SAÉ 3.02 - Développer des applications communicantes

Mohamed Droussi
Johan Abdallah
Nicolas Nonnenmacher
Mathis Desmet

Sommaire

- Définition Du Problème
- Gestion du projet
- Outils utilisés
- Présentation de notre solution
- Démonstration

Définition du problème

Concevoir un système capable de :

- **Scanner** les failles d'un réseau.
- **Visualiser** les résultats sur un tableau de bord Web.
- **Consulter** ses informations via une application Android.

Gestion Du projet

Mise en place d’un RACI pour se répartir les taches

SAE 302	MOHAMED	MATHIS	JOHAN	NICOLAS
Squelette du programme java	A	I	R	C
Maquette de l'app Androïd	C	I	A	R
Maquette Site Web	I	R	A	C
Schema Base de Données	A	I	R	A
Communication web <-> android	R	C	A	I
Creation App Android java	R	C	A	I
Creation Site Web	C	I	R	A
Importation Base de Données	C	A	R	I
Creatopn App Java pour recherche failles (kali)	A	I	R	C

Outils utilisées

Frontend Mobile

- **Android Studio** Environnement de développement et compilation.
- **Java (Android)** Logique métier, Intents et Adaptateurs.
- **XML** Design des interfaces (Layouts, CardViews).

Backend & Données

- **Serveur Java** Pont entre le mobile et le système (Port 8080).
- **SQLite** Base de données locale (Fichier `bd.db`).
- **JDBC** API de connexion Java vers SQLite.

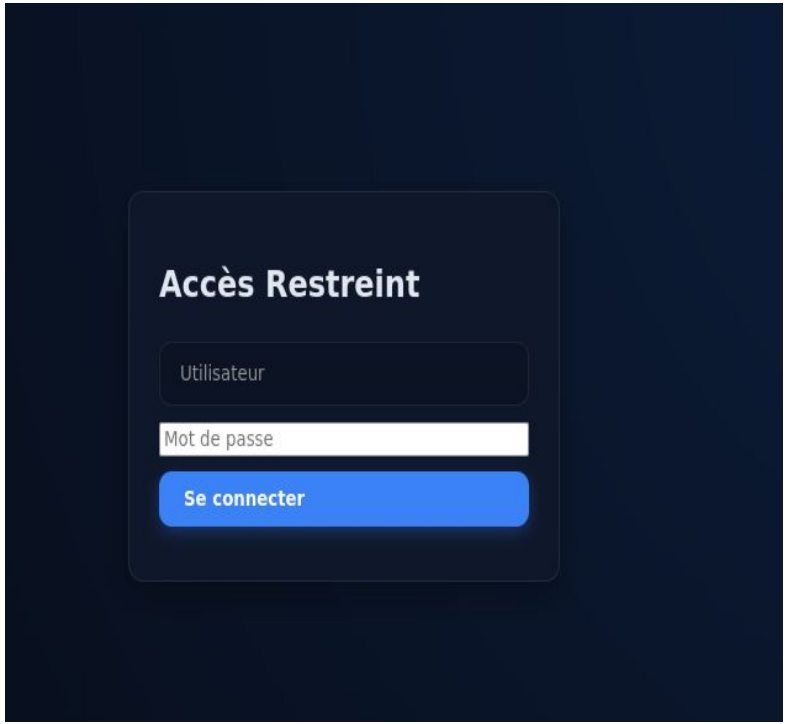
Réseau & Sécurité

- **Nmap** Moteur de scan de ports et vulnérabilités.
- **HTTP / JSON** Protocole d'échange App ↔ Serveur.
- **SHA-256** Hachage sécurisé des mots de passe.

