



République Tunisienne

Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique

École Supérieur Privée d'ingénierie et de technologie

TEK-UP

**APAIA**  
Technology

## RAPPORT DE PROJET DE FIN D'ÉTUDES

Présenté en vue de l'obtention du

Diplôme National d'Ingénieur en Sciences Appliquées et Technologiques  
Spécialité : Génie Logiciel et Systèmes d'Information

Réalisé par

**Alaeddine Mansouri**

---

## Conception et développement d'une application web d'évaluation des CVE

---

Encadrant professionnel : **Monsieur Adel Amri**

CTO APAIA

Encadrant académique : **Monsieur Adel Amri**

Maître-assistant



J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **M. ADEL AMRI**

**Signature et cachet**

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant académique, **M. ADEL AMRI**

**Signature**

# Dédicace

Je dédie ce travail à :

Mes chers parents, mon frère , mes amis, à mes enseignants et à toute ma famille, sans leurs encouragements et sacrifices, rien de cela n'aurait été possible.

Toute ma reconnaissance...

# Remerciements

*Il est particulièrement agréable, avant de présenter cette oeuvre, d'exprimer toute ma gratitude envers les personnes qui de près ou de loin m'ont apporté leur aide inestimable lors de la réalisation de ce projet.*

*Je tiens tout particulièrement à remercier mes tuteurs de stage dans la société*

*Apaia-technology ,*

***M. Adel Amri et M. Vincent Boutteau.***

*J'adresse, mes sincères remerciements à mon encadrant académique,*

***M. Adel Amri,***

*Je tiens aussi à remercier tous les membres du jury pour avoir bien voulu examiner et juger ce travail.*

*Merci.*

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Cadre du projet</b>	<b>3</b>
1.1 Cadre général du projet . . . . .	4
1.1.1 Présentation de l'organisme d'accueil . . . . .	4
1.1.2 Contexte et Objectifs du projet . . . . .	5
1.2 Étude de l'existant . . . . .	6
1.2.1 Étude de l'application Splunk . . . . .	6
1.2.2 Étude de l'application Tenable . . . . .	8
1.2.3 Étude de l'application Kaspersky . . . . .	9
1.3 Synthèse pour le choix de la solution . . . . .	11
1.4 Méthodologie de gestion de projet à adopter . . . . .	12
1.4.1 Méthodologie de SCRUM . . . . .	12
1.4.2 Les rôles dans SCRUM . . . . .	12
1.4.3 Méthodologie de conception à adopter . . . . .	13
<b>2 Analyse et spécification des besoins</b>	<b>14</b>
2.1 Spécification des besoins . . . . .	15
2.1.1 Spécification des besoins fonctionnels . . . . .	15
2.1.2 Spécification des besoins non fonctionnels . . . . .	17
2.2 Diagramme de classe . . . . .	17
2.3 Planification de travail . . . . .	18
2.3.1 Répartition des releases . . . . .	18
2.3.2 Planification des sprints . . . . .	19
2.4 Architecture de l'application . . . . .	19
2.4.1 Architecture logique . . . . .	19
2.5 Les patrons de conception utilisés . . . . .	21
2.5.1 Patrons de création . . . . .	21
2.5.2 Patrons structurels . . . . .	22
2.5.3 Patrons comportementaux . . . . .	22
2.5.4 Patron d'architecture . . . . .	22
2.6 Difficultés rencontrées . . . . .	22

---

2.7	Environnement de travail . . . . .	23
2.7.1	Outils de gestion de projet . . . . .	23
2.7.2	Framework de développement et de tests . . . . .	25
2.7.3	Système de gestion de base de données . . . . .	27
<b>3</b>	<b>Release 1 : authentification et récupération des CVE</b>	<b>28</b>
3.1	Sprint 1 : authentification et intégration de l'API NVD . . . . .	29
3.1.1	Objectifs du sprint 1 . . . . .	29
3.1.2	Backlog du sprint 1 . . . . .	29
3.1.3	Spécification des besoins fonctionnels . . . . .	30
3.1.4	Diagramme de classe . . . . .	31
3.1.5	Diagrammes dynamiques . . . . .	32
3.1.6	Réalisation . . . . .	33
3.2	sprint 2 : filtrage des CVE et implémentation d'une pipeline avancée RAG . . . . .	35
3.2.1	Objectifs du sprint 2 . . . . .	35
3.2.2	Backlog du sprint 2 . . . . .	35
3.2.3	Technologies utilisées . . . . .	36
3.2.4	Spécification des besoins fonctionnels . . . . .	37
3.2.5	Diagrammes dynamiques . . . . .	37
3.2.6	Réalisation . . . . .	38
<b>4</b>	<b>Release 2 : Recuperation de la configuration utilisateur et des scans avancés</b>	<b>42</b>
4.1	sprint 3 : Module Récupération de la configuration système et scans avancés utilisant NMAP et les outils de gestion de configuration . . . . .	43
4.1.1	Objectifs du sprint 3 . . . . .	43
4.1.2	Backlog du sprint 3 . . . . .	43
4.1.3	Spécification des besoins fonctionnels . . . . .	44
4.1.4	Diagramme de classe . . . . .	45
4.1.5	Diagrammes dynamiques . . . . .	45
4.1.6	Réalisation . . . . .	47
4.2	Sprint 4 : Module gestion des paramètres de configuration utilisateur . . . . .	51
4.2.1	Objectifs du sprint 4 . . . . .	51
4.2.2	Backlog du sprint 4 . . . . .	51
4.2.3	Spécification des besoins fonctionnels . . . . .	52
4.2.4	Diagramme de classe . . . . .	53

---

4.2.5	Diagrammes dynamiques . . . . .	54
4.2.6	Réalisation . . . . .	55
<b>5</b>	<b>Release 3 : Analyse approfondie, Tableau de bord, Recommandations et dashboard</b>	<b>58</b>
5.1	Sprint 5 : Module d'Analyse approfondie et Tableau de bord . . . . .	59
5.1.1	Objectifs du sprint 5 . . . . .	59
5.1.2	Backlog du sprint 5 . . . . .	59
5.1.3	Spécification des besoins fonctionnels . . . . .	60
5.1.4	Diagramme de classe . . . . .	61
5.1.5	Diagrammes dynamiques . . . . .	62
5.1.6	Réalisation . . . . .	62
5.2	sprint 6 : Recommandations et Dashboard . . . . .	65
5.2.1	Objectifs du sprint 6 . . . . .	66
5.2.2	Backlog du sprint 6 . . . . .	66
5.2.3	Spécification des besoins fonctionnels . . . . .	66
5.2.4	Recommandations pour les Vulnérabilités . . . . .	67
5.2.5	Réalisation . . . . .	68
	<b>Conclusion générale</b>	<b>71</b>
	<b>Bibliographie</b>	<b>72</b>

# Table des figures

1.1	Interface du site Splunk . . . . .	7
1.2	Interface de la solution Tenable . . . . .	8
1.3	Interface de la solution Kaspersky . . . . .	10
1.4	Cycle de vie de la méthodologie SCRUM [1] . . . . .	12
2.1	Diagramme cas d'utilisation global . . . . .	16
2.2	Diagramme de classe global . . . . .	18
2.3	Planification des sprints . . . . .	19
2.4	Architecture logique . . . . .	20
2.5	Architecture logique de Django avec l'application [2] . . . . .	21
2.6	Logo GitHub . . . . .	24
2.7	Logo Bitbucket . . . . .	24
2.8	Logo Discord . . . . .	24
2.9	Logo Django . . . . .	25
2.10	Logo pytest . . . . .	25
2.11	Logo graph . . . . .	26
2.12	Logo nmap . . . . .	26
2.13	Logo ansble . . . . .	27
2.14	Logo postgresql . . . . .	27
3.1	Diagramme de cas d'utilisation "authentification et intégration de l'API NVD" . . . . .	31
3.2	Diagramme de classe "authentification et intégration de l'API NVD" . . . . .	32
3.3	Diagramme de séquence objet "ajout d'entreprise" . . . . .	33
3.4	interface de création de compte . . . . .	33
3.5	Page login 1.0 . . . . .	34
3.6	Page login 1.1 . . . . .	34
3.7	interface CVEs . . . . .	35
3.8	Diagramme de cas d'utilisation "filtrage des CVE et implémentation d'une pipeline avancée RAG" . . . . .	37
3.9	Diagramme de séquence objet "Tri et filtrage des cve" . . . . .	38
3.10	interface CVE . . . . .	39
3.11	interface de recherche des cve . . . . .	39

