Rapport de projet

De Martino Giada Flora, Dorain Tetu

Table of Contents

Introduction

Beep est une application de messagerie sociale permettant aux utilisateurs de **communiquer, créer des communautés et interagir en temps réel**. Elle propose une expérience complète mêlant messagerie instantanée, appels audio/vidéo, partage de fichiers et gestion collaborative de serveurs (espaces de discussion).

Son architecture repose sur une infrastructure distribuée intégrant un backend robuste, une interface utilisateur réactive, ainsi qu'un service de messagerie électronique basé sur le protocole SMTP.

Ce rapport détaille les résultats de ces analyses, les risques identifiés, ainsi que les actions correctives recommandées pour renforcer la sécurité globale du projet.

Diagramme d'architecture

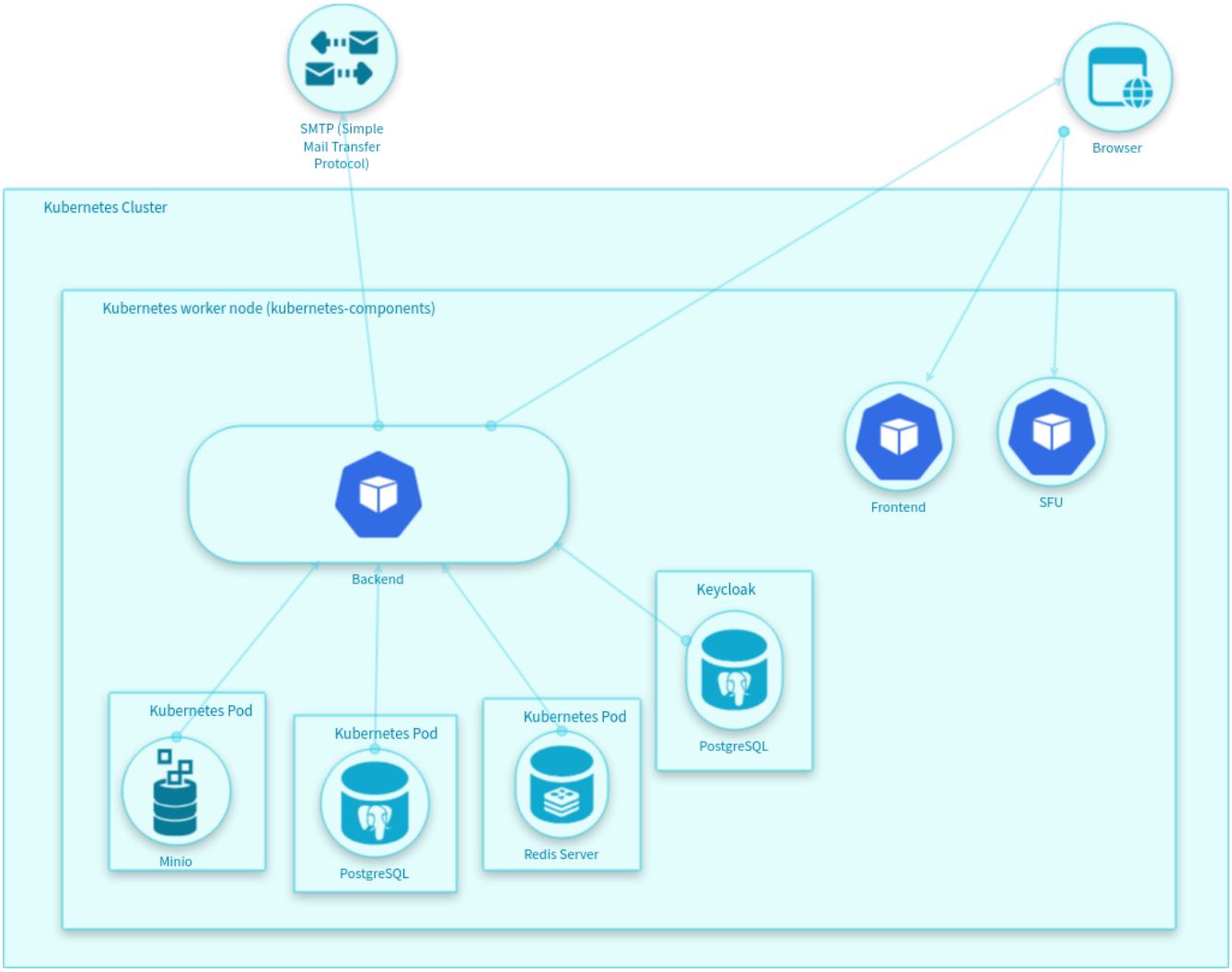

Figure 1. Diagramme d'architecture du projet Beep

Pourquoi effectuer des scans de sécurité

L'objectif des scans de sécurité est de détecter les vulnérabilités potentielles dans l'application afin

de renforcer sa résilience face aux attaques. Deux types de scans ont été réalisés :

- **SAST (Static Application Security Testing)** : Analyse du code source pour identifier les failles de sécurité avant l'exécution.
- **Scan Trilli** : Analyse ciblée des configurations de l'infrastructure, notamment le service SMTP.

Ces analyses permettent de prendre des mesures proactives avant la mise en production et de se conformer aux bonnes pratiques de sécurité logicielle.

Résultats des scans

SAST IriusRisk

Voici les 5 **threats** les plus critiques identifiés dans l'application Beep, principalement centrés autour du protocole SMTP :

- **Threat 1** : `SMTP` – **Mise en œuvre de STARTTLS**, implémenter STARTTLS pour chiffrer la transmission des e-mails et éviter l'interception de données sensibles.
- **Threat 2** : `SMTP` – **Limiter le taux d'envoi (Rate Limiting)**, prévenir les attaques de type flood en limitant le nombre de messages envoyés par IP ou utilisateur.