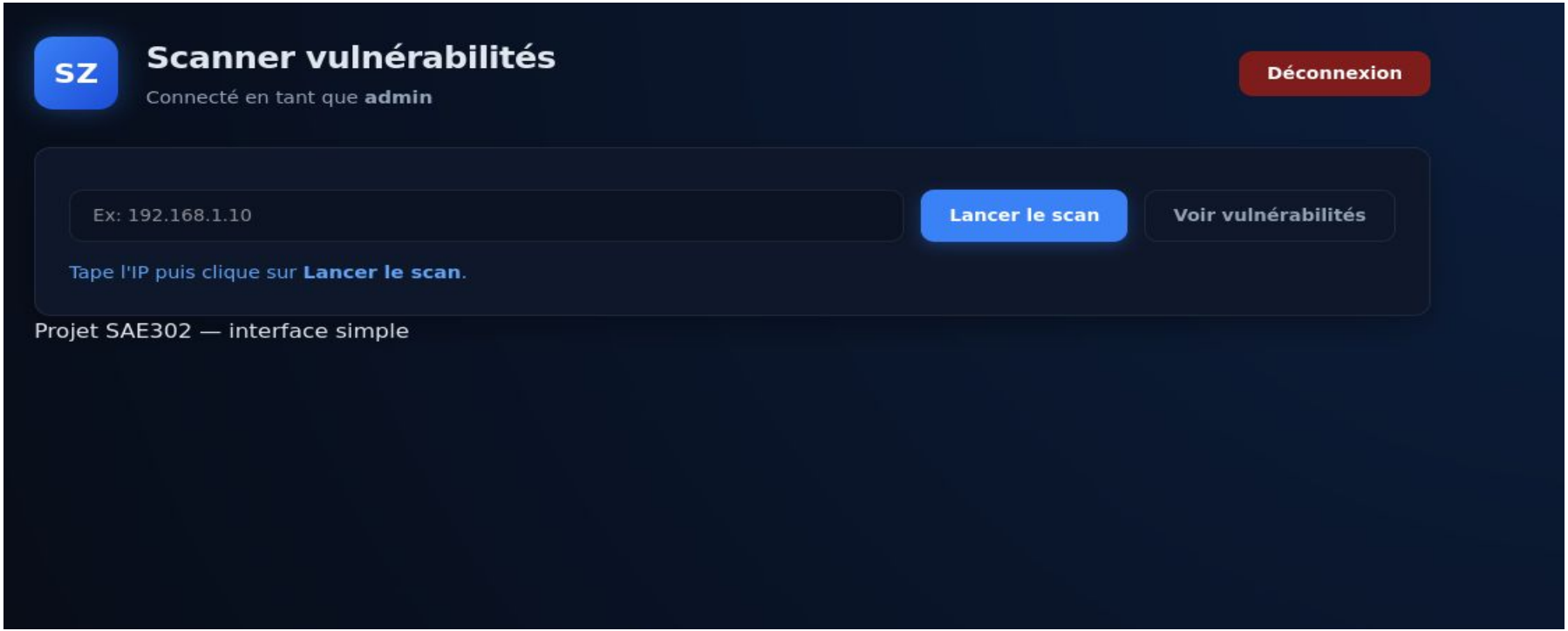
Administration Sys.

- **VM Debian** Environnement d'hébergement du scanner.
- **Apache** Serveur Web et redirection de flux.
- **HTML / CSS** Interface Web d'administration.

