Sécurité des serveurs et des applications Web

Référence : **SECWEB5** Durée : **5 jours** Certification : **Aucune**

CONNAISSANCES PREALABLES

• 1-Expérience en programmation, idéalement en développement Web. • 2-Connaissance de base en cybersécurité.

PROFIL DES STAGIAIRES

• 1-Personnes ayant un profil technique souhaitant acquérir les connaissances suffisantes pour sécuriser leurs développements Web. • 2-DevSecOps. • 3-Programmeurs. • 4-Développeurs. • 5-Architectes. • 6-Chef de projet. • 7-Consultant cybersécurité.

OBJECTIFS

• Éduquer vos équipes de développement aux risques et aux enjeux de la sécurité applicative en mettant en application l'ensemble des points clés du standard OWASP. • Être en mesure d'augmenter rapidement la qualité et la sécurité de leurs développements de façon pertinente et efficace.

CERTIFICATION PREPAREE

Aucune

METHODES PEDAGOGIQUES

- Mise à disposition d'un poste de travail par stagiaire
- Remise d'une documentation pédagogique papier ou numérique pendant le stage
- La formation est constituée d'apports théoriques, d'exercices pratiques, de réflexions et de retours d'expérience
- Le suivi de cette formation donne lieu à la signature d'une feuille d'émargement

FORMATEUR

Consultant-Formateur expert Sécurité défensive

METHODE D'EVALUATION DES ACQUIS

- Auto-évaluation des acquis par le stagiaire via un questionnaire
- Attestation de fin de stage adressée avec la facture

CONTENU DU COURS

Introduction aux risques et aux enjeux de la sécurité applicative

- Quelques idées reçues
- La couche applicative – Une surface d'attaque de choix
- Prise en main de l'environnement de travaux pratiques

Rappels sur les technologies web

- Encodages (URL, HTML, Base64)
- HTTP / HTTPS
- Utilisation d'un proxy Web pour intercepter, analyser et modifier les échanges HTTP(S)

Introduction aux techniques d'attaque et aux mécanismes de défense

- Présentation de l'OWASP (guides, outils et TOP 10 de l'OWASP Web)
- Attaques et mécanismes de défense
- Utilisation du scanner de vulnérabilité OWASP ZAP

La phase de reconnaissance utilisée avant d'attaquer une application

- Axes de fuite d'informations techniques
- Utilisation d'outils de "Crawling" et d'outils de collecte d'information

Le mécanisme de gestion de l'authentification (attaque et défense)

- Mécanismes d'authentification les plus rencontrés
- Failles / Attaques qui ciblent le mécanisme d'authentification
- Moyens de défense permettant de sécuriser le mécanisme d'authentification
- "Brute-force" d'un mécanisme d'authentification
- Interception de données en transit (Sniffing)

Le mécanisme de gestion de la session (attaque et défense)

- Rappel autour des sessions
- Failles / Attaques qui ciblent le mécanisme de gestion de la session
- Moyens de défense permettant de sécuriser le mécanisme de gestion de la session
- Exploitation de la faille permettant la fixation de session

Le mécanisme de gestion des autorisations (attaque et défense)

- Droits horizontaux et droits verticaux
- Failles / Attaques qui ciblent le mécanisme de gestion des autorisations
- Attaques de type Cross-Site Request Forgery (CSRF)
- Attaques de type File Inclusion (RFI / LFI) et Path Traversal
- Moyens de défense permettant de sécuriser le mécanisme de gestion des autorisations
- Exploitation d'une faille de type Path Traversal

La gestion des entrées utilisateurs (injection de code)

- Les différents types d'attaques permettant l'injection de code (SQL, HQL, LDAP, commandes, etc.) et le principe général de ce type d'attaque
- Moyens de défense permettant de sécuriser vos entrées utilisateurs
- Exploitation de failles de type Injection SQL manuellement et de façon automatique (via l'utilisation d'un outil)

Les attaques ciblant les autres utilisateurs (attaque de type cross-site)

- Attaques de type Cross-Site Scripting (XSS)
- Le cas des clients riches JavaScript (AngularJS, Backbone, Ember, NodeJS, ReactJS, etc.)
- Moyens de défense permettant de sécuriser la navigation de vos utilisateurs et de se protéger contre l'injection de code HTML / JavaScript
- Mise en œuvre de différents scénarios d'attaques reposant sur l'exploitation d'une faille de type Cross-Site Scripting (modification de l'affichage, vol de session, redirection arbitraire, etc.)

Sécurité de la journalisation, de la gestion des erreurs et des exceptions

- Principe et enjeux de la journalisation des évènements de sécurité
- Stockage d'informations sensibles dans les journaux et attaques de type injection de "logs"
- Principe et enjeux de la gestion des erreurs et des exceptions
- Axes de prévention et bonnes pratiques dans le domaine

Sécurité des services web (Front end JavaScript, API SOAP & REST)

- Front-end à base de clients riches JavaScript
- Les failles des clients riches JavaScript
- Services Web SOAP et REST
- Failles des Services Web SOAP et des Services REST
- Axes de prévention et bonnes pratiques dans le domaine