- **Threat 3** : `SMTP` – **Configuration sécurisée et audits réguliers**, éviter les fuites de données dues à une mauvaise configuration en auditant régulièrement les serveurs SMTP.
- **Threat 4** : `SMTP` – **Authentification des e-mails**, protéger contre l'usurpation d'identité via SPF, DKIM et DMARC.
- **Threat 5** : `SMTP` – **Authentification et prévention du relai non autorisé**, empêcher l'utilisation non autorisée du serveur SMTP comme relais en appliquant des contrôles stricts.

SAST Snyk

Le scan Snyk a révélé plusieurs vulnérabilités dans les dépendances du projet Beep, notamment :

- **Vulnérabilité 1** : `Dockerfile` – Mettre à jour la verison de Node vers 22.16.0 pour corriger des failles critiques.
- **Vulnérabilité 2** : `Hardcoded Secret` – Changer la valeur utilisée comme clé de chiffrement (dans jsonwebtoken.default.sign). Utiliser la valeur définie de `APP_KEY` dans le `.env`.

Dautres vinérabilitée ont étées trouvées, notament l'utilisation de la fonction send() dans le code source, qui peut exposer des données sensibles si mal configurée.

GitHub Action CI

La CI de GitHub Action, avec l'utilisation des images de Trinik et Snyk, nous révèle les suivantes vulnérabilitées :

- **Vulnérabilité 1** : `Node.js` – Mettre à jour la verison de Cross-spawn vers 7.0.5 pour avoir une

version plus stable.

Exploitation des résultats et plan d’action

Les résultats obtenus orientent la sécurisation du serveur SMTP, composant central de Beep. Voici les actions prioritaires recommandées :

- Mettre à jour la configuration du serveur SMTP pour activer STARTTLS et restreindre les connexions non sécurisées.
- Implémenter une politique de **rate limiting** adaptée aux usages réels.
- Mettre en place des audits automatiques et périodiques de la configuration SMTP.
- Ajouter les enregistrements DNS nécessaires pour SPF, DKIM et DMARC afin de renforcer la légitimité des e-mails.
- Appliquer une authentification forte sur les comptes SMTP et désactiver tout relai non autorisé.

Ces mesures amélioreront considérablement la sécurité de l’application et réduiront le risque de compromission via le canal de communication e-mail.