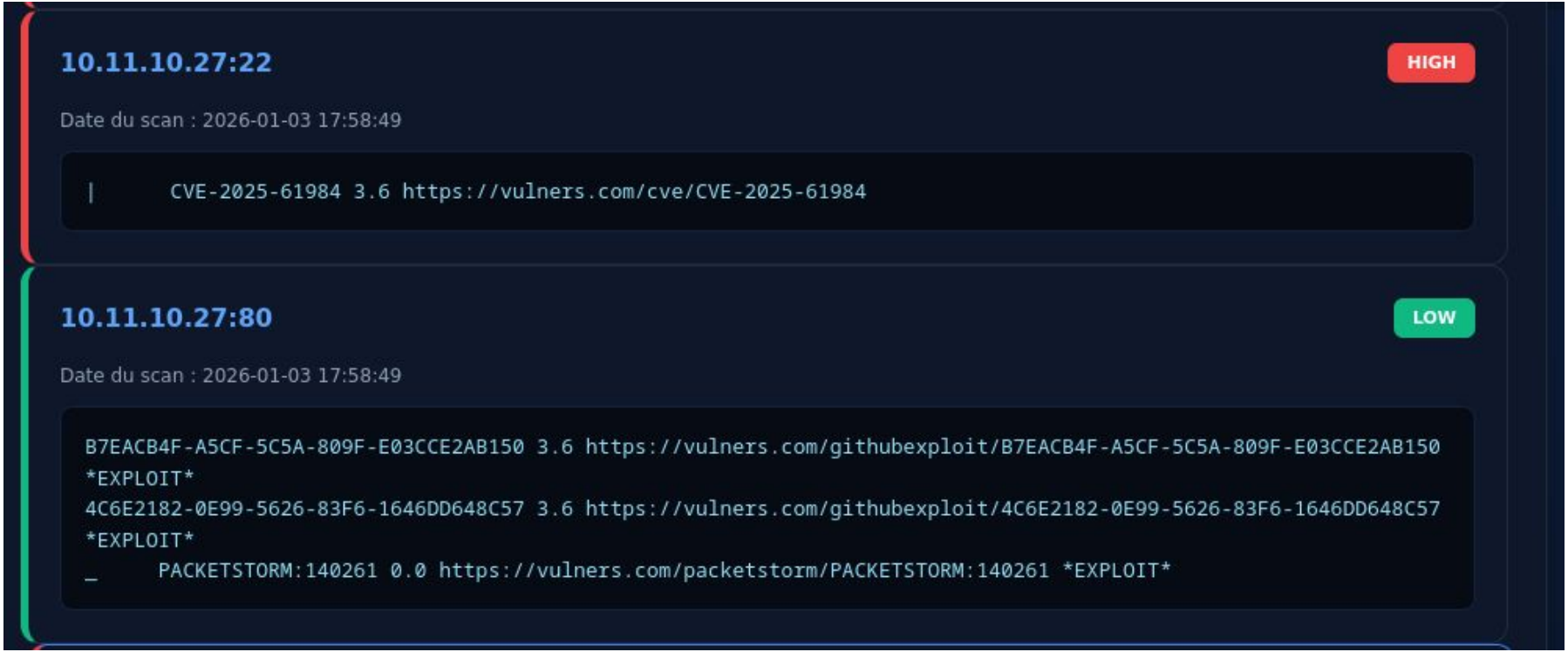
Présentation de notre solution



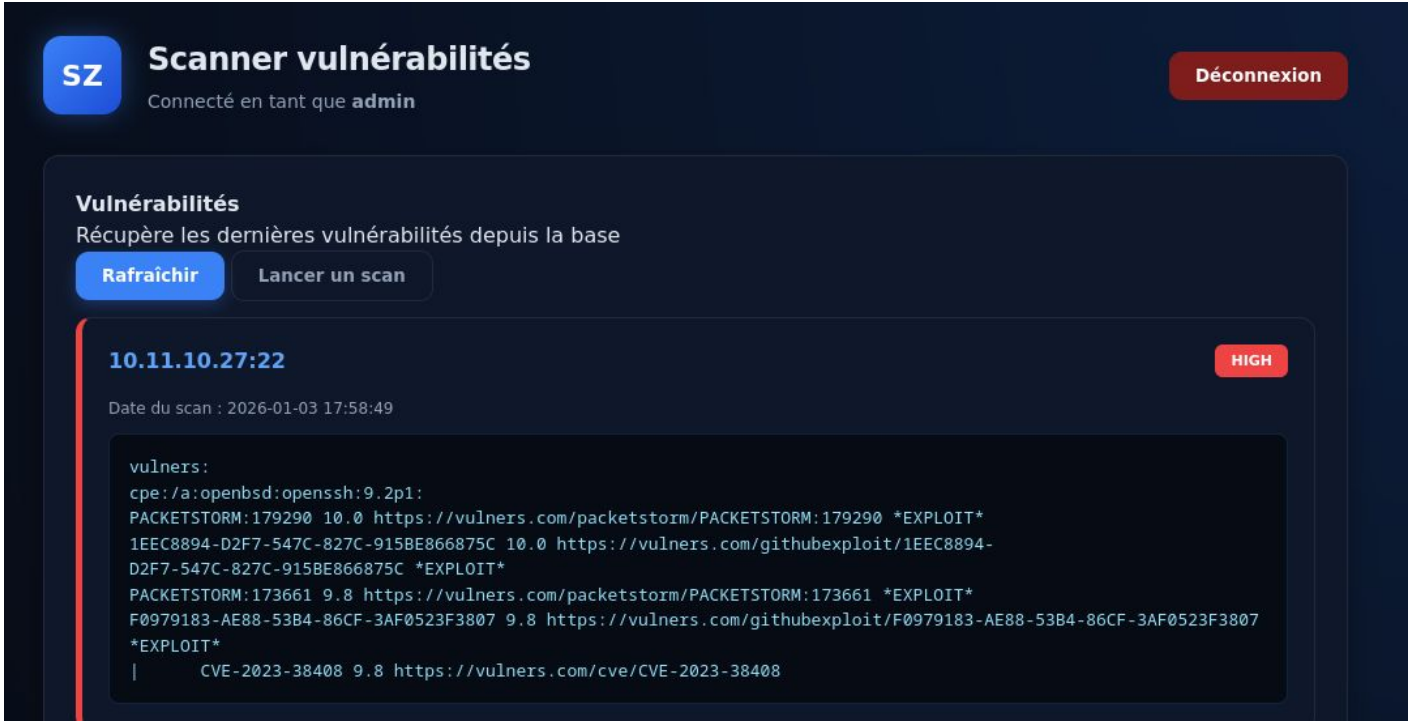
Page accueil site



