|  |  |  |  |
| --- | --- | --- | --- |
|  |  | RAPPORT SAE32  Badaoui WALID Samperez ALEXANDRE Cruz-Mermy JULIEN |  |
|  |  |  |  |
| Une image contenant texte, ordinateur, ordinateur portable, capture d’écran  Le contenu généré par l’IA peut être incorrect. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
|  |  | TABLE DES MATIÈRES |  |
|  |  | Introduction .……....….….…………….….…….3  Méthode .…………....….….…………….….…….4  Discussions ………...….….…………….….…….5  Rrésultats ..………....….….…………….….…….6 |  |

|  |  |  |
| --- | --- | --- |
| INTRODUCTION |  | |
| La SAÉ 3.02 – Développer des applications communicantes s’inscrit dans le cadre du parcours Cybersécurité du BUT R&T. L’objectif principal est de concevoir une solution complète capable de détecter, stocker et visualiser des failles de sécurité sur un réseau local. Le projet repose sur trois volets interdépendants :   * une application Java chargée d’analyser le réseau et d’identifier les vulnérabilités ; * un site web qui centralise les résultats dans une base de données ; * et une application Android permettant de consulter ces failles en temps réel via une interface mobile.   L’ensemble du système communique grâce à des échanges au format JSON, garantissant la compatibilité et la synchronisation entre les différentes plateformes. Ce projet met en avant notre capacité à concevoir des applications interconnectées, à gérer une base de données sécurisée et à automatiser la recherche de vulnérabilités réseau dans un contexte collaboratif et professionnel.  Sous-titre — Objectifs et portée du projet  Le projet vise à **sensibiliser aux enjeux de la cybersécurité** tout en développant des compétences techniques concrètes en **programmation, gestion de données et communication réseau**. Chaque composant joue un rôle complémentaire :   * **Java** pour l’analyse et la détection automatique de failles (via des outils comme *nmap*, *nikto* ou *sqlmap*) ; * **SQLite** pour la gestion locale des données détectées ; * **le site web** pour la visualisation, la gestion et la sécurisation des informations collectées ; * **l’application Android** pour rendre ces données accessibles et exploitables sur mobile.   Au-delà de l’aspect technique, le projet met l’accent sur la **sécurité des échanges**, la **structuration du code** et la **collaboration en équipe** autour d’un **RACI** et d’un dépôt *GitHub* commun, afin de reproduire les conditions réelles d’un projet professionnel. | |  |
| Une image contenant texte, dessin humoristique, illustration  Le contenu généré par l’IA peut être incorrect. | |  |
|  | |  |

|  |  |  |
| --- | --- | --- |
| Méthode |  | |
| Présentation de la maquette  Faille  Notre base Java est composée de la classe « Faille », son but est de représenter une vulnérabilité détectée, elle peut avoir les attributs : int id, String nom, String ip, String description, String severité, String source, String dateDetection. Ces attributs sont encore théoriques.  On peut mettre aussi ses attributs « getId() », getNom(), getIp() aussi toJson() pour l’envoi à l’API peut être.  DatabaseManager  Cette Classe sert à centraliser l’accès à la base de donnée sur un seul et même fichier et isoler le SQL du reste du code.  Comme attributs on peut avoir : String dbPath, Connection conn pour gérer les connexions (ouverture/fermeture), pour ce qui est des méthodes on a : int insertFaille(Faille f) ajoute une vulnérabilité et peut retourner l’id inséré, List<Faille> getAllFailles() récupère toutes les failles dans la bdd et retourne une liste d’objets « Faille », getFailleById(int id) récupère une faille grâce à son id et retourne un objet Faille ou null si objet pas trouvé. List<Faille> getFaillesBySeverity(String severity) recuère les failles d’un certain niveau, void deleteFaille(int id), void updateFaille(Faille f).  ScannerReseau  Déclencher des scans réseau généraux (ex. nmap, nikto) et transformer les résultats en objets Faille.  En attributs nous avons-nous DatabaseManager db, List<String> targets, void detecterFaillesSimule() pour les tests, void lancerNmap(String target) pour tester avec un outil comme nmap, List<Faille> parserNmapOutput(String rawOutput) extraire les données issue du scan nmap,  BdD  Après renseignements on pense faire la base de données sur SQLlite ça va nous permettre d’avoir un seule fichier en .db au lieu de un serveur externe.  **Faille**  **But :** représenter une vulnérabilité détectée. **Attributs (théoriques) :**  int id  String nom  String ip  String description  String sévérité (ex. LOW / MEDIUM / HIGH)  String source (ex. sqlmap, nikto, nmap)  String dateDetection  **Méthodes (exemples) :**  getId(), getNom(), getIp() (getters)  toJson() — convertir l’objet pour l’envoi à l’API | |  |

