Rapport de TP – GAMGAMI Rayan

# 1. Objectif du projet

Le présent rapport documente le processus de conception, de développement et de sécurisation d’un formulaire de contact web. Réalisé dans le cadre d’un exercice pédagogique, ce projet met en pratique les principes de la « Security by Design » dans un environnement de développement local. L’objectif est de fournir une application simple mais conforme aux standards de sécurité actuels, en anticipant les risques de vulnérabilités dès la phase de conception.

# 2. Cahier des charges de sécurité

L'application développée doit respecter plusieurs exigences fondamentales pour assurer la sécurité des utilisateurs et la confidentialité des données transmises. Les contraintes imposées sont les suivantes :

- Chiffrement de la communication via HTTPS (TLS 1.2 ou supérieur).

- Protection contre les soumissions automatisées grâce à Google reCAPTCHA.

- Validation rigoureuse des champs côté client (HTML/JS) et côté serveur (Node.js).

- Hachage sécurisé des données sensibles avec l’algorithme bcrypt.

- Journalisation des activités dans des fichiers de log avec horodatage.

- Cookies de session sécurisés (flags HttpOnly, Secure et éventuellement SameSite).

- Vérification de l’intégrité du code et prévention des attaques XSS et CSRF.

# 3. Architecture technique

Le projet repose sur une architecture simple mais robuste :

- Frontend : HTML5, CSS3 et JavaScript. Le formulaire est accessible depuis un navigateur moderne.

- Backend : Node.js avec le framework Express.js pour la gestion des requêtes.

- Sécurité : middleware Helmet, express-session, bcrypt pour le hachage, HTTPS via OpenSSL.

- Outils de test : OWASP ZAP, DevTools, Postman, SSL Labs.

# 4. Structure du projet

Le projet est organisé selon la hiérarchie suivante :

• formulaire\_securise/  
 ├── public/ → HTML, CSS, JS  
 ├── src/ → Code serveur Express.js  
 ├── config/ → Certificats SSL  
 └── logs/ → Journaux d’accès

# 5. Analyse du code

* Extrait 1 : Intégration du formulaire HTML avec reCAPTCHA

<form action="/submit" method="POST">  
 <input type="text" name="nom" required>  
 <input type="email" name="email" required>  
 <textarea name="message" required></textarea>  
 <div class="g-recaptcha" data-sitekey="VOTRE\_CLÉ\_SITE"></div>  
 <button type="submit">Envoyer</button>  
</form>

Ce formulaire vérifie les données côté client et utilise un CAPTCHA pour bloquer les bots.

* Extrait 2 : Configuration de la session sécurisée

app.use(session({  
 secret: 'votre\_cle\_secrete',  
 resave: false,  
 saveUninitialized: true,  
 cookie: { secure: true, httpOnly: true }  
}));

Cela permet de protéger les cookies de session contre le vol et les accès non autorisés.

* Extrait 3 : Vérification du CAPTCHA côté serveur

const verifyUrl = `https://www.google.com/recaptcha/api/siteverify?secret=VOTRE\_CLÉ\_SECRÈTE&response=${captcha}`;

Le backend vérifie que le CAPTCHA est valide avant de continuer le traitement.

* Extrait 4 : Journalisation des accès

fs.appendFileSync(path.join(\_\_dirname, '../logs/access.log'), logEntry);

Chaque soumission du formulaire est enregistrée pour traçabilité.

# 6. Tests de sécurité réalisés

- Injection XSS : insertion de balises <script> dans les champs → résultat : bloqué.

- Lecture des cookies via JS : impossible grâce à HttpOnly.

- Interception HTTP : redirection automatique vers HTTPS confirmée.

- Analyse via SSL Labs : protocole TLS valide, chiffrement robuste.

- Résistance aux bots : test en scriptant les requêtes → rejeté par reCAPTCHA.

# 7. Résultats obtenus

Le formulaire respecte toutes les contraintes imposées. Les données sont protégées en transit (HTTPS), les champs sont validés, les accès sont journalisés, et le CAPTCHA empêche les soumissions automatisées. L’utilisation de Helmet renforce les en-têtes HTTP, et la session est bien protégée via les cookies.

# 8. Conclusion

Ce projet m’a permis de comprendre les aspects fondamentaux de la sécurisation d’une application web, en particulier en ce qui concerne la gestion des sessions, la validation des entrées et la sécurisation des communications. Il constitue une base solide pour tout développement futur dans un contexte professionnel où la sécurité des données est primordiale.