3.12	Filtrer les CVE par plage de dates . . . . .	40
3.13	visualisation en graph après l'indexation des cve . . . . .	40
3.14	visualisation en graph après l'indexation des cve(Zoom) . . . . .	41
4.1	Diagramme de cas d'utilisation " Recuperation de la configuration utilisateur et des scans avancés" . . . . .	44
4.2	Diagramme de classe " Recuperation de la configuration utilisateur et des scans avancé . . . . .	45
4.3	Diagramme de séquence système " Recuperation de la configuration utilisateur et des scans avancés" . . . . .	46
4.4	interface Home . . . . .	47
4.5	interface de téléchargement de configuration . . . . .	48
4.6	la configuration yaml afficher après le process système . . . . .	48
4.7	interface de resultat du scan aggressive . . . . .	49
4.8	interface de resultat du scan aggressive développée . . . . .	49
4.9	formulaire d'Outils de gestion de configuration . . . . .	50
4.10	Liste des dispositifs . . . . .	50
4.11	Détails des applications . . . . .	51
4.12	Diagramme de cas d'utilisation "Gestion des paramètres de configuration utilisateur" .	53
4.13	Diagramme de classe "Gestion des paramètres de configuration utilisateur" . . . . .	54
4.14	Diagramme de séquence objet "Gestion des paramètres de configuration utilisateur" .	55
4.15	interface de configuration LLM . . . . .	56
4.16	interface de configuration Utilisateur . . . . .	56
5.1	Diagramme de cas d'utilisation "Analyse approfondie et Tableau de bord" . . . . .	60
5.2	Diagramme de classe "Analyse approfondie et Tableau de bord" . . . . .	61
5.3	Diagramme de séquence système "Analyse approfondie et Tableau de bord" . . . . .	62
5.4	Interface des Résultats de l'Analyse des Vulnérabilités . . . . .	63
5.5	Interface de Progression en Arrière-Plan de l'Analyse des Vulnérabilités . . . . .	63
5.6	Interface de recherche . . . . .	64
5.7	Interface des Résultats du Filtrage par Niveau d'Impact . . . . .	64
5.8	Interface de Rapport d'Analyse Détaillée des CVE . . . . .	65
5.9	Interface des Composants Principals Affectés . . . . .	65
5.10	Diagramme de cas d'utilisation "Recommandations et Dashboard" . . . . .	67
5.11	Recommandations pour les Vulnérabilités . . . . .	68
5.12	Dashboard Employé . . . . .	69

5.13 Type des equipements . . . . .	69
5.14 Distribution des systemes d'exploitation . . . . .	69
5.15 Derniers CVEs et Vulnérabilités . . . . .	70

# Liste des tableaux

1.1	Points forts et points faibles de l'application Splunk . . . . .	8
1.2	Points forts et points faibles de l'application Tenable . . . . .	9
1.3	Points forts et points faibles de l'application Kaspersky . . . . .	11
2.1	Répartition des releases . . . . .	19
3.1	Backlog du Sprint 1 . . . . .	30
3.2	Backlog du Sprint 2 . . . . .	36
4.1	Backlog du Sprint 3 . . . . .	44
4.2	Backlog du Sprint 4 . . . . .	52
5.1	Backlog du Sprint 5 . . . . .	60
5.2	Backlog du Sprint 6 . . . . .	66

# Liste des abréviations

- **API** = Application Programming Interface
- **DAO** = Data Access Object
- **DI** = Dependency Injection
- **DTO** = Data Transfer Object
- **GLSI** = Génie Logiciel et Système d'Information
- **HTTP** = Hypertext Transfer Protocol
- **IT** = Information Technology
- **JSON** = JavaScript Object Notation
- **JWT** = JSON Web Token
- **LLM** = Large Language Model
- **RAG** = Retrieval Augmented Generation
- **REST** = Representational state transfer
- **UML** = Unified Modeling Language

# Introduction générale

L'essor de l'informatique et des technologies numériques a profondément bouleversé nos sociétés et transcende l'ensemble des activités humaines. De nos jours, la majorité de ces activités a été simplifiée et automatisée grâce à l'informatique. Ce phénomène, parfois qualifié de « révolution numérique », permet aujourd'hui le traitement et l'échange d'informations entre les individus, grâce à des logiciels et des réseaux informatiques.

Ainsi, les entreprises du monde entier utilisent aujourd'hui des logiciels de gestion, leur permettant d'effectuer des tâches complexes rapidement.

C'est dans ce contexte que le projet « CVE Assessment » a vu le jour. Ce projet propose une solution de gestion et d'évaluation des vulnérabilités (CVE), automatisée et utilisant des technologies avancées telles que l'intelligence artificielle générative (IA générative) et le RAG (Retrieval-Augmented Generation). Le projet « CVE Assessment » consiste en une application permettant de mettre à disposition des utilisateurs un outil performant pour le suivi et l'analyse des vulnérabilités, présentant des fonctionnalités diverses et utiles en toutes circonstances.

L'objectif de cette application est, d'une part, de simplifier et d'accélérer le processus de gestion des vulnérabilités, et d'autre part, d'améliorer la sécurité des systèmes informatiques en fournissant des analyses complètes et des recommandations concrètes. Pour ce faire, l'application s'appuie sur des modèles de langage naturel avancés comme Llama 3 et une pipeline RAG pour réaliser des analyses approfondies et pertinentes.

L'entreprise de services informatiques Apaia-technology, attachée à ces valeurs d'agilité et d'amélioration continue des processus, a accepté d'allouer les ressources nécessaires à la réalisation de ce projet.

Le présent rapport sera organisé de la manière suivante :

- Dans un premier chapitre, nous présenterons le cadre du projet. Il sera constitué d'une présentation du contexte général ainsi que d'une étude de l'existant, réalisée à partir d'un projet similaire. Ce premier chapitre contiendra aussi une présentation de la méthode de gestion de projet et de la méthode de conception que nous avons choisi d'adopter.
- Le second chapitre présentera une analyse et une spécification des besoins, fonctionnels et non fonctionnels, réalisés à partir de l'identification des acteurs de l'application. Il exposera aussi le diagramme de cas d'utilisation globale, ainsi que le diagramme de classes de l'application, la répartition des releases et l'architecture de l'application.
- Le troisième chapitre portera sur la Release 1 : « Sécurité et Gestion de la récupération des CVE

», qui a consisté en la réalisation d'un Sprint 1 portant sur l'authentification, l'intégration de l'API NVD, et la configuration du stockage dans une base de données PostgreSQL, ainsi que la réalisation d'un Sprint 2, portant sur l'implémentation d'une pipeline avancée RAG (graph RAG) pour l'extraction des entités et des relations, et leur stockage dans le graph de connaissances.

- Dans un quatrième chapitre, nous présenterons la Release 2, « Gestion de la configuration utilisateur et des scans avancés », qui comprend le Sprint 3, portant sur Gestion de la configuration Système et les scans avancés utilisant NMAP et les outils de gestion de configuration, et le Sprint 4, portant sur la gestion des paramètres de configuration utilisateur et le filtrage des CVE.
- Dans un cinquième chapitre, nous présenterons la Release 3 , « Analyse approfondie , Recommandations et dashboard», qui comprend le Sprint 5, portant sur l'analyse approfondie en utilisant la pipeline RAG avec le modèle Llama 3, et le Sprint 6, portant sur la création du tableau de bord pour visualiser les résultats de l'analyse ainsi les recommandations d'atténuation et l'optimisation de l'interface utilisateur

---

# CADRE DU PROJET

---

## Plan

1	Cadre général du projet . . . . .	4
2	Étude de l'existant . . . . .	6
3	Synthèse pour le choix de la solution . . . . .	11
4	Méthodologie de gestion de projet à adopter . . . . .	12

## Introduction

Dans ce chapitre, nous situons le projet dans son cadre global. Nous commençons par présenter l'entreprise dans laquelle nous avons réalisé notre stage de projet de fin d'étude. Puis, nous expliquons les objectifs du projet et nous analysons une application web existante. Enfin, nous présentons la méthodologie de projet que nous adoptons ainsi que l'environnement de travail.

### 1.1 Cadre général du projet

#### 1.1.1 Présentation de l'organisme d'accueil

##### 1.1.1.1 Apaia-Technology

L'entreprise APAIA TECHNOLOGY, pionnière dans le domaine de l'intelligence artificielle, se positionne au cœur de l'innovation technologique et de l'optimisation des processus d'affaires. Fondée sur une vision claire de l'impact de l'IA générative, elle offre des solutions visant à augmenter la productivité et à transformer la valeur proposée aux clients. En combinant conseil, expertise et formation, APAIA TECHNOLOGY accompagne ses clients dans la mise en œuvre de technologies avancées pour des gains significatifs en efficacité et en compétitivité.

L'entreprise est basé à Paris et dispose de bureaux à Tunis .

##### 1.1.1.2 Service

L'entreprise APAIA TECHNOLOGY fournit une expertise et des solutions innovantes, guidée en permanence par la volonté de résoudre les problèmes des entreprises de ses clients. Ses services peuvent être classifiés selon quatres axes principaux :

- **Solutions publicitaires** : Cette branche se concentre sur l'utilisation de l'IA générative pour créer des campagnes publicitaires ciblées et personnalisées, optimisant ainsi l'impact marketing et augmentant le retour sur investissement.
- **Consulting** : À travers une approche basée sur l'IA et le deep learning, APAIA TECHNOLOGY offre des conseils stratégiques pour l'automatisation des processus, l'amélioration de l'expérience client et la transformation numérique des entreprises.
- **Produits et services pour le domaine artistique** : En exploitant les capacités de l'IA générative, APAIA TECHNOLOGY développe des outils pour la création de contenu artistique, tels que la génération d'œuvres d'art, de musique, et d'animations, permettant aux artistes et aux créateurs de repousser les limites de leur imagination et d'augmenter leur productivité.

- **Recherche et Développement (RD) :** APAIA TECHNOLOGY investit continuellement dans la recherche et le développement pour rester à la pointe de l'innovation. Cette démarche permet de découvrir de nouvelles applications de l'IA générative et d'améliorer constamment les solutions proposées à ses clients, garantissant ainsi une compétitivité et une pertinence accrues sur le marché.

### 1.1.2 Contexte et Objectifs du projet

#### 1.1.2.1 Contexte

Le contexte de développement de l'application « CVE Assessment » s'inscrit dans un environnement où la cybersécurité est devenue une priorité essentielle pour les entreprises de toutes tailles. Les vulnérabilités, connues sous le nom de Common Vulnerabilities and Exposures (CVE), représentent des failles potentielles dans les systèmes informatiques qui peuvent être exploitées par des attaquants malveillants. La gestion et l'évaluation efficaces de ces vulnérabilités sont cruciales pour prévenir les cyberattaques et protéger les actifs informationnels.