DatabaseManager

**But :** centraliser l’accès à la base SQLite, isoler la logique SQL du reste du code.  
**Attributs :**

* String dbPath — chemin vers failles.db
* Connection conn (conceptuel) — gestion de la connexion (ouverture/fermeture)

**Méthodes principales :**

* void createTableIfNotExists() — crée la table failles si nécessaire
* void open() / void close() — ouvrir/fermer la connexion à la BDD
* int insertFaille(Faille f) — ajoute une faille et retourne l’id inséré
* List<Faille> getAllFailles() — retourne toutes les failles sous forme d’objets Faille
* Faille getFailleById(int id) — retourne un Faille ou null si non trouvé
* List<Faille> getFaillesBySeverity(String severity) — filtre par sévérité
* void updateFaille(Faille f) — met à jour une faille existante
* void deleteFaille(int id) — supprime une faille

ScannerReseau

**But :** déclencher des scans réseau (nmap, nikto, …) et transformer les résultats en objets Faille.  
**Attributs :**

* DatabaseManager db
* List<String> targets — liste d’IP / hôtes à scanner

**Méthodes (squelette) :**

* void detecterFaillesSimule() — insertion de résultats fictifs (TD1)
* void lancerNmap(String target) — stub pour exécuter nmap (TD2)
* List<Faille> parserNmapOutput(String rawOutput) — extraire des Faille depuis la sortie nmap
* void lancerNikto(String target) — stub pour nikto
* List<Faille> parserNiktoOutput(String rawOutput) — parser la sortie nikto
* void runFullScan() — orchestration des scans

RACI

Une image contenant texte, capture d’écran, nombre, Police

Le contenu généré par l’IA peut être incorrect.

Développement Java Parse

Ici il s’agit de concevoir le module Java chargé d’analyser et d’interpréter les fichiers ou flux de données issus des outils de scan (nmap, sqlmap, etc.).

* **Walid (R)** : écrit et teste le code du parseur Java.
* **Alexandre (A)** : valide la logique de parsing et les choix techniques.
* **Julien (I)** : est informé des avancées et peut tester le module à la fin.

*Ce module permet de transformer les résultats bruts des outils de sécurité en objets exploitables (classe Faille).*

Le module **Java Parse**, lui, intervient **après le scan**.  
Il lit les fichiers produits par Java Scan, **analyse leur contenu** et **extrait uniquement les informations utiles** (nom du service, IP, faille détectée, gravité, etc.).  
Ces données filtrées sont ensuite **converties en objets Faille** et envoyées vers la base de données (via DatabaseManager) ou vers le site web (API JSON).

Développement Java Scan

Il faut développer la partie Java qui exécute les outils de scan réseau ou web et récupère leurs résultats.

* **Walid (A)** : supervise le développement et les tests du scanner.
* **Alexandre (R)** : code les fonctions de lancement et de récupération des résultats.
* **Julien (I)** : est informé de la structure des données pour les réutiliser côté Android.

