Rapport du TP de sécurité des applications

ESTOUP-STREIFF Chad
BARRAUD Matthieu
JEAN Maël
LAHELLEC Elouan
JAMME Maxence
MARTINEZ Gabriel

Sommaire

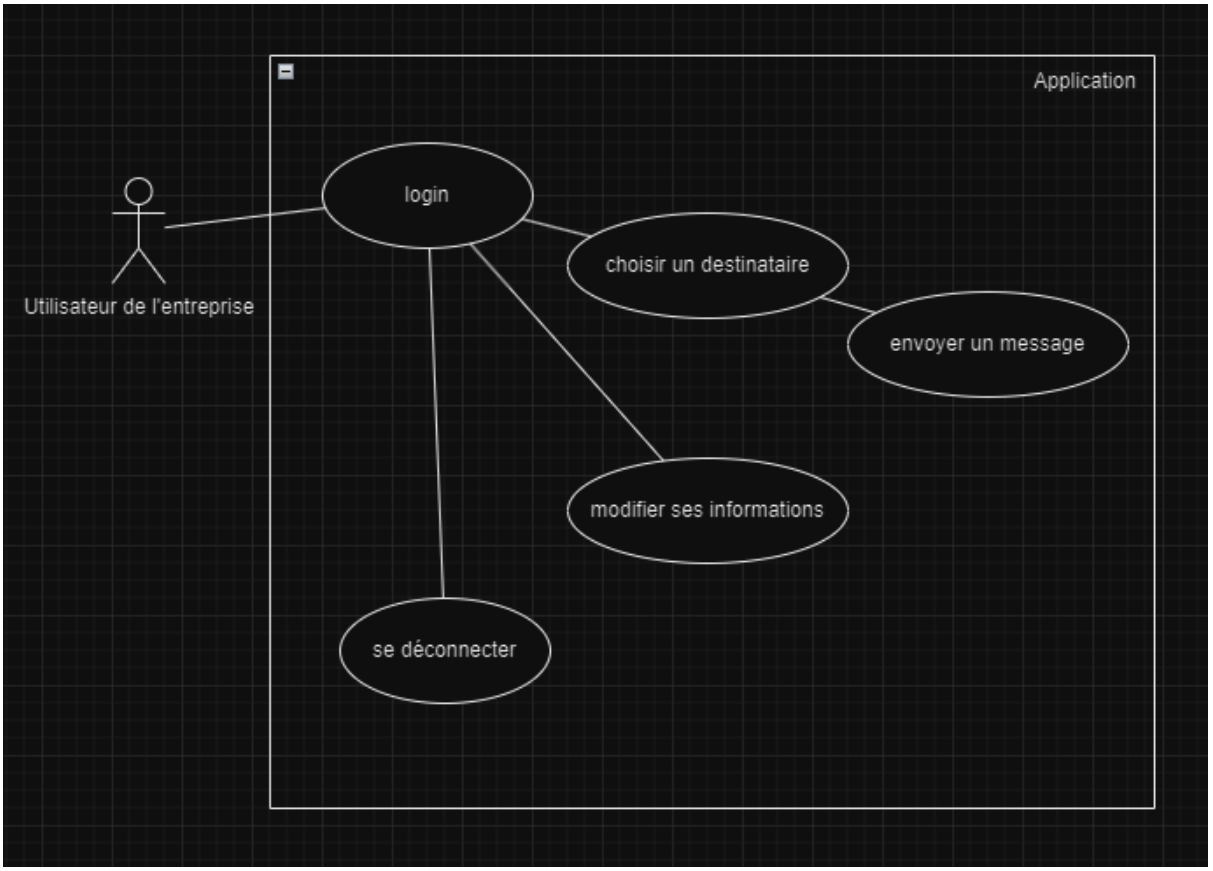
Introduction:

La communication sécurisée revêt une importance cruciale dans divers contextes, allant des échanges diplomatiques et militaires aux transactions commerciales sensibles. Un moyen essentiel pour assurer la confidentialité, l'intégrité et l'authenticité des informations échangées est l'utilisation de systèmes de chat privés sécurisés. Ces systèmes sont conçus pour garantir une communication confidentielle et protéger les données contre les menaces potentielles telles que l'interception non autorisée et la manipulation malveillante. C’est dans cette perspective que nous avons conçu un chat sécurisé.

Le fonctionnement principal de notre chat est divisé en deux parties essentielles : la création de conversations entre deux personnes.
Ces conversations, seront alors des communications directes et sécurisées.