Cependant, la gestion des CVE est une tâche complexe et chronophage. Les entreprises doivent surveiller continuellement les nouvelles vulnérabilités, évaluer leur impact potentiel sur leurs systèmes, et mettre en place des mesures d'atténuation appropriées. Cette tâche est d'autant plus ardue que le nombre de CVE signalées augmente chaque année, et que les menaces deviennent de plus en plus sophistiquées.

#### 1.1.2.2 Objectifs du projet

Le projet « CVE Assessment » a pour objectif principal de fournir une solution innovante et automatisée pour la gestion et l'évaluation des vulnérabilités (CVE). Plus précisément, les objectifs du projet sont les suivants :

- **Automatiser la gestion des vulnérabilités :** Développer une application capable de récupérer périodiquement les informations sur les CVE via l'API NVD, réduisant ainsi la charge de travail manuel et les erreurs potentielles associées au suivi des vulnérabilités.
- **Utiliser l'IA générative pour l'analyse des CVE :** Intégrer des technologies avancées comme Llama 3 (LLM) et une pipeline RAG (Retrieval-Augmented Generation) pour analyser en profondeur les vulnérabilités, permettant une évaluation précise et rapide des risques associés.
- **Fournir des recommandations concrètes :** Proposer des stratégies d'atténuation basées sur les résultats de l'analyse, afin d'aider les entreprises à prendre des mesures proactives pour sécuriser leurs systèmes informatiques.

- **Améliorer la personnalisation et la flexibilité :** Offrir des paramètres de configuration utilisateur pour le filtrage et le suivi des CVE, ainsi que des options pour des scans avancés utilisant NMAP et des outils de gestion de configuration comme Ansible ou Puppet.
- **Développer un tableau de bord intuitif :** Créer une interface utilisateur conviviale et un tableau de bord complet permettant de visualiser les résultats de l'analyse, les recommandations d'atténuation et l'état de la sécurité des systèmes en temps réel.
- **Renforcer la sécurité des systèmes :** Contribuer à une meilleure gestion des vulnérabilités en fournissant des outils et des analyses qui permettent aux entreprises de rester proactives face aux menaces de cybersécurité, réduisant ainsi les risques de cyberattaques.
- **Promouvoir l'innovation continue :** Encourager l'innovation en investissant dans la recherche et le développement pour améliorer constamment les capacités de l'application et découvrir de nouvelles applications de l'IA générative dans le domaine de la cybersécurité.

En atteignant ces objectifs, le projet « CVE Assessment » vise à révolutionner la manière dont les entreprises gèrent et atténuent les vulnérabilités, en leur fournissant des outils puissants et intelligents pour protéger leurs actifs informationnels et améliorer leur posture de sécurité globale.

## 1.2 Étude de l'existant

L'étude de l'existant est une étape indispensable, elle permet d'extraire les forces et les faiblesses des applications existantes. Cela nous aidera à la réalisation de notre projet. Nous avons choisi d'analyser trois applications existantes.

- L'application « Splunk »
- L'application « Tenable »
- L'application « Kaspersky »

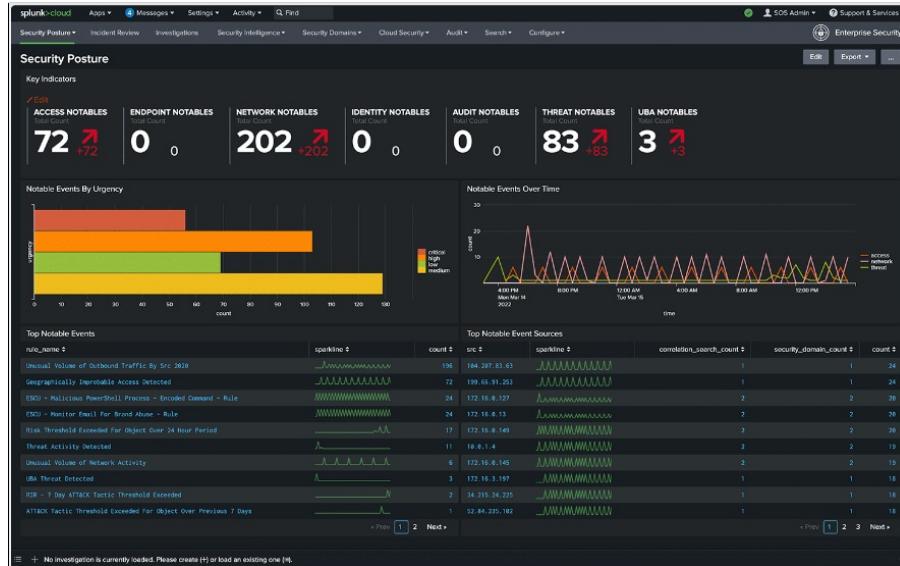
### 1.2.1 Étude de l'application Splunk

Adresse (URL) : <https://www.splunk.com/fr>

#### 1.2.1.1 Description

Splunk propose une plateforme complète de gestion des CVE qui permet aux entreprises de suivre, analyser et répondre aux vulnérabilités. En intégrant les données de la base de données nationale des vulnérabilités (NVD), Splunk aide les entreprises à évaluer leurs stratégies de cybersécurité et à mettre en place des mécanismes de défense robustes. L'outil offre également des fonctionnalités avancées de création de rapports et de visualisation des tendances des CVE.

- **Interface du site :** C'est l'interface d'accueil qui contient la description du site.



**FIGURE 1.1 : Interface du site Splunk**

### 1.2.1.2 Étude fonctionnelle

Les principales fonctionnalités proposées par l'application, tout en les reliant aux acteurs qui en bénéficient.

- **Utilisateur :**

- Recevoir des notifications sur les nouvelles vulnérabilités.
- Consulter les rapports d'analyse des vulnérabilités.
- Accéder aux recommandations de sécurité et aux mesures d'atténuation.

- **Administrateur :**

- Gérer les configurations de sécurité des systèmes.
- Planifier et exécuter des scans de vulnérabilités.
- Analyser les résultats des scans et prioriser les correctifs.
- Suivre les tendances des vulnérabilités pour améliorer les stratégies de sécurité.

### 1.2.1.3 Points forts et points faibles

Points forts	Points faibles
--------------	----------------

Points forts	Points faibles
<ul style="list-style-type: none"> <li>— Intégration avec les bases de données de vulnérabilités reconnues (NVD).</li> <li>— Automatisation des processus de détection et d'analyse des vulnérabilités.</li> <li>— Interface utilisateur intuitive et conviviale.</li> <li>— Capacités avancées de reporting et de visualisation des données.</li> </ul>	<ul style="list-style-type: none"> <li>— Coût élevé des solutions pour les petites entreprises.</li> <li>— Nécessité de ressources techniques pour l'installation et la gestion.</li> <li>— Dépendance à l'égard des mises à jour régulières pour maintenir l'efficacité.</li> </ul>

**TABLEAU 1.1 :** Points forts et points faibles de l'application Splunk

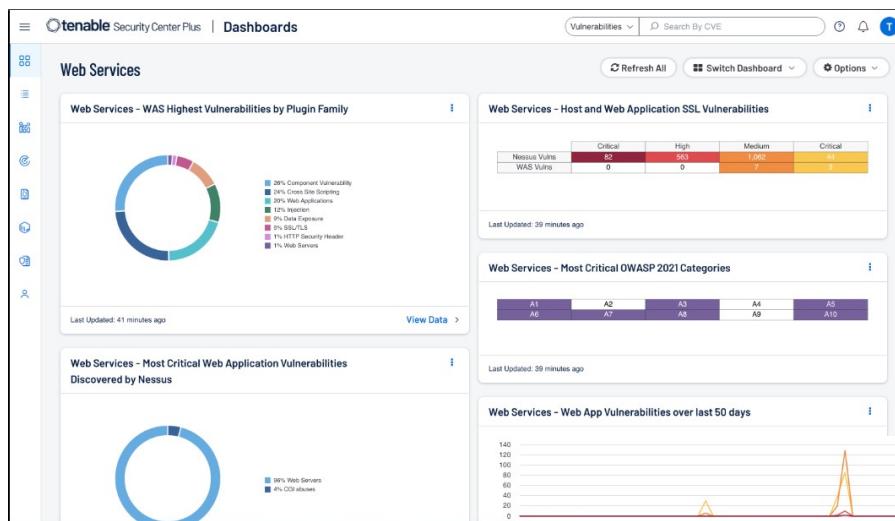
### 1.2.2 Étude de l'application Tenable

Adresse (URL) : <http://www.tenable.com>

#### 1.2.2.1 Description

Tenable offre des solutions comme Nessus, qui permettent de scanner et d'évaluer les systèmes pour identifier les vulnérabilités. Tenable se concentre sur l'automatisation de la détection des vulnérabilités et fournit des recommandations pour la correction des failles. La plateforme intègre également des informations sur les exploitations en cours, ce qui permet aux entreprises de prioriser les correctifs en fonction des menaces actuelles.

- **Interface de la solution :** La figure 1.2 Tenable propose une interface robuste pour la gestion des CVE, permettant aux utilisateurs .

**FIGURE 1.2 :** Interface de la solution Tenable

### 1.2.2.2 Étude fonctionnelle

Les principales fonctionnalités offertes par l'application, tout en les associant aux acteurs qui en bénéficient.