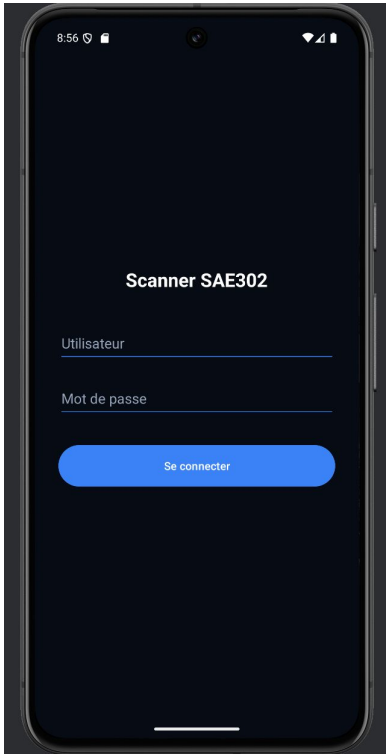
Listes des vulnérabilités

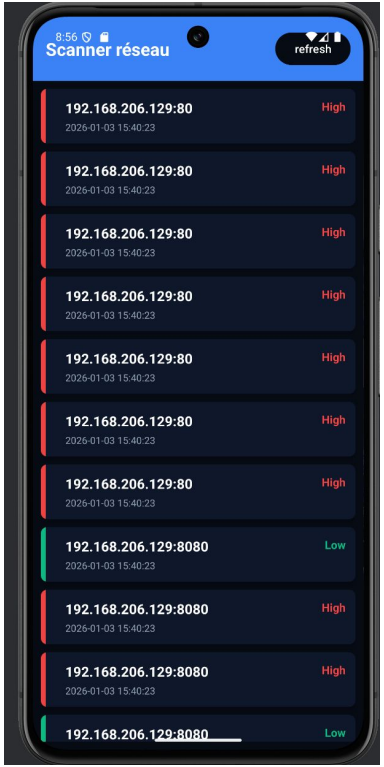


Détails du scan



Interface Android





Démonstration