Présentation Fonctionnelle du projet:

Notre chat se présentera sous la forme d'une application web (webapp), offrant une interface client intuitive et accessible. L'application sera dotée d'un système d'authentification afin d'identifier chaque utilisateur autorisé. Une fois cette étape obligatoire passée avec succès, l'application dirigera l'utilisateur vers la section principale du chat et donnera l'accès à toutes les fonctionnalités de discussions.
Nous aurons la possibilité de créer des messages privés aux différents usagers de l'application ainsi que de faire des groupes de discussions.



lien du projet :
https://gigachat.chades.fr/

Présentation Technique du projet :

Cette webapp sera construite conteneurisée avec docker pour un meilleur déploiement et une meilleure maintenance.
Les technologies utilisées seront le python en backend avec le framework fastAPI.
Framework efficace et sécurisé offrant de nombreuses fonctionnalités.
La communication serveur-client est gérée par une websocket.
Nous utiliserons en frontend le langage JavaScript natif.
Le serveur de production sera quant à lui hébergé sur un ordinateur portable à Montpellier, nous utiliserons un serveur ubuntu avec un reverse proxy nginx et des certificats TLS OVH.
Ce serveur est protégé par une box internet bouygues qui possède un firewall, le service pi-hole tourne aussi sur le serveur pour augmenter sa sécurité.

Description précise des technologies utilisées :

Web Application (WebApp) :
La base de notre chat sécurisé est une WebApp, offrant une interface utilisateur intuitive et accessible via un navigateur web. Cette approche permet une utilisation flexible sans nécessiter d'installation complexe sur les appareils des utilisateurs.

Docker :
Pour assurer une gestion efficace des conteneurs et simplifier le déploiement, nous utilisons Docker. Cette technologie permet d'encapsuler notre application et ses dépendances, garantissant une portabilité et une reproductibilité accrues.

Backend - FastAPI (Python) :
Notre backend est construit en utilisant FastAPI, un framework Python moderne et rapide pour le développement d'API. Cette solution garantit une mise en œuvre rapide, une documentation automatique et une sécurité intégrée, répondant ainsi aux exigences essentielles de notre chat.

Frontend - JavaScript natif:
Le frontend de notre application repose sur du JavaScript natif, offrant ainsi une performance optimale et une flexibilité dans le développement. Cette approche minimale permet de maintenir un code léger et réactif pour une expérience utilisateur fluide.

Serveur de Production - Reverse Proxy Nginx :
Pour gérer le routage et la distribution du trafic, nous utilisons Nginx en tant que reverse proxy. Il joue un rôle crucial dans la sécurisation des connexions et l'optimisation des performances en dirigeant le trafic vers le backend approprié.

Certificat TLS OVH :

La sécurité des communications est une priorité, c'est pourquoi nous utilisons un certificat TLS fourni par OVH. Cela garantit le chiffrement des données transitant entre le serveur et les utilisateurs, renforçant ainsi la confidentialité des échanges.

Système d'exploitation - Ubuntu 22.04.3 LTS sur PC portable :
Le serveur de production fonctionne sous le système d'exploitation Ubuntu, assurant une base solide et bien documentée pour héberger notre application. De plus, la version 22 étant une version stable, la sécurité et la documentation en est d'autant plus élevé.

Box Bouygues avec firewall :
Ce routeur internet permet à la fois d'accéder à notre serveur à distance tout en le protégeant des tentatives d'attaques grâce à de nombreux services comme son firewall par exemple.

Service Pi-hole :
Le service Pi-hole qui tourne sur le serveur permet de bloquer l'utilisation de tracker ce qui ajoute une couche de sécurité sur le côté confidentiel de l'application.

Sécurité :

Cible de sécurité :

Notre cible de sécurité sont les employés d'une entreprise pouvant discuter de choses confidentielles. Pas du niveau secret d'états mais ayant signé des clauses de confidentialité. Il ne faut donc pas que les discussions deviennent publiques et que des utilisateurs arrivent à avoir accès aux discussions ou qu'ils ne soient pas concernés.

Nous garantissons la confidentialité, l'authentification et l'autorisation, l'intégrité, la disponibilité et la non-répudiation

Mesures de sécurité :

Authentification :

- Délai de traitement du mot de passe de 3 secondes si le précédent input est un échec pour un utilisateur spécifique (traitement côté serveur)
    - bloque le bruteforce mono et multi threads
- Si le token de connexion expiré, l'utilisateur est redirigé vers la page de connexion afin de ne pas laisser l'accès à un user non-authentifié.
- Un Captcha sera aussi utilisé pour rendre plus sécurisé la connexion

Confidentialité :

- les communication front - API sont chiffrée par le certificat "Let's Encrypt" validé par OVH
- le pi-hole du server permet de bloquer les trackers

Disponibilité :

- Gestion d'erreurs pour gérer le maximum de cas d'erreurs et ainsi éviter les pannes et crash du serveur.
- Le firewall du serveur bloque les tentatives de DDOS (firewall de base des box bouygues)

Intégrité :

Non-répudiation :

Autres mesures :

- Les mises à jour sont déployé via docker ce qui rend la maintenance de l'application plus facile et minimise les erreurs de déploiement
- Bonne pratique de code pour la sécurité
    - les mots de passes sont dans le .env sur le serveur et pas dans le code
    - On utilise le innerText en js et pas le innerHTML pour éviter l'exécution de script JavaScript dans les champs d'écriture de messages

- Requêtes préparés pour les champs d'authentification pour éviter les injections SQL
- Constructions d’une doc et de commentaire pour faciliter la compréhension du code

Intéresse le prof :

- *Savoir ce qui est mis en place pour la dispo et fiabilité (redondance, backup, …)*
- *Utiliser un Linter (MegaLinter conseillé)*