vue d’ensemble du fonctionnement d’HTTP
  - Le protocole HTTP (Hypertext Transfer Protocol)
  - Interaction entre navigateur et serveur Web
  - Fonctionnement général de HTTP 1.0
  - Requêtes HTTP de client
  - Exemple de requête GET
  - Réponse du serveur
  - Codes de statut de HTTP
  - Autres méthodes HTTP
  - Envoyer une requête HTTP
  - POST ou GET
  - POST vs. GET en action
  - Désactiver autocomplete
  - Qu’est-ce qu’un cookie ?
  - Pourquoi les cookies ?
  - Fonctionnement des cookies
  - Recommandations en matière de cookies
- Technologies de sécurité
  - Chiffrement
  - Résistance du chiffrement
  - Chiffrement à clé symétrique
  - Chiffrement à clé publique
  - Intercepter le trafic Web
  - Authentification à clé publique
  - Certificat numérique
    - Cycle de vie d'un certificat
    - La fabrique de certificats Java
    - Les certificats de modification X509
  - Contenu d’un certificat numérique
  - Types de certificats
  - Certificat : Processus d’émission
  - Authentification à clé publique avec certificats numériques
  - Examiner un certificat numérique
  - Empreinte numérique
  - Intégrité des données
  - Vérifier l’intégrité des données
  - Intégrité des messages
  - Signature numérique
  - Codes d’authentification de message
- Technologies de sécurité
  - MITIM Attacks
  - Modifier un message signé
  - Codage
  - Codage et chiffrement
  - Résumé du chapitre
  - Questions de synthèse

Améliorer la sécurité des serveurs web

- Gestion de la configuration
  - Protection des serveurs Web
  - Processus de gestion des correctifs
  - Minimalisation des serveurs
  - Supprimer les logiciels, services et utilisateurs non essentiels
  - Accès distant
  - Vérifier la configuration de sécurité du système
  - Méthodes HTTP
  - Fuite d'informations
  - Vérifier l'intégrité du système
- Exercice 3.1 15
  - Exercice 3.1 : Détecter des modifications du système de fichiers
- Authentification par serveur
  - Accès des serveurs Web au contenu Web : Apache sur Windows
  - Accès des serveurs Web au contenu Web : Apache sur Linux/UNIX
  - Accès des serveurs Web au contenu Web : IIS
  - Authentification gérée par le serveur
  - Fonctionnement de l'authentification de base
  - Authentification de base : Apache Tomcat
  - Authentification de base : IIS
  - Découvrir le mot de passe d'un utilisateur
  - Sécuriser l'accès à un site Web
- Permissions du système de fichiers
  - Fichiers journaux
  - Sécurité des fichiers journaux
  - Permissions pour Apache sur Windows
  - Permissions pour IIS
  - Serveurs d'application
  - Permissions des serveurs d'application
  - Permissions et objets
- Exercice 3.2
  - Exercice 3.2 : Configurer des permissions de fichiers de serveur Web
- Chiffrer le trafic Web
  - Transport Layer Security (TLS)
  - Fonctionnement de TLS
  - Le protocole de transfert TLS
  - Détail du protocole de connexion
  - Authentification par certificat
  - Mécanismes de sécurité de TLS
  - Alertes TLS
  - Activer HTTPS sur un serveur Web
- Exercice 3.39
  - Exercice 3.3 : Activer HTTPS
  - Résumé du chapitre
  - Questions de synthèse

Sécuriser les applications web

- Sécurité des applications Web
  - Le projet OWASP (Open Web Application Security Project)
  - Guides OWASP
  - Le Top 10
  - Le SANS TOP 25