- **Utilisateur :**

- Recevoir des notifications sur les nouvelles vulnérabilités.
- Consulter les rapports d'analyse des vulnérabilités.
- Accéder aux recommandations de sécurité et aux mesures d'atténuation.

- **Administrateur :**

- Gérer les configurations de sécurité des systèmes.
- Planifier et exécuter des scans de vulnérabilités.
- Analyser les résultats des scans et prioriser les correctifs.
- Suivre les tendances des vulnérabilités et utiliser ces informations pour améliorer les stratégies de sécurité.

### 1.2.2.3 Points forts et points faibles

Points forts	Points faibles
<ul style="list-style-type: none"><li>— Intégration avec les bases de données de vulnérabilités reconnues (NVD).</li><li>— Automatisation des processus de détection et d'analyse des vulnérabilités.</li><li>— Interface utilisateur intuitive et conviviale.</li><li>— Capacités avancées de reporting et de visualisation des données.</li></ul>	<ul style="list-style-type: none"><li>— Coût élevé des solutions pour les petites entreprises.</li><li>— Nécessité de ressources techniques pour l'installation et la gestion.</li><li>— Dépendance à l'égard des mises à jour régulières pour maintenir l'efficacité.</li></ul>

**TABLEAU 1.2** : Points forts et points faibles de l'application Tenable

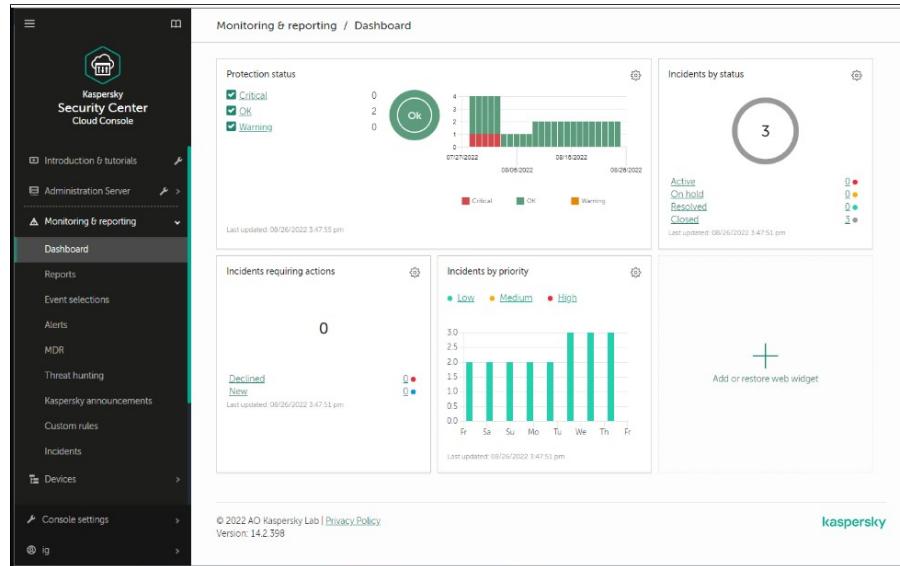
### 1.2.3 Étude de l'application Kaspersky

#### 1.2.3.1 Description

Kaspersky propose des solutions de gestion des vulnérabilités et de patch management. Leur plateforme aide à identifier et à corriger les vulnérabilités critiques dans les systèmes d'exploitation et

les applications. Kaspersky met également en avant l'importance de la rapidité de réaction après la divulgation publique des vulnérabilités, car les premières semaines sont souvent les plus critiques pour la prévention des attaques.

- **Interface de l'application :** La figure 1.3 représente l'interface d'accueil de l'application qui contient un dashboard qui présente toutes les fonctionnalités pour l'admin.



**FIGURE 1.3 : Interface de la solution Kaspersky**

### 1.2.3.2 Étude fonctionnelle

Les principales fonctionnalités proposées par l'application, tout en les reliant aux acteurs qui en bénéficient.

- **Utilisateur :**

- Recevoir des notifications sur les nouvelles vulnérabilités.
- Consulter les rapports d'analyse des vulnérabilités.
- Accéder aux recommandations de sécurité et aux mesures d'atténuation.

- **Administrateur :**

- Gérer les configurations de sécurité des systèmes.
- Planifier et exécuter des scans de vulnérabilités.
- Analyser les résultats des scans et prioriser les correctifs.
- Suivre les tendances des vulnérabilités et utiliser ces informations pour améliorer les stratégies de sécurité.

### 1.2.3.3 Points forts et points faibles

Points forts	Points faibles
<ul style="list-style-type: none"> <li>— Intégration avec les bases de données de vulnérabilités reconnues (NVD, CVE).</li> <li>— Automatisation des processus de détection et d'analyse des vulnérabilités.</li> <li>— Interface utilisateur intuitive et conviviale.</li> <li>— Capacités avancées de reporting et de visualisation des données.</li> </ul>	<ul style="list-style-type: none"> <li>— Coût élevé des solutions pour les petites entreprises.</li> <li>— Nécessité de ressources techniques pour l'installation et la gestion.</li> <li>— Dépendance à l'égard des mises à jour régulières pour maintenir l'efficacité.</li> </ul>

**TABLEAU 1.3 :** Points forts et points faibles de l'application Kaspersky

## 1.3 Synthèse pour le choix de la solution

Après l'analyse de trois applications, nous avons conclu que Notre application de gestion des CVE utilise l'IA générative précisément des modèles de langage naturel (LLM : Large Language Model) open source et privés pour offrir une analyse approfondie et un traçage précis des vulnérabilités. Voici les principales caractéristiques de notre solution :

- **Utilisation de l'IA générative :** En intégrant des technologies avancées telles que l'IA générative et les modèles de langage naturel, notre application est capable d'analyser de vastes ensembles de données complexes, fournissant des insights pertinents et des solutions adaptées aux besoins spécifiques de chaque entreprise.
- **Analyse et traçage des CVE :** Grâce à l'utilisation de la pipeline RAG (Retrieval-Augmented Generation), notre application assure une analyse approfondie des vulnérabilités et un traçage efficace des CVE, permettant ainsi aux entreprises de rester proactives face aux menaces de cybersécurité.
- **Compatibilité avec toutes les tailles d'entreprises :** Notre application est conçue pour répondre aux besoins des entreprises de toutes tailles, offrant des fonctionnalités adaptées tant aux petites startups qu'aux grandes entreprises.
- **Options de récupération de configuration :** L'application facilite la gestion des configurations de sécurité en permettant la récupération des configurations de manière manuelle, textuelle

(écrite par un ingénieur de sécurité), ou via des scans agressifs ainsi que des outils de gestion de configuration comme Ansible ou Puppet.

## 1.4 Méthodologie de gestion de projet à adopter

### 1.4.1 Méthodologie de SCRUM

Nous recherchons toujours la méthode la plus efficace et la plus rapide de mettre en œuvre notre projet. Pour cela, nous utilisons des méthodes agiles. Scrum est le cadre agile le plus simple. Scrum garantit la meilleure vue d'ensemble de notre projet et vise à réduire les difficultés, telles que le manque de planification, le travail est effectué à travers un cycle court appelé Sprint. Dans Sprint, notre équipe travaille à partir d'une liste d'éléments appelée Backlog" [B1].

Nous avons choisi la méthodologie **SCRUM** qui fait partie de la méthodologie **Agile** pour différentes raisons parmi lesquelles :

- La transparence.
- La tolérance aux différents changements.
- Les besoins ne sont pas bien connus au départ ils ont été changés par la suite.
- La présence du mélée quotidien qui permet de bien résoudre les problèmes et de trouver des solutions rapidement.

La figure 1.4 représente le parcours de la méthodologie de SCRUM.

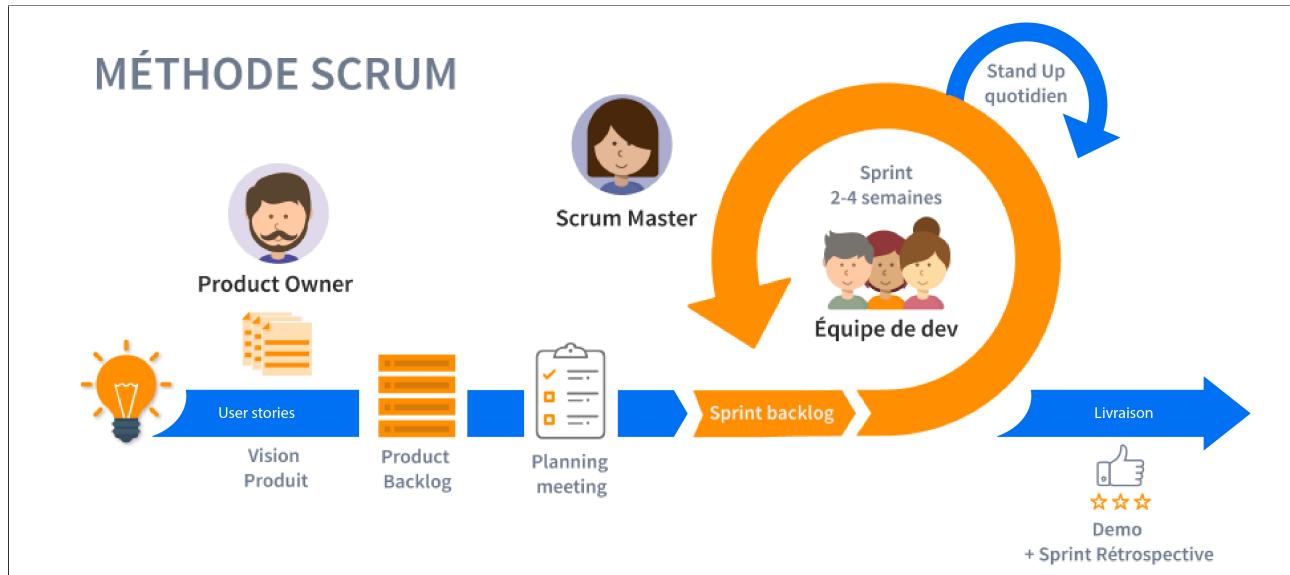


FIGURE 1.4 : Cycle de vie de la méthodologie SCRUM [1]

