Projet - Application web vulnérable

Objectifs

- Comprendre les vulnérabilités courantes en web et pouvoir les implémenter volontairement
- Identifier et exploiter des vulnérabilités web
- Mettre en oeuvre des correctifs afin de corriger les vulnérabilités
- Être capable d'effectuer un audit technique sur une application web
- Travail en équipe

Énoncé

Le but de ce projet est de développer une application web dans le langage de son choix.
L'application web doit obligatoirement faire appel à une base de données (peu importe le SGBD).

Il faut ensuite choisir 5 vulnérabilités différentes (hors brute force) et les implémenter volontairement au niveau de l'application web.

Vous pouvez soit choisir des vulnérabilités vues en cours ou en sélectionner d'autres (liées à des API par exemple avec du mass assignment)

Vous pouvez implémenter les fonctionnalités que vous souhaitez pour vous permettre d'y insérer les vulnérabilités souhaitées. Attention cependant, l'application doit être réaliste. Vous ne pouvez pas simplement créer 5 pages basiques avec juste chacune la vulnérabilité implémentée (comme on retrouve dans DVWA).

Une fois les vulnérabilités volontairement implémentées, les exploiter pour montrer la bonne compréhension de celles-ci et les documenter dans le rapport (voir ensuite).

Fournir ensuite une autre application web qui se base sur la première (avec les vulnérabilités) mais qui patch ces vulnérabilités précédemment mises en place afin qu'elle soit totalement sécurisée.

Groupe de 4 personnes max. Projet à rendre avant le dimanche 8 décembre à 23h59.

Des séances sont prévues ensemble (2 et 3 décembre) pour que vous puissiez me poser des questions et que je vous aide au niveau de la mise en place/détection/remédiation des vulnérabilités. Vous pouvez cependant prendre de l'avance de votre côté.

Livrables

Au final, vous devrez rendre trois éléments :

- Une application web vulnérable
- Une application web sécurisée
- Un rapport

Le rapport doit permettre de savoir quelles sont les vulnérabilités mises en place, un PoC (Proof of Concept) de leur exploitation (avec des captures + explications), les remédiations associées et un "contre-audit" pour prouver qu'elle n'est plus exploitable.

Attention cependant, si d'autres vulnérabilités sont trouvées (lors de l'évaluation) ou que le patch n'est pas suffisant et qu'il peut être bypass, vous perdrez des points.

Les éléments devront être présentés comme suit :

- Un gros dossier projet avec tout le contenu à l'intérieur :
  - Un dossier `vulnerable` qui contient le site web vulnérable
  - Un dossier `secure` qui contient le site web sécurisé
  - Le rapport
  - Un README qui explique comment installer les applications

Le projet doit être rendu **AU MAXIMUM le 8 décembre 2024 à 23h59** !
Tout retard sera sanctionné au niveau de la note.

N'oubliez pas de mettre le nom de toutes les personnes du groupe sur votre rapport.

Critères d'évaluation

Vous serez évalué sur différents aspects du projet :

- Compréhension des vulnérabilités et capacité à les identifier / reproduire
- Qualité et cohérence de l'application développée
- Exhaustivité, explications et structure du rapport
- Documentation du code (votre code doit être lisible et documenté)

Conseils - Détails

Voici quelques conseils pour vous permettre de mener à bien ce projet :

- Utilisation d'un docker compose pour faciliter l'installation et la correction des projets
- Rapport sous format PDF de préférence
- Partage du dossier projet sur un Github ou un Drive (Google, DropBox, Mega...) : attention cependant à bien configurer les droits d'accès pour que je puisse y accéder via un lien par exemple pour l'évaluation et la correction.
- Pour le rapport, imagez avec des captures d'écran et expliquez vos raisonnements (pourquoi vous avez fait telle ou telle chose).
- Pour le code, pensez à bien le documenter/commenter pour faciliter la correction.
- Une application non fonctionnelle retirera forcément des points.
- Pour la démonstration de l'exploitation des vulnérabilités, vous pouvez utiliser des outils vus en cours (Burp Suite, ffuf, etc...) ou bien les exploiter manuellement. Dans tous les cas, il faut expliquer pourquoi telle ou telle chose fonctionne, comment et détailler le processus d'exploitation (ça ne sert à rien de juste lancer SQLmap pour prouver qu'il y a une SQLi s'il n'y a aucune explication de la faille derrière).

Je reste disponible même en dehors des cours si vous avez des questions : quentin.simier@synoslabs.com