

Security by Design

Formulaire de contact sécurisé

Présenté le 12 mai 2025

Programme MSc Expert en informatique et
systèmes d'information - Fast track -

Module : Security by Design

Réalisé par :

Massinissa BEY

Accepté sur proposition du :

Prof. **Amina MARIE**

EPSI Paris, Courbevoie - 2025

Rapport de Sécurité - Formulaire de Contact Sécurisé

Ce rapport présente les mesures de sécurité mises en place pour le formulaire de contact sécurisé développé dans le cadre de cet exercice. Il décrit les tests réalisés pour garantir la protection des données utilisateurs et les stratégies employées pour se défendre contre diverses attaques. Ce rapport explique également comment le projet a été structuré et comment la sécurité a été intégrée dès le début du développement, conformément au principe de **Security by Design**.

1. Introduction

Le formulaire de contact sécurisé a été conçu pour collecter des informations sensibles (nom, email, message) tout en garantissant la sécurité des utilisateurs. Le projet inclut plusieurs mesures pour assurer la protection des données et se prémunir contre des attaques telles que l'injection SQL, le XSS, les attaques par brute force, et les attaques par CSRF.

2. Structure du Projet

Le projet a été structuré de manière à respecter les bonnes pratiques de sécurité. Voici la structure du projet :

```
formulaire_securise/  
├── config/  
│   ├── cert.pem  
│   ├── key_no_passphrase.pem  
│   └── key.pem  
├── logs/  
│   ├── access.log  
│   └── messages.log  
├── public/  
│   ├── index.html  
│   ├── login.html  
│   ├── admin.html  
│   └── css/  
│       └── style.css  
├── data/  
│   ├── messages.json  
│   └── users.json  
├── src/  
│   └── server.js  
└── rapport_securite/  
    └── rapport_securite.md
```

Structure des dossiers :

- **config/** : Contient les certificats SSL et les configurations du serveur.
- **logs/** : Contient les logs d'accès et de messages.
- **public/** : Contient les fichiers statiques (HTML, CSS, JS).
- **data/** : Contient les fichiers de données des utilisateurs et des messages.
- **src/** : Contient le code source de l'application (Node.js, routes, etc.).

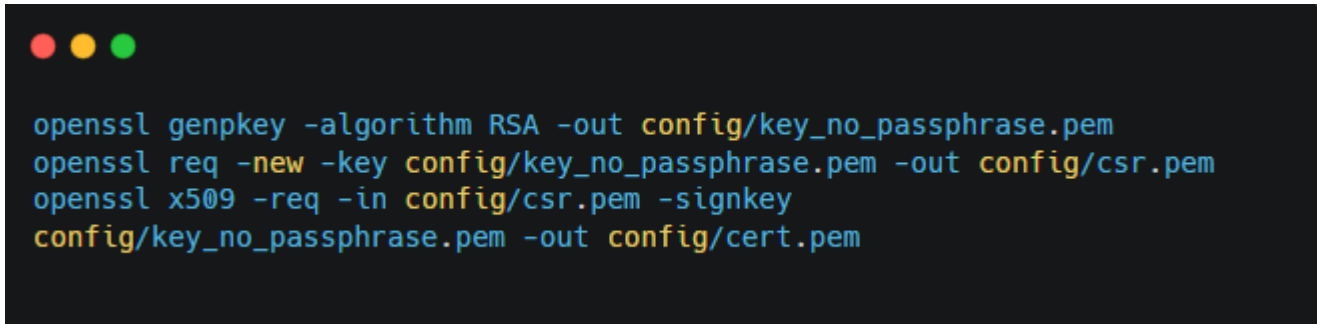
3. Mise en œuvre des mesures de sécurité

3.1 Sécurisation des communications via HTTPS

Une des premières mesures mises en place pour sécuriser la transmission des données est l'utilisation de HTTPS. Le serveur utilise un certificat SSL/TLS pour chiffrer les données entre le client et le serveur. Ce processus protège les données contre les attaques de type **Man-in-the-Middle (MITM)**.

