Rapport Projet : Élaboration de démonstrateurs d'attaques web OWASP 2021

Date : 18/11/2024

Binôme : Salma El Bougrini et Antoine Pennamen

Encadrants : Christophe Kiennert et Gregory Blanc

Projet

Ce projet se propose d'étudier les vulnérabilités mises en avant dans le dernier Top Ten (datant de 2021) et d'implémenter un certain nombre de démonstrateurs visant à illustrer l'exploitation de certaines de ces vulnérabilités via des attaques sur des applications Web réalistes.

Objectifs

Il faudra dans un premier temps s'approprier les concepts qui rendent l'attaque possible (menaces et vulnérabilités) puis proposer une implémentation dans des conditions réalistes (environnement récent, fonctionnement à distance). Enfin, il faudra proposer un correctif pour l'application afin de rendre l'attaque inopérante.

Avancement

1. **Étude et restitution de la compréhension du top10 des vulnérabilités OWASP**
    1. Broken Access Control
    2. Cryptographic Failures
    3. Injection
    4. Insecure Design
    5. Security Misconfiguration
    6. Vulnerable and Outdated Components
    7. Identification and Authentication Failures
    8. Software and Data Integrity Failures
    9. Security Logging and Monitoring Failures
    10. Server-Side Request Forgery

2. **Implémentation d'un site internet**

Deux versions du site internet sont développées en parallèle, une pour tester les vulnérabilités et une pour la mise en place des mitigations :

- Back-end en php, front-end en css, html
- Base de données fonctionnelle MySQL
- Code accessible sur Github : https://github.com/pennamenantoine/PFE

3. **Étude et développement de quelques vulnérabilités :**

Nous avons commencé par les injections.

XSS :

- Dans le cadre des injections (xss) : Détourner le but de l'application web (ne renvoie plus seulement des pages mais exécute du code). Reflected et Stored XSS ont été étudiées et sont actuellement implémentées sur notre site ou en cours d'implémentation.

SQLi :

- Les injections SQL sont possibles dans le formulaire de connexion (OR statement, bypass authentication with comments).

Les Contre-Mesures mises en place :

- Modification du code vulnérable en utilisant les good practices de l'OWASP.
- Utilisation de la fonction php htmlspecialchars pour échapper les entrées utilisateur.
- Reste à vérifier si seulement 5 caractères sont échappés avec la fonction htmlspecialchars (< > « ‘ &). Est-il possible de contourner cette fonction en utilisant un autre moyen que les caractères qui sont échappés ?

Prochaines Étapes :

- Continuer l'étude des vulnérabilités.
- Choisir les exemples de vulnérabilités à implémenter sur notre application.
- Définir des scénarios d'enchainement de l'exploitation des vulnérabilités.