- Validation des entrées
  - Validation des entrées
  - Entrée d’utilisateur
  - Édition des champs non éditables
  - Stratégie de validation
  - Frontière de confiance des applications Web
  - Désactiver la validation côté client
  - Techniques de validation
  - Logique de validation
  - Éléments d’expressions régulières
  - Veiller à valider et non pas juste repérer
  - Validation de requête .NET
- Exercice 4.10
  - Exercice 4.1 : Sécuriser la validation des entrées
- Failles d’injection2
  - Injection : Définition
  - Failles d’injection
  - Injection de commande
  - Injection SQL
  - Injection SQL en action
  - Empêcher les injections SQL
  - Requêtes paramétrées fortement typées
  - Prévenir les injections SQL par des énoncés préparés Java
  - Prévenir les injections SQL par des paramètres nommés en .NET
- Exercice 4.25
  - Exercice 4.2 : Éviter les injections SQL
- Cross site scripting (XSS)
  - Cross site scripting (XSS)
  - XSS en action
  - Se protéger contre le cross site scripting
  - Codage d’entité HTML : Java
  - Codage d’entité HTML : .NET
- Exercice 4.35
  - Exercice 4.3 : Éviter les vulnérabilités XSS
- Sécuriser l’accès
- Authentification et gestion de session
  - Maintenir les sessions d’utilisateur
  - Identifiants de session faibles
  - Intercepter des identifiants de session
  - Détournement de session
  - Identifiants de session solides
- Stockage et communication non sécurisés
  - Stocker et envoyer des données
  - Stockage sécurisé des données
  - Ne pas afficher complètement les données confidentielles
  - Sécuriser les communications
- Restriction d’accès à l’URL
  - Contrôler l’accès aux pages
  - Faible restriction d’accès aux URL
  - Forte restriction d’accès aux URL
  - Mettre en œuvre la restriction d’accès aux URL
  - Filtres de servlet Java
- Exercice 4.4
  - Exercice 4.4 : Restreindre l’accès aux URL
- Sécuriser les informations

- Référence d’objet direct non sécurisée
  - Référence d’objet non sécurisée
  - Éviter les références d’objet non sécurisées
  - Falsifier une identité d’utilisateur
  - Redirections et transferts non validés
  - Mesures de prévention contre les redirections et les transferts non validés
- Falsification de requêtes inter-sites
  - Falsification de requêtes inter-sites
  - Comment la falsification de requêtes inter-sites est-elle possible ?
  - Fonctionnement de la falsification de requêtes inter-sites
  - Réduire les vulnérabilités de falsification de requêtes inter-sites
  - Éviter la falsification de requêtes inter-sites pour toute application
  - Éviter la falsification de requêtes inter-sites : Java
  - Éviter la falsification de requêtes inter-sites : .NET
- Fuite d’informations et traitement des erreurs
  - Fuite d’informations
  - Traitement des erreurs et des exceptions
  - Toujours s’attendre à l’inattendu
  - Traitement global des exceptions avec Java
  - Traitement global des exceptions avec .NET
- Exercice 4.5
  - Exercice 4.5 : Fuites d’information
  - Résumé du chapitre
  - Questions de synthèse

Améliorer la sécurité Ajax

- Principes d’Ajax
  - Ajax : Définition
  - Action d’Ajax
  - Fonctionnement d’Ajax
  - Exemple de code XHR
  - Ajax en action
  - Firebug en action
  - Transfert de données d’Ajax
  - JSON (JavaScript Object Notation)
- Identifier une exposition accrue aux risques
  - Vulnérabilités Ajax/Web 2.0
  - Surface d’attaque accrue
  - Reconnaissance réseau
  - Restrictions de sécurité XHR
  - Proxy de transfert Ajax
  - JavaScript à la demande
  - Ajax peut-il détourner une session ?
  - Falsification de requêtes inter-sites avec JavaScript et Ajax
  - Voler des données JSON sensibles
  - Éviter le vol de données JSON
- Exercice 5.19
  - Exercice 5.1 : Empêcher le vol de données sous Ajax
  - Résumé du chapitre
  - Questions de synthèse

Sécurité des services web

- Fondamentaux XML
  - Fondamentaux de XML
  - Documents XML
  - Espaces de nommage XML
  - Schéma XML
  - Valider un document XML
  - Contenu des schémas
- Fonctionnement des services Web
  - Service : Définition
  - Services Web
  - Standards des services Web en XML
  - Messages SOAP
  - Les services Web de type REST (Representational State Transfer)
  - Services Web et Ajax
- Vulnérabilités XML des services Web
  - Sécurité des services Web : Problèmes
  - Faiblesses de XML
  - Atténuer les faiblesses de XML
- Exercice 6.13
  - Exercice 6.1a : Valider des messages SOAP
  - Exercice 6.1b : Valider des messages REST FULL
- Sécurité des messages
  - HTTPS, un moyen facile et efficace
  - Inconvénients de HTTPS
  - Chiffrement XML
  - Document après chiffrement XML
  - Sécurité XML pour services Web
  - WS-Security
- Exercice 6.22
  - Exercice 6.2 : Explorer WS-Security
  - Résumé du chapitre
  - Questions de synthèse
  - Notes