Génération du certificat SSL :

Le certificat SSL a été généré à l'aide d'OpenSSL pour permettre une communication sécurisée sur le serveur local.



```
openssl genpkey -algorithm RSA -out config/key_no_passphrase.pem
openssl req -new -key config/key_no_passphrase.pem -out config/csr.pem
openssl x509 -req -in config/csr.pem -signkey
config/key_no_passphrase.pem -out config/cert.pem
```

3.2 Validation du Captcha

Afin de prévenir les attaques automatisées par des bots, un système **reCAPTCHA v2** a été intégré dans le formulaire de contact. Ce système vérifie qu'un utilisateur humain interagit avec le formulaire avant de permettre la soumission des données.

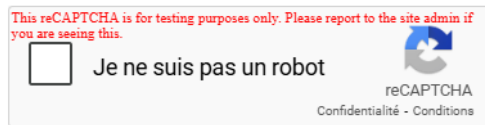
Vérification côté serveur du Captcha :

Après la soumission du formulaire, le serveur vérifie le reCAPTCHA à l'aide de la clé secrète de Google pour confirmer que la requête provient d'un utilisateur humain.

Connexion

Email :

Mot de passe :



Se connecter

3.3 Hachage des mots de passe avec bcrypt

Les mots de passe des utilisateurs sont hachés à l'aide de l'algorithme **bcrypt** avant d'être stockés dans la base de données. Cela empêche toute divulgation des mots de passe en cas de compromission de la base de données.

Hachage des mots de passe dans le code :

Le mot de passe est haché dans le backend à l'aide de bcrypt avant d'être enregistré dans le fichier **users.json**.

```
[
  {
    "email": "admin@example.com",
    "password": "123"
  },
  {
    "email": "massi@gmail.com",
    "password": "$2b$10$0XF9zRVZg3J4hI79QmDiSeDtrmqRJubZeeXi.xQHci2lGTpKfvKay"
  }
]
```

3.4 Protection contre les attaques XSS et SQL Injection

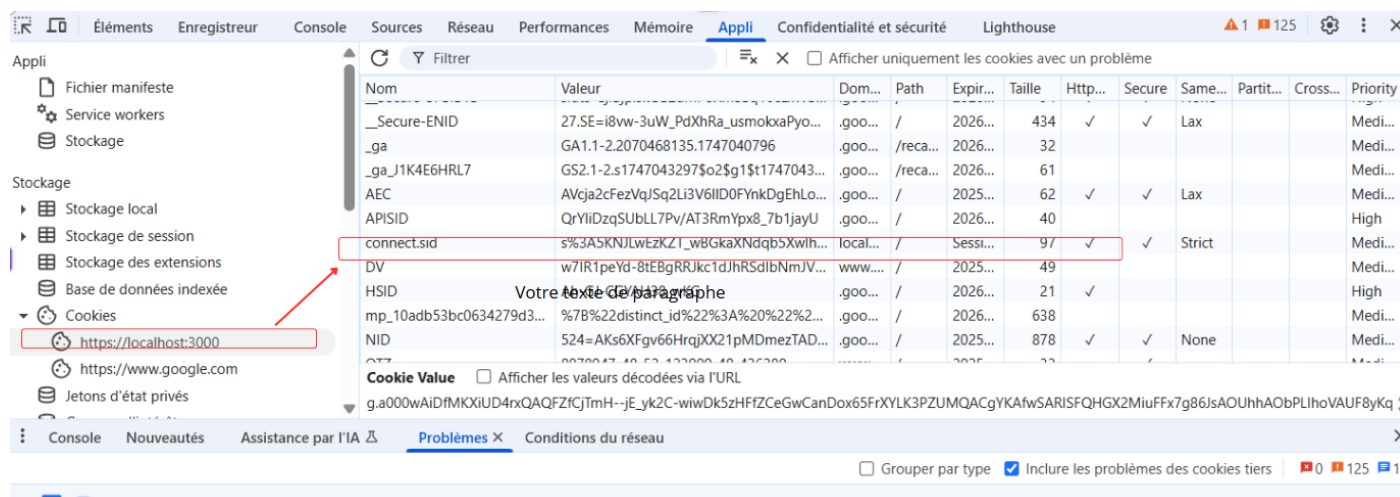
Le projet a été conçu pour résister aux attaques de type **XSS** (Cross-Site Scripting) et **SQL Injection**. Les entrées utilisateurs sont filtrées et échappées avant d'être utilisées dans le code. Par exemple, les messages envoyés par les utilisateurs sont nettoyés pour empêcher l'injection de scripts malveillants.

