Rapport de Sécurité : Formulaire de Contact Sécurisé

Nour Ben Zid

13 mai 2025

1. Introduction

Ce rapport présente les mesures de sécurité mises en œuvre pour le développement d'un formulaire de contact sécurisé. L'objectif du projet est d'assurer la confidentialité et l'intégrité des données des utilisateurs, tout en se prémunissant contre les attaques courantes du web. Le principe de *Security by Design* a guidé toutes les étapes du projet.

2. Structure du Projet

Le projet a été conçu selon une architecture modulaire respectant les bonnes pratiques de sécurité.

- **config**/ : Certificats SSL et configurations serveur.
- **logs**/ : Journaux d'accès et messages utilisateurs.
- **public**/ : Fichiers statiques (HTML, CSS, JS).
- **data**/ : Données utilisateurs et messages.
- **src**/ : Code source de l'application (Node.js, routes, logique).

3. Mise en œuvre des mesures de sécurité

3.1 Chiffrement via HTTPS

Un certificat SSL a été généré avec `OpenSSL` pour activer HTTPS. Cela assure une protection contre les attaques de type Man-in-the-Middle (MITM).

3.2 Intégration de Google reCAPTCHA

Un reCAPTCHA v2 a été intégré afin d'éviter les soumissions automatisées. La vérification est assurée côté serveur via la clé secrète Google.

3.3 Hachage des mots de passe avec `bcrypt`

Les mots de passe sont hachés avant enregistrement avec `bcrypt`, stockés dans le fichier `users.json` pour garantir la sécurité en cas de fuite.

3.4 Prévention des attaques XSS et SQL Injection

Les entrées des utilisateurs (nom, email, message) sont filtrées et échappées afin d'éviter l'injection de code malveillant. Les caractères spéciaux sont désactivés pour empêcher l'exécution de scripts.

3.5 Cookies sécurisés

Les cookies sont configurés avec les attributs `HttpOnly` et `Secure`, empêchant leur accès par JavaScript et garantissant une transmission uniquement sur HTTPS.

4. Journaux de Sécurité

Des fichiers de logs assurent la traçabilité des activités :
— `access.log` : Requêtes HTTP reçues.
— `messages.log` : Messages soumis par les utilisateurs.
Les messages sont également sauvegardés hachés dans `messages.json`.

5. Tests de sécurité

5.1 Tests anti-injection SQL et XSS

Des injections telles que `' OR '1'='1` et `<script>alert('XSS')</script>` ont été bloquées avec succès.

5.2 Vérification des cookies

Via les outils développeur du navigateur, il a été confirmé que les attributs de sécurité des cookies sont correctement définis.

6. Captures d'écran

— Page de connexion utilisateur (`nour@gmail.com` / mot de passe : `motdepasse123`).
— Tableau de bord administrateur.
(Insérez ici les images avec `\includegraphics`)

7. Conclusion

Le formulaire sécurisé développé garantit la confidentialité et la protection des données collectées grâce à :
— La validation des entrées utilisateurs.
— L'utilisation de HTTPS, `bcrypt`, et cookies sécurisés.
— Des tests complets contre les attaques courantes.
Ce rapport démontre que le projet est conforme aux exigences de sécurité modernes pour les applications web.