### 1.4.2 Les rôles dans SCRUM

La méthodologie d'agile SCRUM implique trois rôles principaux qui sont :

- **Product owner** : C'est le représentant des clients et des utilisateurs et c'est lui qui est l'expert métier de l'équipe. C'est à lui de définir et prioriser la liste des fonctionnalités du produit et effectuer l'analyse nécessaire pour la prise des décisions.
- **Scrum master** : C'est le garant de la méthodologie de SCRUM, qui garantit que tout le monde peut maximiser ses capacités en éliminant les obstacles, et en protégeant l'équipe des perturbations externes. Par ailleurs il garantit que l'équipe chargée du projet adopte les principes et les valeurs de SCRUM.
- **Équipe** : L'équipe rassemble tous les rôles généralement nécessaires à un projet, elle est organisée et reste inchangée pendant la durée d'un sprint.

#### 1.4.3 Méthodologie de conception à adopter

La modélisation du système d'information a pour but de permettre aux entreprises de communiquer avec des services ou des entreprises spécialisées dans l'informatique et de décrire leurs opérations et leurs besoins. Le modèle peut être résumé de manière claire et compréhensible pour tout le monde. Pour cela, nous avons choisi le langage de modélisation UML (Unified Modeling Language), car il s'appuie sur la standardisation et pour la diversité de ses diagrammes. Ce qui permet de réaliser une analyse détaillée des besoins, des vues statiques et dynamiques...

## Conclusion

Dans ce chapitre, nous avons présenté le projet dans son cadre général, nous passons maintenant à la spécification des besoins.

# ANALYSE ET SPÉCIFICATION DES BESOINS

---

## Plan

1	Spécification des besoins . . . . .	15
2	Diagramme de classe . . . . .	17
3	Planification de travail . . . . .	18
4	Architecture de l'application . . . . .	19
5	Les patrons de conception utilisés . . . . .	21
6	Difficultés rencontrées . . . . .	22
7	Environnement de travail . . . . .	23

## Introduction

Dans ce chapitre, nous présentons une analyse des besoins fonctionnels et non fonctionnels de notre applications en précisant les différents acteurs de notre système, le diagramme de classe et la planification de travail. Nous allons ainsi présenté l'architecture et les fonctionnalités utilisées.

### 2.1 Spécification des besoins

#### 2.1.1 Spécification des besoins fonctionnels

##### 2.1.1.1 Identification des acteurs

Dans cette section, nous allons identifier l'acteur impliqué dans l'utilisation de notre application d'analyse des CVE et spécifier son besoin fonctionnel.

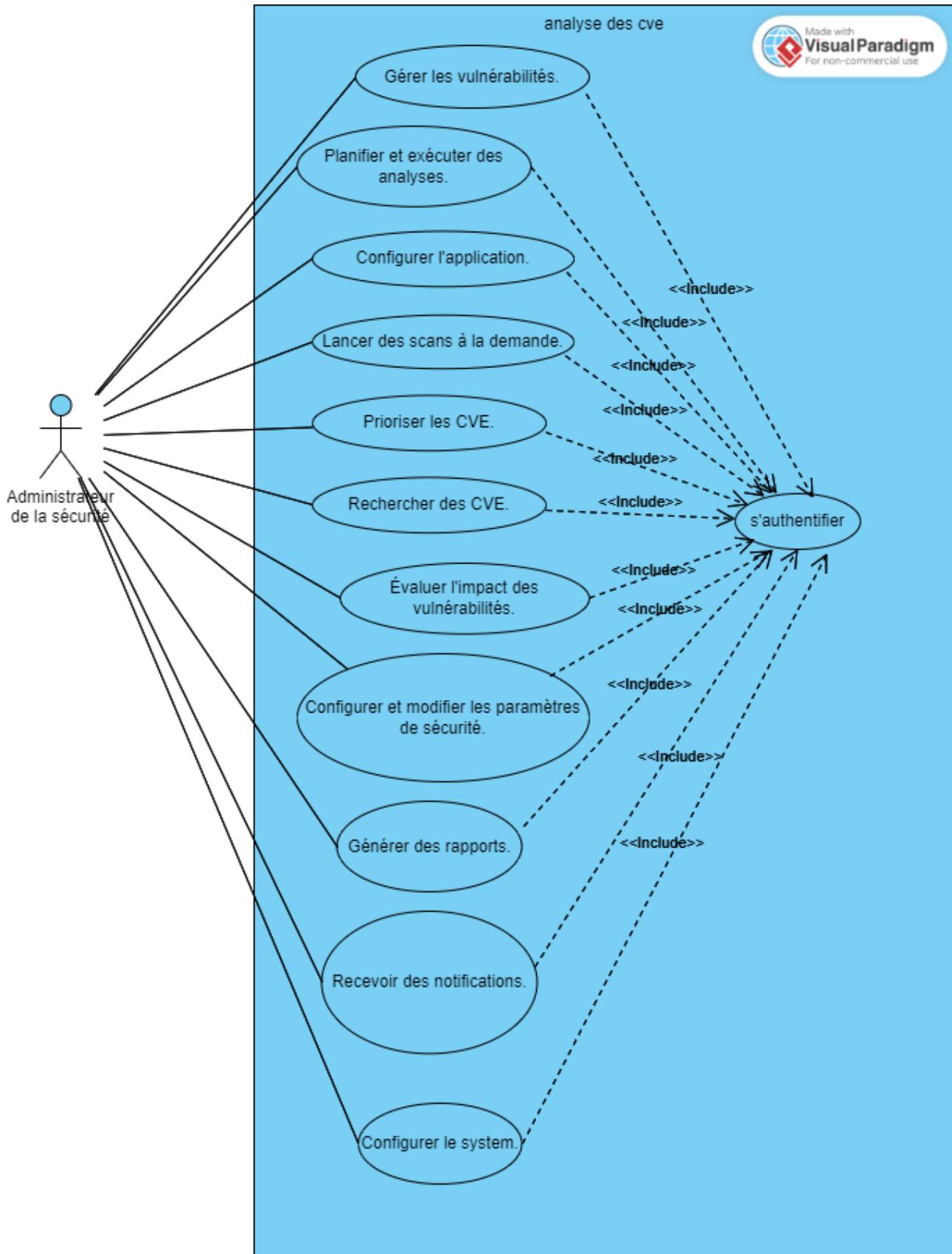
- **Administrateur de la sécurité** L'administrateur de la sécurité est un utilisateur clé de l'application d'analyse des CVE. Ce rôle est généralement occupé par un responsable de la sécurité informatique au sein d'une organisation, chargé de superviser la sécurité des systèmes d'information. L'administrateur de la sécurité a pour mission de protéger les données sensibles de l'entreprise contre les vulnérabilités et les cyberattaques.

##### 2.1.1.2 Spécification des besoins fonctionnels par acteur

- **Administrateur de la sécurité :**

- gérer les vulnérabilités.
- Planifier et exécuter des analyses.
- configurer l'application.
- lancer des scans à la demande.
- prioriser les CVE en fonction de leur risque pour l'entreprise et de la disponibilité des correctifs.
- Rechercher des cve.
- évaluer l'impact potentiel des vulnérabilités détectées.
- configurer et modifier les paramètres de sécurité en fonction des besoins spécifiques de l'entreprise..
- générer des rapports détaillés sur les scans de vulnérabilités, les analyses effectuées, et les mesures d'atténuation recommandées.

### 2.1.1.3 Diagramme de cas d'utilisation global



**FIGURE 2.1 :** Diagramme cas d'utilisation global

Notons qu'un utilisateur peut être un Administrateur de la sécurité dans une entreprise ou un simple utilisateur.

### 2.1.2 Spécification des besoins non fonctionnels

La spécification des besoins ne se limite pas à l'identification des acteurs et à la définition des besoins fonctionnels. D'autres limites doivent être définies pour faciliter l'utilisation, afin de mieux comprendre la structure et la fonctionnalité et assurer une bonne expérience utilisateur.

Étant donné que notre application Web est destinée aux administrateurs de sécurité, ses interfaces doivent être ergonomiques pour faciliter la navigation sur le site et pour que l'utilisateur comprenne les caractéristiques et la structure.

- **La navigation (l'expérience utilisateur)** : Notre application devrait fournir à l'utilisateur le confort de navigation (un bon système de navigation), permettant de minimiser l'efforts pour atteindre la partie qu'il cherche. Ce dernier doit s'éloigner en un seul clic et revenir facilement à la section précédente ou dans la position de départ grâce à un fil d'Ariane (Sidebar).
- **L'accessibilité (responsivité)** : L'accès à l'application couvre un grand nombre de visiteurs avec différents matériels. Cette accessibilité est principalement liée à la taille et à la résolution de l'écran. Pour ce faire, vous devez vous assurer une expérience de lecture idéale pour l'utilisateur, quelle que soit la gamme d'appareil grâce à la responsive design.
- **L'interactivité** : L'échange entre l'application et l'utilisateur doit être simple et facile. Le but principal de ce critère est de surmonter les obstacles afin de garantir une bonne étape de découverte jusqu'à la construction d'une relation entre l'employé et l'administrateur.
- **L'harmonie et la clarté** : Respecter la charte graphique dans notre application est essentiel pour l'harmonie, la cohérence graphique, la visibilité et la lisibilité des textes et du contenu de chaque page.
- **La rapidité** : Le système doit agir rapidement dans les différentes demandes envoyées par les utilisateurs.
- **La sécurité et l'intégrité** :
  - Le système garantit le contrôle d'accès.
  - Chaque utilisateur ne contribue qu'aux pages autorisées par son rôle.