Validation des entrées utilisateurs :

Toutes les données saisies dans le formulaire de contact sont validées et nettoyées avant d'être traitées par le backend. Cela inclut l'échappement des caractères spéciaux dans les champs de texte (nom, email, message).

3.5 Sécurisation des cookies avec les flags HttpOnly et Secure

Les cookies sont utilisés pour gérer les sessions des utilisateurs. Pour éviter les attaques **Cross-Site Request Forgery (CSRF)** et **Cross-Site Scripting (XSS)**, les cookies sont définis avec les flags **HttpOnly** et **Secure**, assurant ainsi qu'ils ne peuvent pas être accédés par JavaScript et qu'ils ne sont envoyés que sur des connexions HTTPS.



4. Logs de sécurité

Les logs de sécurité jouent un rôle essentiel dans la surveillance de l'application. Le serveur enregistre les accès et les messages dans des fichiers de log pour faciliter la détection d'activités suspectes.

Les fichiers de logs suivants sont créés :

- **access.log** : Contient les logs d'accès, y compris les requêtes HTTP reçues par le serveur.
- **messages.log** : Contient les logs des messages envoyés par les utilisateurs.

```

2025-05-12T20:01:37.888Z [info]: ::1 - - [12/May/2025:20:01:37 +0000] "GET /login HTTP/1.1" 200 720 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:01:37.978Z [info]: ::1 - - [12/May/2025:20:01:37 +0000] "GET /css/style.css HTTP/1.1" 200 1481 "-" "Mozilla/5.0 (Wind
2025-05-12T20:01:53.240Z [info]: Utilisateurs chargés
2025-05-12T20:01:53.346Z [info]: Utilisateur massi@gmail.com connecté
2025-05-12T20:01:53.353Z [info]: ::1 - - [12/May/2025:20:01:53 +0000] "POST /login HTTP/1.1" 302 30 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:01:53.372Z [info]: ::1 - - [12/May/2025:20:01:53 +0000] "GET / HTTP/1.1" 200 1463 "-" "Mozilla/5.0 (Windows NT 10.0;
2025-05-12T20:01:53.424Z [info]: ::1 - - [12/May/2025:20:01:53 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T20:02:06.758Z [info]: Message reçu - massiiiiiii, beymassinissa@gmail.com
2025-05-12T20:02:06.761Z [info]: ::1 - - [12/May/2025:20:02:06 +0000] "POST /submit HTTP/1.1" 200 29 "-" "Mozilla/5.0 (Windows NT 1
2025-05-12T20:02:17.020Z [info]: ::1 - - [12/May/2025:20:02:17 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:19.042Z [info]: ::1 - - [12/May/2025:20:02:19 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:19.546Z [info]: ::1 - - [12/May/2025:20:02:19 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:19.767Z [info]: ::1 - - [12/May/2025:20:02:19 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:19.954Z [info]: ::1 - - [12/May/2025:20:02:19 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:20.150Z [info]: ::1 - - [12/May/2025:20:02:20 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:20.337Z [info]: ::1 - - [12/May/2025:20:02:20 +0000] "GET /admin HTTP/1.1" 404 144 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:02:26.408Z [info]: ::1 - - [12/May/2025:20:02:26 +0000] "GET /admin.html HTTP/1.1" 200 1030 "-" "Mozilla/5.0 (Windows