*C’est le cœur de la détection des vulnérabilités. Il interagit avec nmap, nikto ou sqlmap.*

**Java Scan — la détection**

* Le module **Java Scan** a pour rôle de **lancer les outils d’analyse** du réseau ou des sites web (par exemple *nmap*, *nikto* ou *sqlmap*) et de **récupérer leurs résultats bruts**.  
  Il s’occupe donc de la **phase active de recherche** : il interagit avec le système, exécute les commandes et enregistre les sorties des outils dans des fichiers ou flux de texte.

Développement du site web

**Objectif :** concevoir l’interface web (front + back) permettant d’afficher les failles et d’interagir avec la base de données.

* **Walid (C)** : conseille sur les besoins en affichage et sécurité.
* **Alexandre (R)** : développe le site (API REST + pages web).
* **Julien (A)** : valide le rendu visuel et la cohérence fonctionnelle.

*Le site permet de consulter les failles, d’ajouter ou de modifier des données, et sert de passerelle entre la base et l’application Android.*

Conception de la base de données (BdD)

**Objectif :** définir la structure de la base SQLite et son intégration avec les différentes applications.

* **Walid (R)** : crée la base et les tables nécessaires (ex. table failles).
* **Alexandre (A)** : valide la structure (types, contraintes, relations).
* **Julien (I)** : est informé pour adapter l’application Android à cette structure.

🗣️ *La base centralise toutes les données collectées par le scanner et partagées via l’API.*

Gestion de projet (GitHub, RACI, Rapport)

**Objectif :** organiser, documenter et superviser l’avancement global du projet.

* **Walid (I)** : reste informé des décisions et de la planification.
* **Alexandre (R)** : gère le dépôt GitHub, les branches, les merges et la documentation.
* **Julien (A)** : valide le contenu du rapport et la cohérence des livrables.

*Cette partie garantit une bonne communication et un suivi constant entre les membres.*

Développement application Android

**Objectif :** concevoir l’application mobile pour consulter les failles et interagir avec l’API web.

* **Walid (A)** : supervise la structure de l’application et son lien avec l’API.
* **Alexandre (I)** : est informé de l’avancement pour garder la cohérence avec la base et le site.
* **Julien (R)** : développe l’application Android (interfaces, requêtes HTTP, affichage des failles).

*L’application permet d’afficher les failles détectées et de consulter leurs détails en temps réel depuis un smartphone.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Discussions | | |  | |
| Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi | | | |  |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Purus in massa tempor nec feugiat nisl pretium. Aliquam vestibulum morbi blandit cursus risus at ultrices mi. Ut morbi tincidunt augue interdum velit euismod in. Tortor posuere ac ut consequat semper viverra nam libero. Egestas sed sed risus pretium quam vulputate. Nunc mattis enim ut tellus elementum sagittis vitae et. Tristique senectus et netus et. Pulvinar neque laoreet suspendisse interdum consectetur libero id faucibus. Lacus laoreet non curabitur gravida arcu ac. Odio facilisis mauris sit amet massa. |  | Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. | |  |
| Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Purus in massa tempor nec feugiat nisl pretium. Egestas sed sed risus pretium quam vulputate. Nunc mattis enim ut tellus elementum sagittis vitae et. Tristique senectus et netus et. Pulvinar neque laoreet suspendisse interdum consectetur libero id faucibus. | | | |  |

|  |  |  |
| --- | --- | --- |
| Résultats |  | |
| Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.  Lorem ipsum dolor sit amet, cons ectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Purus in massa tempor nec feugiat nisl pretium. Aliquam vestibulum morbi blandit cursus risus at ultrices mi. Ut morbi tincidunt augue interdum velit euismod in. | |  |
| |  |  |  |  | | --- | --- | --- | --- | |  | Cours 1 | Cours 2 | Cours 3 | | EXPÉRIMENTEZ 1 | 90 | 70 | 85 | | EXPÉRIMENTEZ 2 | 70 | 65 | 85 | | EXPÉRIMENTEZ 3 | 85 | 80 | 60 | | |  |
|  | |  |