Rechercher les faiblesses des applications

- Vue d’ensemble des tests de vulnérabilités
  - Détecter les failles de sécurité des applications
  - Scanneurs de vulnérabilités généralistes
  - Tester et analyser des stratégies
  - 1. Analyse manuelle du code source
  - 2. Analyse automatisée du code source
  - 3. Analyse manuelle à partir du réseau
  - 4. Analyse automatisée à partir du réseau
- Outils manuels de test
  - WebScarab
  - Fonctions de WebScarab
  - Analyse d’identifiants de session avec WebScarab
  - Fuzzing avec WebScarab
- Exercice 7.1
  - Exercice 7.1 : Analyse manuelle avec WebScarab

- Outils d’analyse d’applications Web
  - Action d’un scanneur d’application
  - Faux positifs et faux négatifs
  - IBM AppScan (et/ou outil utilisé par Cegid)
  - OWASP ZAP
- Exercice 7.26
  - Exercice 7.2 : Analyse automatisée d’une application Web
  - Résumé du chapitre
  - Questions de synthèse
  - Notes

Règles de Secure Coding Microsoft .NET & C#

- Bonnes pratiques de développement sécurisé des applications .NET & C#
  - Permission Requests
  - Securing State Data
  - Securing Method Access
  - Securing Wrapper Code
  - Security and Public Read-only Array Fields
  - Securing Exception Handling
  - Security and User Input
  - Security and Remoting Considerations
  - Security and Serialization
  - Security and Race Conditions
  - Security and On-the-Fly Code Generation
  - Dangerous Permissions and Policy Administration
  - Problèmes de sécurité et d'installation
  - Comment lancer partiellement un code fiable en Sandbox ?

Règles de Secure Coding Java

- Bonnes pratiques de développement sécurisé des applications Java
  - Sécurisation de la JVM
    - Limites naturelles imposées par Java : gestion mémoire
    - Contrôle du bytecode par la machine virtuelle
    - Mise en œuvre du Security ClassLoader
  - Protection de l'exécution
    - Exécution protégée : SecurityManager, ClassLoader
    - Surcharge des méthodes d'accès : lecture, écriture, exécution, ouverture de socket, autorisation de connexions...
  - Contrôle
    - Rappel sur les ACL
    - Le paquetage java.security.acl
    - Ajout d'entrée, vérification d'accès
  - Obfuscation
    - Principe
    - Techniques d'obfuscation
    - Solutions commerciales
  - JAAS
    - Présentation
    - Fonctionnement et mise en œuvre
  - CERT Oracle Secure Coding Rules :
    - Input Validation and Data Sanitization (IDS)
    - Declarations and Initialization (DCL)
    - Expressions (EXP)
    - Numeric Types and Operations (NUM)
    - Object Orientation (OBJ) Methods (MET)
    - Exceptional Behavior (ERR)
    - Visibility and Atomicity (VNA)
    - Locking (LCK)
    - Thread APIs (THI)
    - Thread Pools (TPS)
    - Thread-Safety Miscellaneous (TSM)
    - Input Output (FIO)
    - Serialization (SER)
    - Platform Security (SEC)
    - Runtime Environment (ENV)
    - Miscellaneous (MSC)

Règles de Secure Coding Microsoft PHP

- Bonnes pratiques de développement sécurisé des applications PHP
  - Client-side Security
    - Javascript security
    - AJAX security
    - HTML5 security
  - PHP Security Services
    - ?Cryptography extensions in PHP
    - Input validation APIs
  - PHP Environment
    - ?Server Configuration
    - Securing PHP configuration
    - Environment security
    - Hardening
    - Configuration management
  - Advice and Principles
    - ?Matt Bishop’s principles of robust programming
    - The security principles of Saltzer and Schroeder
  - Input validation
    - Input validation concepts
    - Remote PHP code execution
    - MySQL validation errors – beyond SQL Injection
    - Variable scope errors in PHP
    - File uploads, spammers
    - Environment manipulation
  - Improper use of security features
    - ?Problems related to the use of security features
    - Insecure randomness
    - Weak PRNGs in PHP
    - Stronger PRNGs we can use in PHP
    - Password management – stored passwords
    - Some usual password management problems
    - Storing credentials for external systems
    - Privacy violation
    - Improper error and exception handling
    - Classification of security flaws
  - Time and State problems
    - ?Concurrency and threading
    - Concurrency in PHP
    - Preventing file race condition
    - Double submit problem
    - PHP session handling
    - A PHP design flaw – open_basedir race condition
    - Database race condition
    - Denial of service possibilities
    - Hashtable collision attack
    - Classification of security flaws
  - Using Security Testing Tools
    - Web vulnerability scanners
    - SQL injection tools
    - Public database
    - Google hacking
    - Proxy servers and sniffers

Sociétés concernées

Cette formation s’adresse à la fois aux particuliers ainsi qu’aux entreprises, petites ou grandes,

souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.