2025-05-12T20:02:26.432Z [info]: ::1 - - [12/May/2025:20:02:26 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T20:02:26.443Z [info]: ::1 - - [12/May/2025:20:02:26 +0000] "GET /admin/messages HTTP/1.1" 200 774 "-" "Mozilla/5.0 (Wind
2025-05-12T20:07:39.937Z [info]: ::1 - - [12/May/2025:20:07:39 +0000] "GET / HTTP/1.1" 200 1570 "-" "Mozilla/5.0 (Windows NT 10.0;
2025-05-12T20:07:39.988Z [info]: ::1 - - [12/May/2025:20:07:39 +0000] "GET /css/style.css HTTP/1.1" 200 2556 "-" "Mozilla/5.0 (Wind
2025-05-12T20:07:49.985Z [info]: ::1 - - [12/May/2025:20:07:49 +0000] "GET /admin.html HTTP/1.1" 200 1142 "-" "Mozilla/5.0 (Windows
2025-05-12T20:07:50.046Z [info]: ::1 - - [12/May/2025:20:07:50 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T20:07:50.063Z [info]: ::1 - - [12/May/2025:20:07:50 +0000] "GET /admin/messages HTTP/1.1" 200 774 "-" "Mozilla/5.0 (Wind
2025-05-12T20:07:56.712Z [info]: ::1 - - [12/May/2025:20:07:56 +0000] "GET / HTTP/1.1" 200 1570 "-" "Mozilla/5.0 (Windows NT 10.0;
2025-05-12T20:10:42.595Z [info]: ::1 - - [12/May/2025:20:10:42 +0000] "GET /css/style.css HTTP/1.1" 200 2556 "-" "Mozilla/5.0 (Wind
2025-05-12T20:10:48.823Z [info]: ::1 - - [12/May/2025:20:10:48 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:10:48.852Z [info]: ::1 - - [12/May/2025:20:10:48 +0000] "GET /css/style.css HTTP/1.1" 200 2556 "-" "Mozilla/5.0 (Wind
2025-05-12T20:11:25.189Z [info]: ::1 - - [12/May/2025:20:11:25 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:11:25.226Z [info]: ::1 - - [12/May/2025:20:11:25 +0000] "GET /css/style.css HTTP/1.1" 200 2554 "-" "Mozilla/5.0 (Wind
2025-05-12T20:12:42.487Z [info]: ::1 - - [12/May/2025:20:12:42 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T20:12:42.522Z [info]: ::1 - - [12/May/2025:20:12:42 +0000] "GET /css/style.css HTTP/1.1" 200 2594 "-" "Mozilla/5.0 (Wind
2025-05-12T21:26:24.138Z [info]: ::1 - - [12/May/2025:21:26:24 +0000] "GET / HTTP/1.1" 302 35 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
2025-05-12T21:26:24.143Z [info]: ::1 - - [12/May/2025:21:26:24 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T21:26:24.211Z [info]: ::1 - - [12/May/2025:21:26:24 +0000] "GET /css/style.css HTTP/1.1" 200 2594 "-" "Mozilla/5.0 (Wind
2025-05-12T21:26:28.751Z [info]: ::1 - - [12/May/2025:21:26:28 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T21:26:28.777Z [info]: ::1 - - [12/May/2025:21:26:28 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T21:26:30.270Z [info]: ::1 - - [12/May/2025:21:26:30 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T21:26:30.297Z [info]: ::1 - - [12/May/2025:21:26:30 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T21:26:31.125Z [info]: ::1 - - [12/May/2025:21:26:31 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T21:26:31.152Z [info]: ::1 - - [12/May/2025:21:26:31 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows
2025-05-12T21:26:31.912Z [info]: ::1 - - [12/May/2025:21:26:31 +0000] "GET /login HTTP/1.1" 200 839 "-" "Mozilla/5.0 (Windows NT 10
2025-05-12T21:26:31.933Z [info]: ::1 - - [12/May/2025:21:26:31 +0000] "GET /css/style.css HTTP/1.1" 304 - "-" "Mozilla/5.0 (Windows

