Reformulation du sujet — Projet Procom IMT Atlantique

Idrissa Houcine Othmane Néo

13 novembre 2025

Objectif principal

Développer un **assistant IA local et explicable** capable d'automatiser l'analyse de sécurité de dépôts de code `C/C++`, en identifiant les vulnérabilités pertinentes et en justifiant les conclusions de manière compréhensible et contextualisée.

Axes de développement

Module d'analyse des vulnérabilités (prioritaire)

- Extraire automatiquement les dépendances COTS d'un projet `C/C++` (via `git submodules`, fichiers `CMake` ou `Makefile`).
- Identifier les versions de chaque dépendance.
- Interroger les bases **CVE** pour lister les vulnérabilités connues.
- Réaliser une **analyse contextuelle intelligente** : déterminer si une vulnérabilité est effectivement exploitable dans le code.
  - Construction de l'AST (*Abstract Syntax Tree*) — **Houcine**.
  - Justification du choix, aspects légaux et traçabilité — **Othmane & Houcine**.
- Étudier les stratégies de #merge et concevoir un **outil de visualisation**.
- Raffiner la détection (versions, contexte d'appel, dépendances croisées).

IA explicable (prioritaire)

- Utiliser des modèles de langage locaux (via **Ollama**) pour :
  - Expliquer les causes d'une détection.
  - Décrire l'impact réel de la vulnérabilité dans le contexte du projet.
  - Adapter les explications selon le profil utilisateur (*développeur, RSSI, auditeur*).
- Créer des jeux d'exemples et scénarios de test.
- Référent technique IA explicable : **Houcine**.

Moteur de recommandations (optionnel)

— Générer automatiquement des **plans de remédiation**.

— Proposer des **stratégies de mise à jour ou contournements**.

— Prioriser les actions selon leur impact réel sur la sécurité du système.

Analyse de la surface d'attaque (optionnel – scénario 2)

— Cartographier les interfaces exposées.

— Vérifier la conformité du code aux bonnes pratiques de sécurité.

— Proposer des mécanismes de défense adaptatifs ou dynamiques.

Contraintes techniques

— Exécution **100 % locale (on-premise)** : aucune dépendance à des API externes.

— Environnement conteneurisé via **Docker**.

— Stack logicielle : **Python**, **Ollama**, **OpenWebUI**, **mlflow**, **pgVector**.

— Utilisation d'une **machine GPU (fournie par SOLENT)** pour exécuter les modèles IA.

Livrables attendus

— **Code source complet**, conteneurisé (Docker).

— **Module d'analyse des dépendances** fonctionnel.

— **Documents techniques** :

  — Architecture logicielle.

  — Analyse comparative des solutions existantes.

  — Rapport sur la surface d'attaque.

— **Présentations régulières** : justification des choix, synthèse d'avancement, démonstrations intermédiaires.

Synthèse

Le projet vise à concevoir un **scanner intelligent et explicable de vulnérabilités** qui ne se contente pas d'énumérer des CVE, mais **comprend** leur contexte d'exploitation, **explique** leur impact réel et **suggère** des pistes d'action, tout en garantissant la **confidentialité des données** grâce à une exécution locale.