## 2.2 Diagramme de classe

Ci-dessous dans la figure 2.2, nous représentons le diagramme de classe global de notre application. Nous détaillons ses classes dans chaque sprint.

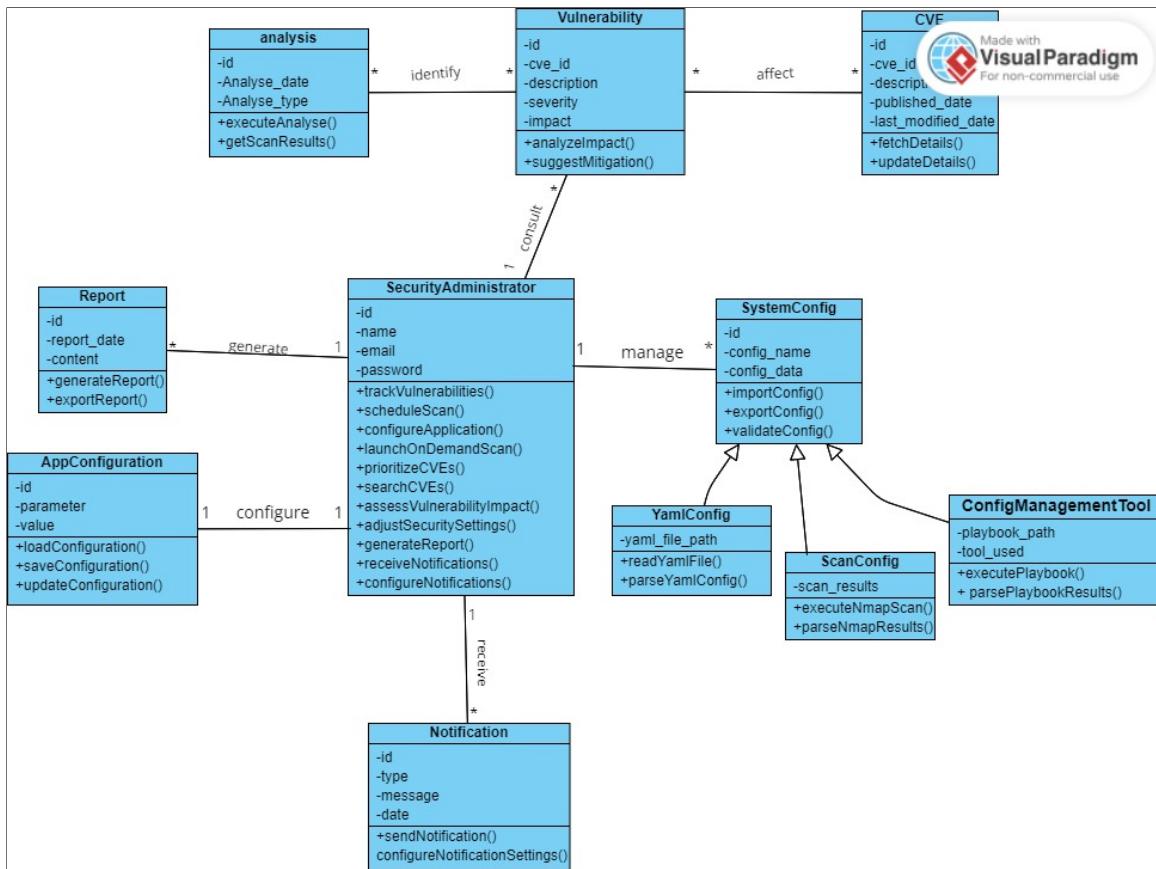


FIGURE 2.2 : Diagramme de classe global

## 2.3 Planification de travail

### 2.3.1 Répartition des releases

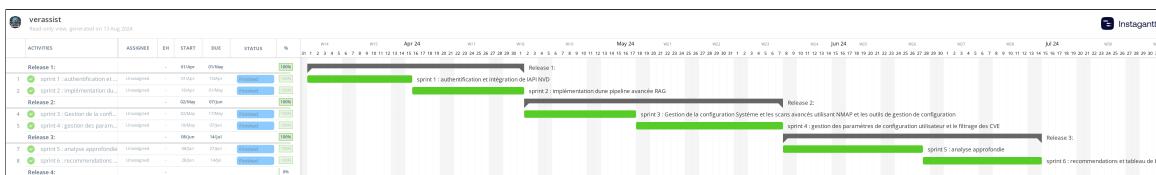
Release ID	Nom du Sprint
1	<ul style="list-style-type: none"> <li>Sprint 1 :authentification et intégration de l'API NVD.</li> <li>Sprint 2 : filtrage des CVE et implémentation d'une pipeline avancée RAG .</li> </ul>
2	<ul style="list-style-type: none"> <li>sprint 3 :Module Gestion de la configuration Système et les scans avancés utilisant NMAP et les outils de gestion de configuration</li> <li>Sprint 4 :Module gestion des paramètres de configuration utilisateur</li> </ul>

Release ID	Nom du Sprint
3	<ul style="list-style-type: none"> <li>Sprint 5 : Module analyse approfondie.</li> <li>Sprint 6 : Recommendations et tableau de bord.</li> </ul>

**TABLEAU 2.1 :** Répartition des releases

### 2.3.2 Planification des sprints

La figure 2.3 représente le diagramme de Gantt illustrant la répartition du travail tout au long de la période de stage.

**FIGURE 2.3 :** Planification des sprints

## 2.4 Architecture de l'application

### 2.4.1 Architecture logique

L'architecture logique de l'application CVE Assessment est conçue pour offrir une solution intégrée et efficace permettant une gestion proactive des vulnérabilités.

L'application se compose de plusieurs composants clés interconnectés : une interface utilisateur intuitive, un service de gestion des CVE, un service d'analyse, un service de recommandation, et une base de données centralisée. L'interface utilisateur permet aux administrateurs de sécurité de configurer et de suivre les vulnérabilités, de planifier des analyses, et de consulter des CVE en temps réel.

Le service de gestion des CVE importe et analyse les vulnérabilités à partir de la base de données nationale des vulnérabilités (NVD), les stocke dans la base de données centrale et les transforme en un graphe à l'aide de la pipeline RAG.

Le service d'analyse exécute des analyses régulières et à la demande, en utilisant des outils comme Nmap, Ansible, et Puppet pour mettre à jour les configurations de sécurité. Enfin, le service envoie des recommandations aux utilisateurs, assurant une réponse rapide aux nouvelles menaces.

Cette architecture modulaire et flexible permet une adaptation facile aux besoins spécifiques des entreprises de toutes tailles, tout en offrant des capacités d'analyse avancées grâce à l'intégration de technologies d'IA générative.

La figure 2.4 représente l'architecture logique de l'application Vulnerability Assessment

