Formation Sécurité des applications et des serveurs web

Code : SEC-WEBAPP-100 Durée : 5 jours (35 heures)

Public visé

Développeurs, ingénieurs DevOps, pentesters et RSSI en sécurité applicative.

{{< formation-prerequis >}}

Objectifs pédagogiques

À l'issue de cette formation, vous serez capable de :

- Intégrer la sécurité dès le début du cycle de développement (DevSecOps)
- Analyser les attaques pour identifier les vecteurs et les méthodes
- Identifier les principales menaces et vulnérabilités des applications Web
- Décrire les différentes techniques d'attaques utilisées par les pirates
- Appliquer des pratiques de codage sécurisé
- Utiliser des frameworks et bibliothèques de sécurité
- Implémenter des mécanismes d'authentification et de gestion des sessions
- Mettre en place des contrôles d'accès
- Effectuer des audits et tests de sécurité
- Assurer la conformité (RGPD, OWASP Level 2, PCI-DSS)
- Encourager la collaboration développement-sécurité
- Promouvoir une culture de sécurité organisationnelle

Programme

Jour 1 - Matin

Introduction à la cybersécurité

- Les enjeux d'un système d'information
- Le panorama des risques actuels
- Les profils des attaquants et leurs objectifs
- Les méthodes et outils des attaquants
- Les vecteurs d'attaque
- Les grandes familles d'attaques

- Les phases d'une attaque (Cyber Kill Chain)

Le pilotage et la maîtrise des risques

- La prise de conscience de l'importance de la sécurité
- Organismes d'État et indépendants (ANSSI, SGDSN, CERT, OWASP)
- Exigences légales et contexte juridique (Article 323, Loi Godfrain, RGPD)
- Référentiels et normes (ISO 270xx, IEC 62443, OWASP, HDS, PCI-DSS)
- Standards de gestion de vulnérabilités (MITRE, NVD, CVE, CVSS, CWE, IOC, OTX, Exploit, TTP)
- Threat Modeling et framework ATT&CK
- Enjeux de la sécurité des applications Web

Travaux pratiques : Modélisation des enjeux de sécurité d'un site Web

Jour 1 - Après-midi

Introduction au DevSecOps

- Le rôle de la sécurité dans le cycle de développement
- La méthodologie DevSecOps
- Les principes de sécurité dans le développement
- Les outils DevSecOps (Jenkins, GitLab CI/CD, SonarQube)
- Les frameworks sécurisés (Spring Security, Express.js avec Helmet)
- L'intégration continue de la sécurité (SAST, DAST)

Rappels sur les technologies du Web

- Le protocole HTTP
- Les headers
- Les status codes
- Les méthodes

Travaux pratiques : Affichage et analyse d'une requête GET et POST dans les DevTools

Les technologies de sécurisation

- Les techniques d'authentification (LM, Challenging, Kerberos, LDAP, MFA)
- L'authentification centralisée et unique (CAS, SSO, WebSSO, OAuth, OpenID)
- Les techniques de hash (MD5, AES, RSA, SSL, TLS)
- Les techniques de chiffrement (symétrique, asymétrique, AES, TLS)
- Les clés et certificats numériques
- Les protocoles de vérification (WindBind, SASL, GSSAPI)
- Les modèles de définition des autorisations (ACL 1.x, ACL 2.x)
- Les normes de gestion des groupes et rôles (RBAC, PDP, PEP)

Travaux pratiques : Mise en oeuvre de l'environnement de formation

Jour 2 - Matin

Les vulnérabilités du développement

- Introduction au Top 10 OWASP, Top 25 SANS et Veracode
- Présentation des principales catégories de vulnérabilité :

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery (SSRF)

Jour 2 - Après-midi

Travaux pratiques : Mise en évidence et exploitation de vulnérabilités :

- *Broken Access Control*
- *Injection (côté serveur et client)*
- *Insecure Design*
- *Security Misconfiguration*
- *Vulnerable and Outdated Components (avec Nessus, OpenVAS)*
- *Web application scanning (Acunetix, Netsparker, Nikto, Wapiti, Qualys WAS)*
- *Identification and Authentication Failures (Hydra, Medusa, Burp Suite, Wfuzz, Patator)*
- *Software and Data Integrity Failures*
- *Security Logging and Monitoring Failures*
- *Server-Side Request Forgery (SSRF)*

