Développez vos talents
Certifiez vos compétences

Sécurité défensive

Sécurité applicative avec PHP

3 jours (21h00) | ★★★★★ 5/5 | SEC-PHP | Évaluation qualitative de fin de stage | Formation délivrée en présentiel ou distanciel [1]

Formations Informatique › Cybersécurité › Sécurité défensive

À l'issue de ce stage vous serez capable de :

- Acquérir des compétences en programmation
- Sécuriser efficacement un serveur Web / une application.

Niveau requis

Avoir des connaissances généralistes en programmation Web.

Public concerné

Pentesters et développeurs.

Cette formation :

- Est animée par un consultant-formateur dont les compétences techniques, professionnelles et pédagogiques ont été validées par des diplômes et/ou testées et approuvées par l'éditeur et/ou par M2i Formation
- Bénéficie d'un suivi de son exécution par une feuille de présence émargée par demi-journée par les stagiaires et le formateur.

(1) Modalité et moyens pédagogique :

Formation délivrée en présentiel ou distanciel * (e-learning, classe virtuelle, présentiel à distance). Le formateur alterne entre méthodes ** démonstrative, interrogative et active (via des travaux pratiques et/ou des mises en situation). La validation des acquis peut se faire via des études de cas, des quiz et/ou une certification.

Les moyens pédagogiques mis en oeuvre (variables suivant les formations) sont : ordinateurs Mac ou PC (sauf pour les cours de l'offre Management), connexion internet fibre, tableau blanc ou paperboard, vidéoprojecteur ou écran tactile interactif (pour le distanciel). Environnements de formation installés sur les postes de travail ou en ligne. Supports de cours et exercices.

* Nous consulter pour la faisabilité en distanciel. ** Ratio variable selon le cours suivi.

Programme

Jour 1

Introduction

- Panorama de la sécurité Web
- Les normes et lois
- Les référentiels
- Les groupes de réflexions
- Evolution du langage PHP

Protocole HTTP avec PHP

- Principes d'une application PHP trois tiers
- Requête Ajax
- La fonction header()
- http_response_code()
- Les méthodes HTTP via le module cURL pour PHP

Exemple de travaux pratiques (à titre indicatif)

- *Ouverture sur Burp Suite*

Top 10 OWASP 2017 (basé sur Burp Suite)

- Mise en place du Lab
- Introduction au Top 10 OWASP, Top 25 SANS et Veracode
- Les différentes injections (SQL, LDAP, code...)
- Authentification
  - Exposition de données sensibles

Exemples de travaux pratiques (à titre indicatif)

- *Injection SQL et injection de code*
- *Session hijacking (MITM proxy), brute force (cewl + Cupp.py)*
- *Burp Spider, Shodan, dorks, dirbuster, inspection de code et GIT*

Jour 2

Top 10 OWASP 2017 (basé sur Burp Suite) - Suite

- XXE (XML eXternal Entity)
  - Sécurisation des accès
- Mauvaise configuration de sécurité
- Cross-Site Scripting (XSS)
  - Stored
  - Reflected
  - Dom based
- Désérialisation non sécurisée
- Composants vulnérables
- Logging et monitoring

Exemples de travaux pratiques (à titre indicatif)

- *Challenge XXE*
- *Elévation de privilèges (bypass CORS et cookie tamering)*
- *Vulnérabilité SSRF (Server Side Request Forgery)*
- *Defacing avec XSS*
- *Vol de cookies via CSRF*
- *Elévation de privilège via cookie sérialisé*

- *Scan de vulnérabilité (WPScan, Nikto, Openvas, NMAP) et framework offensif (Metasploit)*
- *DoS d'une application*
- *Revue et démonstration sur une faille de sécurité PHP : Drupal remote execution*

Jour 3

Hardening d'une application PHP

- Les forces et faiblesses du langage PHP
- Sécuriser une authentification (captcha et anti-bruteforce PHP)
- Gestion des mots de passe (password_hash / password_verify)
- Renforcement du système de sessions PHP
- Contrôle d'accès (de l'intérêt de la Programmation Orientée Objet en PHP)
- Validation des entrées (filter_var / strip_tags)
- Encodage des sorties (htmlentities / htmlspecialchars)
- Sécuriser un upload de fichier en PHP
- Comment générer des tokens anti-CSRF (Cross Site Request Forgery) ?
- Management des logs (php.ini)

Exemple de travaux pratiques (à titre indicatif)

- *Création d'un portail d'authentification sécurisé en PHP*

Hardening client / serveur par la pratique

- PHPINFO() / PHPSECINFO()
- php.ini
- CSP (Content Security Policy)
- SOP / CORS
- Tests unitaires PHP
- Analyse statique / dynamique avec RIPS
- Durcissement des trames en PHP
- Ouverture avec l'OWASP testing guide, ASVS (Application Security Verification Standard)

Modalités d'évaluation des acquis

L'évaluation des acquis se fait :

- En cours de formation, par des études de cas ou des travaux pratiques
- Et, en fin de formation, par un questionnaire d'auto-évaluation ou une certification (M2i ou éditeur)