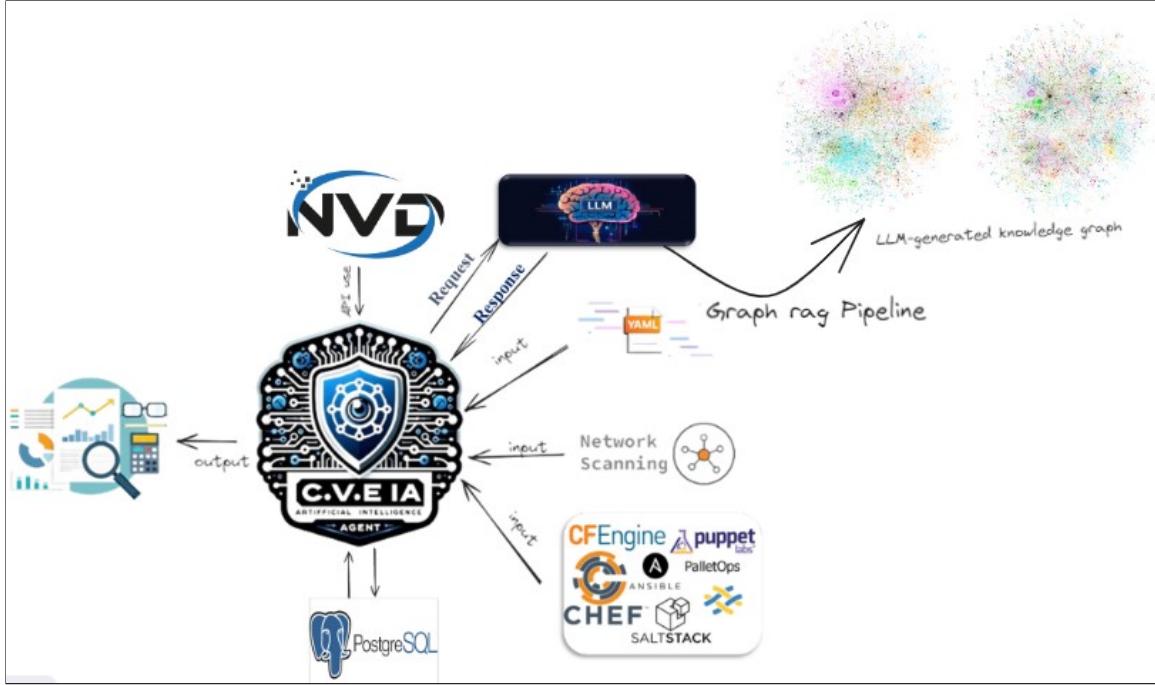


FIGURE 2.4 : Architecture logique

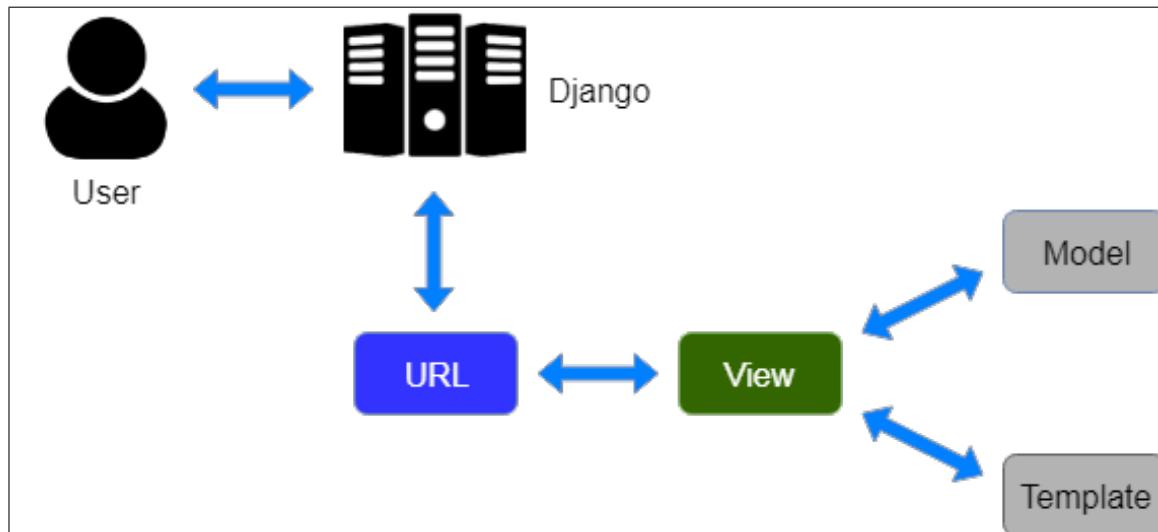
#### 2.4.1.1 Architecture logique et physique de django avec l'application

Django est un framework web de haut niveau écrit en Python qui encourage le développement rapide et une conception propre et pragmatique. Voici une description de l'architecture logique de Django :

- Client :
  - **Description :** La partie visible de l'application web, généralement exécutée dans un navigateur.
  - **Interaction :** Envoie des requêtes (GET, POST, etc.) au serveur et reçoit les réponses sous forme de pages HTML, images, CSS, JavaScript, etc.
- Serveur :
  - **Framework Django (Python) :** C'est le cœur du serveur qui traite les requêtes des clients et génère les réponses appropriées.
- Composants du framework Django :
  - **urls.py :** Gère le routage des URL. Chaque URL est associée à une vue spécifique, Analyse les URL entrantes et détermine quelle vue doit être appelée.
  - **views.py :** Contient les fonctions ou les classes qui gèrent la logique métier de l'application, Exécute des fonctions associées pour traiter les requêtes et renvoyer des réponses.
  - **models.py :** Définit les modèles de données de l'application, Interagit avec la base de données via une interface ORM (Object-Relational Mapping).

- **templates.py** : Contient les gabarits (templates) utilisés pour générer le HTML des pages, Génère des pages HTML dynamiques en combinant des données et des gabarits.
- Base de données :
  - **Description** : Stocke les données de l'application.
  - **Interaction** : Interagit avec models.py pour enregistrer, mettre à jour, supprimer et récupérer des données.
- Fonctionnement général :
  - **Requête** : Le client envoie une requête HTTP au serveur (par exemple, une requête GET pour afficher une page ou une requête POST pour soumettre un formulaire).
  - **Traitement de la requête** : Django utilise urls.py pour faire correspondre l'URL de la requête à une vue spécifique dans views.py. La vue traite la requête, souvent en interagissant avec les modèles dans models.py pour accéder aux données.
  - **Réponse** : La vue utilise un gabarit de templates.py pour générer une page HTML. Django renvoie la réponse HTML au client, qui l'affiche dans le navigateur.

La figure 2.5 représente l'architecture logique de Django avec l'application :



**FIGURE 2.5** : Architecture logique de Django avec l'application [2]

## 2.5 Les patrons de conception utilisés

### 2.5.1 Patrons de création

#### 2.5.1.1 Patron Singleton

Singleton est un patron de conception de création qui garantit que l'instance d'une classe n'existe qu'en un seul exemplaire, tout en fournissant un point d'accès global à cette instance.[3] :

On utilise ce patron souvent en utilisant spring boot lors de l'injection de dépendance avec l'annotation Autowired ou bien Inject.

### 2.5.2 Patrons structurels

#### 2.5.2.1 Patron Proxy

Le Proxy est un patron de conception structurel qui vous permet d'utiliser un substitut pour un objet. Elle donne le contrôle sur l'objet original, vous permettant d'effectuer des manipulations avant ou après que la demande ne lui parvienne. Ce patron de conception vous propose de créer une classe Proxy qui a la même interface que l'objet du service original. Vous passez ensuite l'objet procuration à tous les clients de l'objet original. Lors de la réception d'une demande d'un client, la procuration crée l'objet du service original et lui délègue la tâche.[4] :

Nous avons utilisé ce patrons plusieurs fois dans la partie backend avec les classes DTO Data Transfer Object.

### 2.5.3 Patrons comportementaux

#### 2.5.3.1 Patron État

État est un patron de conception comportemental qui permet de modifier le comportement d'un objet lorsque son état interne change. L'objet donne l'impression qu'il change de classe.[5] :

Nous avons utilisé ce patrons lors du traitements des cves, chacun posséde un état et il change de comportement dès qu'on change son état.

### 2.5.4 Patron d'architecture

#### 2.5.4.1 Patron Data Access Object DAO

C'est un patron qui permet d'encapsuler et de centraliser l'accès à la base de données et le lien entre l'application et le système de stockage. Le plus grand avantage de ce patron est qu'il permet de mieux maîtriser les changements susceptibles d'être opérés sur le système de stockage à savoir une migration d'un système à un autre. Un objet DAO fournit des opérations basiques (CRUD) comme la lecture, la mise à jour, la création, l'affichage et la suppression d'une entité sans exposer les détails de la base de données.

## 2.6 Difficultés rencontrées

Lors du développement de l'application de gestion des CVE, nous avons rencontré plusieurs défis significatifs, principalement liés à l'analyse des CVE par rapport aux configurations système et à

l'utilisation des modèles de langage naturel (LLM) par rapport à sa contexte Windows.

**Taux élevé des CVE à analyser** L'un des principaux défis est le volume élevé des CVE à analyser. Chaque jour, de nouvelles vulnérabilités sont découvertes et enregistrées dans la base de données nationale des vulnérabilités (NVD). Analyser ce flux constant de CVE en temps réel pour chaque configuration système spécifique représente une tâche colossale. La complexité augmente d'autant plus lorsque les configurations système varient largement entre les environnements, rendant difficile l'identification rapide des vulnérabilités pertinentes.

**Limitations des LLM dans le contexte Windows** L'utilisation des modèles de langage naturel (LLM) pour analyser les CVE et les configurations système a également posé des défis. Les LLM, bien que puissants, ont une fenêtre de contexte limitée, ce qui signifie qu'ils ne peuvent traiter qu'une quantité limitée d'informations à la fois. Dans le contexte de Windows, où les configurations et les journaux peuvent être volumineux, cette limitation devient un obstacle majeur. Les LLM doivent être capables de gérer de grandes quantités de données et de les analyser efficacement, ce qui est difficile à réaliser avec les contraintes actuelles.

**Solution : Utilisation des graphes pour le filtrage sémantique** Pour surmonter ces défis, nous avons intégré l'utilisation de graphes dans notre pipeline de traitement des CVE. Les graphes nous permettent de représenter les relations sémantiques entre les CVE et les configurations système de manière plus efficace. En utilisant des graphes, nous pouvons filtrer et identifier les CVE les plus pertinentes et sémantiquement proches des configurations système spécifiques. Cette approche nous permet de réduire le volume de données à analyser pour chaque configuration et d'améliorer la précision des recommandations de sécurité.

**Conclusion** Les difficultés rencontrées dans l'analyse des CVE par rapport aux configurations système et l'utilisation des LLM dans un contexte Windows nous ont poussés à innover et à adopter des solutions plus efficaces. L'intégration des graphes pour le filtrage sémantique des CVE est une de ces solutions, permettant de surmonter les limitations des LLM et de gérer efficacement le volume élevé de CVE.

## 2.7 Environnement de travail

### 2.7.1 Outils de gestion de projet

- **GitHub**

GitHub est une plateforme de développement bien connue qui offre également des fonctionnalités

de gestion de projet robustes. Les développeurs peuvent utiliser GitHub pour gérer les dépôts de code, suivre les problèmes (issues), et collaborer efficacement au sein d'équipes de toutes tailles. Les principales fonctionnalités de gestion de projet incluent les boards de projet, qui permettent d'organiser le travail en colonnes représentant les phases de projet ou les types de tâches, facilitant ainsi la visualisation du statut du projet. Les issues et les pull requests permettent de suivre les tâches, les bugs et les demandes de fonctionnalités. De plus, GitHub s'intègre avec divers outils tiers comme ZenHub, Codetree, et Slack, augmentant ainsi ses capacités de gestion de projet.