Jour 3 - Matin

Les classes d'outils exploités par les attaquants

- Les scanners de vulnérabilité Web (Burp Suite, Netsparker, Acunetix, WPscan, Nikto)
- Les outils d'injection SQL (Sqlmap, Havij, SQLNinja)
- Les outils de fuzzing (Wfuzz, Skipfish, Arachni)
- Les outils de brute-force (Hydra, Medusa, Patator)
- Les outils de manipulation de requêtes (Burp Suite, Postman)
- Les outils de contournement de l'authentification
- Les outils d'exploitation des failles XSS (XSSer, Xenotix, BeEF)
- Les outils de déni de service (LOIC, HOIC, Slowloris)

Travaux pratiques : Mise en oeuvre d'outils (scan, injection, fuzzing, brute-force, manipulation de requêtes, contournement, XSS, déni de service)

Jour 3 - Après-midi

Les attaques avancées

- Les LFI et RFI
- Les Wrappers

- L'écriture d'un Shell code de type RCE

Travaux pratiques : Exploitation d'une LFI pour créer un Shell code de type RCE

Jour 4 - Matin

La sécurité du développement
Les apports de l'OWASP :

- Les objectifs
- Les ressources (Testing Guide, Security Guide, Code Review Guide, WebGoat, Juice Shop)
- Le Level 2 et ses directives

Les bonnes pratiques de sécurisation :

- Broken Access Control
- Cryptographic Failures
- Injection
- Insecure Design
- Security Misconfiguration
- Vulnerable and Outdated Components
- Identification and Authentication Failures
- Software and Data Integrity Failures
- Security Logging and Monitoring Failures
- Server-Side Request Forgery

Jour 4 - Après-midi

La sécurité du développement - Suite

- La validation et le filtrage des données en entrée et en sortie
- Les tokens anti-CSRF
- La sécurité des sessions et des cookies
- L'utilisation de CSP (Content Security Policy)
- L'impact de SOP (Same-Origin Policy)
- L'utilisation de CORS (Cross-Origin Resource Sharing)
- L'utilisation de HSTS (HTTP Strict Transport Security)
- L'option X-Frame
- La sécurité des API et des WebServices
- L'obfuscation et la minification

Travaux pratiques : Sécurisation d'un script contre les attaques

L'automatisation de la sécurisation

- Automatisation des tests de sécurité
- Intégration de la sécurité dans les pipelines DevOps
- Méthodes de détection des vulnérabilités dans le code
- Utilisation d'outils de test d'intrusion automatisés
- L'analyse statique du code (SAST)

- L'analyse dynamique du code (DAST)

Jour 5 - Matin

Travaux pratiques :

- *Analyse statique de code (SonarQube, Checkmarx, Veracode, RIPS)*
- *Analyse dynamique de code (OWASP ZAP, Burp Suite, Acunetix, Netsparker, AppSpider, W3af)*
- *Proposition de solutions pour corriger les problèmes*

La sécurité du chemin d'exécution

- Limiter la phase de reconnaissance
- Limiter les risques et conséquences de la phase de compromission
- Limiter les possibilités de l'attaquant après l'intrusion
- Éléments fondamentaux nécessaires sur un système
- Éléments à supprimer d'un environnement de production

Jour 5 - Après-midi

La sécurité du chemin d'exécution - Suite
Le durcissement du système d'exploitation

- Gestion de l'authentification et des comptes utilisateurs
- Politique de mot de passe et écueils courants
- Politique de gestion des pare-feux
- Configuration système
- Mise en place et surveillance du système de fichiers
- Limiter les droits d'exécution

Le durcissement des applications serveurs

- Bonne pratique d'installation et de maintenance
- L'exposition de données sensibles
- Les défauts de paramétrage de sécurité
- Vérifier les failles connues sur les composants utilisés
- Le manque d'historique et de monitoring
- Les serveurs Web
- Le langage côté serveur
- Les bases de données

Travaux pratiques : Sécurisation de l'environnement de formation

Modalités d'évaluation des acquis

En cours de formation, par des études de cas ou des travaux pratiques. En fin de formation, par un questionnaire d'auto-évaluation.

{{< formation-require >}}

{{< formation-seealso >}}