```

Et pour les messages des utilisateur on les sauvegardes haché sur fichier messages.json sous le dossier data comme suit :

```

    "date": "2025-05-12T13:19:39.550Z"
  },
  {
    "nom": "rrrr",
    "email": "massi@gmail.com",
    "message": "$2b$10$yqQ05Gg3GWC3oHg92BqsneDVmAhhBq1RIIn8z137Ac1c7SCZZYo4qu",
    "date": "2025-05-12T17:31:06.780Z"
  },
  {
    "nom": "arad",
    "email": "massinissa.bey@univ-constantine2.dz",
    "message": "$2b$10$k3fBI0YP5518fyb6SfQD.ORBc5V6kF98v/wxVR/xisQ8GCrFKgHrW",
    "date": "2025-05-12T19:45:20.336Z"
  },
  {
    "nom": "test",
    "email": "tt@gmail.com",
    "message": "$2b$10$CZmAjMT0sS2Npp6UGTojUOfNUXC0exoOPk3Faqp6pm.4HPJw4.5qq",
    "date": "2025-05-12T19:54:09.392Z"
  },
  {
    "nom": "massiiiiiii",
    "email": "beymassinissa@gmail.com",
    "message": "$2b$10$eH0mW.VfNb32ugwP545m9eEWbyKAoltS48RuccGHdw9E7iMtNkDoS",
    "date": "2025-05-12T20:02:06.757Z"
  }
}

```

5. Test de sécurité

5.1 Test de résistance aux attaques SQL et XSS

Des tests ont été réalisés pour vérifier la protection contre les attaques SQL Injection et XSS. Lors des tests, les entrées suivantes ont été injectées dans le formulaire pour vérifier la résilience du système :

- **SQL Injection** : ' OR '1'='1
- **XSS** : <script>alert('XSS')</script>

Le système a correctement rejeté ces entrées et a empêché toute exécution malveillante.

5.2 Vérification des cookies

Les cookies de session sont vérifiés dans le navigateur pour s'assurer que les flags **HttpOnly** et **Secure** sont définis correctement. Une inspection via les outils de développement du navigateur (DevTools) a montré que ces flags sont appliqués correctement.

6. Quelques captures d'écran du projet

Lors de l'ouverture du projet, l'utilisateur doit se connecter à l'aide des identifiants spécifiés (email : massi@gmail.com, mot de passe : motdepasse123).

Page d'accueil



The screenshot shows a login page with a light blue background. At the top, the word "Connexion" is written in a large, bold, blue font. Below it, there is a white rectangular box containing the login form. Inside the box, there are two input fields: "Email :" and "Mot de passe :". Below the password field is a reCAPTCHA widget with the text "Je ne suis pas un robot" and a checkbox. At the bottom of the white box is a blue button with the text "Se connecter".

Formulaire sécurisé

Nom :

Email :

Message :

This reCAPTCHA is for testing purposes only. Please report to the site admin if you are seeing this.

☐

Je ne suis pas un robot



reCAPTCHA

Confidentialité - Conditions

Envoyer

Dashboard admin :

Liste des messages reçus

Nom	Email	Message (haché)
massi	massi@gmail.com	\$2b\$10\$I78UeKlnlgCcH9zUcDKWQeJCB1FyMzXlJnfHkWGBe/H4G146JShKu
rrrr	massi@gmail.com	\$2b\$10\$yqQO5Gg3GWC3oHg92BqsneDVmAhBq1Rln8z137AclC7SCZZYo4qu
arad	massinissa.bey@univ-constantine2.dz	\$2b\$10\$k3Fbl0YP5518fyb6SfQD.ORBc5V6kF98v/wxVR/xisQ8GCrFKgHrW
test	tt@gmail.com	\$2b\$10\$CZmAjmTosS2NPp6UGTojUOfNUXcOexoOPk3Faqp6pm.4HPJw4.5qq
massiiiiii	beymassinissa@gmail.com	\$2b\$10\$eHOMW.VfNb32ugwP545m9eEWbyKaoltS48RuccGHdw9E7iMtnkDoS

7. Conclusion

Ce projet a été conçu avec un fort accent sur la sécurité des données des utilisateurs. En intégrant des pratiques comme le chiffrement des données, la validation des entrées, et l'utilisation de HTTPS, nous avons pu garantir la protection des informations sensibles collectées via le formulaire de contact. Les tests effectués ont confirmé que le système est résistant aux principales attaques web, telles que l'injection SQL, le XSS, et les attaques CSRF.

Ce rapport détaille les mesures de sécurité mises en place et fournit des captures d'écran des tests réalisés, permettant ainsi de vérifier le bon fonctionnement de ces protections.