FIGURE 2.6 : Logo GitHub

- **Jira Software**

Jira est une plateforme de gestion de projet destinée au développement logiciel. Elle permet de préparer, matérialiser et suivre les développements des teams SCRUM.[6]

la figure 2.7 montre une capture de jira qui contient les tâches terminées et les tâches en cours de sprint 1.



FIGURE 2.7 : Logo Bitbucket

- **Discord**

Discord permet aux membres de l'équipe de discuter entre eux en tête-à-tête ou en groupe via un serveur. Nous l'utilisons pour envoyer des messages directs, passer des appels vidéo, discuter avec la voix et même partager l'écran.[7]



FIGURE 2.8 : Logo Discord

### 2.7.2 Framework de développement et de tests

- **Django**

Django est un framework web de haut niveau qui facilite le développement rapide et propre des applications web. En termes de gestion de projet, Django offre plusieurs avantages notables. L'ORM intégré simplifie la gestion des bases de données en utilisant des modèles Python, tandis que l'interface administrateur permet une gestion facile des utilisateurs, des permissions et des données via une interface utilisateur prête à l'emploi. L'architecture Modèle-Vue-Template (MVT) de Django encourage une séparation claire des préoccupations, améliorant ainsi la maintenabilité et l'évolutivité du projet.



**FIGURE 2.9 :** Logo Django

- **Pytest**

Pytest est un cadre de test Python qui permet d'écrire des cas de test concis et lisibles. Pour l'utiliser avec Django, il faut installer la bibliothèque pytest-django qui intègre Pytest avec le framework Django. Pytest simplifie la création et la gestion des tests, offre des fixtures pour gérer les configurations de test et permet de tester les vues, les modèles et les API de Django de manière efficace. Par exemple, vous pouvez utiliser le client de test Django dans Pytest pour simuler des requêtes HTTP et vérifier les réponses, créant ainsi une suite de tests robuste et facile à maintenir.



**FIGURE 2.10 :** Logo pytest

- **Microsoft Graph RAG**

Microsoft propose le framework Graph RAG (Retrieval-Augmented Generation) pour améliorer

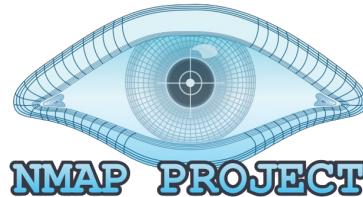
l'analyse des données complexes. Ce framework utilise des graphes pour représenter et analyser les relations sémantiques entre les données, ce qui est particulièrement utile pour filtrer et identifier les CVE les plus pertinentes par rapport aux configurations système. Cela permet de surmonter les limitations des modèles de langage naturel (LLM) en offrant une analyse plus précise et contextuelle.



**FIGURE 2.11 :** Logo graph

- **Nmap**

Nmap est un outil open-source utilisé pour la découverte de réseau et l'audit de sécurité. Il permet de scanner les réseaux pour identifier les hôtes actifs et les services sur un réseau, ainsi que leurs configurations. Nmap peut également détecter des vulnérabilités spécifiques en utilisant des scripts Nmap, et ses résultats peuvent être intégrés facilement dans des systèmes plus larges de gestion de sécurité pour une analyse continue.



**FIGURE 2.12 :** Logo nmap

- **Ansible**

Ansible est un outil open-source de gestion de configuration et d'automatisation qui facilite la gestion de l'infrastructure informatique. Ses fonctionnalités incluent les playbooks, des scripts simples écrits en YAML pour automatiser les tâches de configuration et de déploiement. Ansible fonctionne sans nécessiter d'agents sur les hôtes gérés, simplifiant ainsi la gestion. Il peut gérer des infrastructures de toute taille, de quelques machines à des milliers de noeuds, offrant ainsi une grande scalabilité.



**FIGURE 2.13 :** Logo ansible

### 2.7.3 Système de gestion de base de données

- **PostgreSQL**

PostgreSQL est un système de gestion de base de données relationnelle open-source reconnu pour sa robustesse et ses fonctionnalités avancées. Il assure la fiabilité des transactions de base de données grâce à la conformité ACID, permet l'ajout de nouvelles fonctionnalités via des extensions, et offre des contrôles d'accès granulaires ainsi que des mécanismes de sécurité avancés, essentiels pour les applications de gestion des CVE.



**FIGURE 2.14 :** Logo postgresql

## Conclusion

La phase de spécification des besoins est une phase très importante dans le cycle de vie d'un projet, car elle offre une vue plus claire du système et des principales caractéristiques à atteindre. Par conséquent, nous avons introduit le backlog produit et la planification des releases pour passer à la phase de conception.

# RELEASE 1 : AUTHENTIFICATION ET RÉCUPÉRATION DES CVE

---

## Plan

1	Sprint 1 : authentification et intégration de l'API NVD . . . . .	29
2	sprint 2 : filtrage des CVE et implémentation d'une pipeline avancée RAG	35

## Introduction

Au cours de ce chapitre, nous allons présenter les différentes étapes de réalisation du premier sprint "authentification et intégration de l'API NVD", et du deuxième sprint "implémentation d'une pipeline avancée RAG".

### 3.1 Sprint 1 : authentification et intégration de l'API NVD

Dans cette section nous allons présenter les différents étapes de la réalisation du premier sprint.

#### 3.1.1 Objectifs du sprint 1

L'objectif principal de ce sprint est de mettre en place un système d'authentification sécurisé pour les utilisateurs, permettant ainsi un accès contrôlé à l'application. Parallèlement, ce sprint vise à intégrer l'API de la National Vulnerability Database (NVD) afin de récupérer automatiquement les CVE (Common Vulnerabilities and Exposures) les plus récentes. Cette intégration permettra de stocker ces données dans la base de données centrale de l'application, facilitant ainsi leur gestion et leur analyse. De plus, la configuration initiale des paramètres de sécurité sera mise en place pour garantir une utilisation sûre et efficace du système.

#### 3.1.2 Backlog du sprint 1

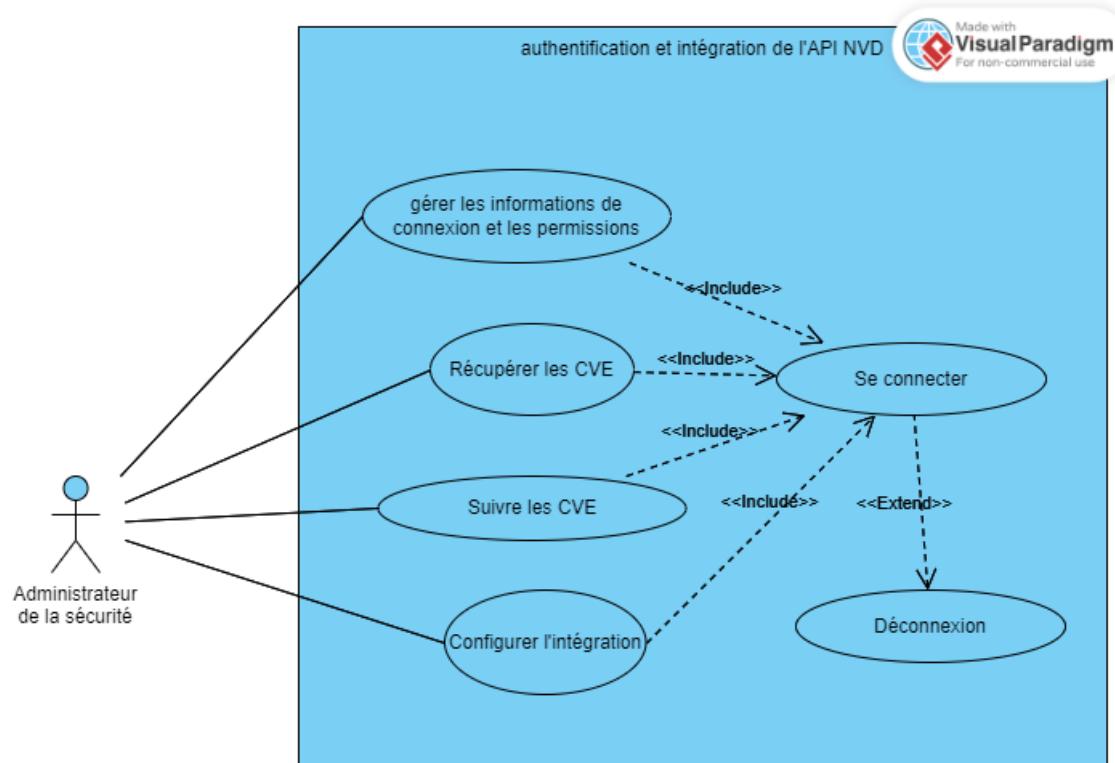
Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant qu'administrateur système, je veux pouvoir me connecter à l'application avec un nom d'utilisateur et un mot de passe afin de sécuriser l'accès à l'application.	1	3
2	En tant qu'administrateur système, je veux pouvoir gérer les informations de connexion et les permissions afin de contrôler l'accès à l'application.	2	3
3	En tant qu'administrateur système, je veux récupérer et suivre toutes les CVE avec toutes les informations nécessaires et les stocker dans la base de données centrale	1	4

Id	Fonctionnalités	Priorité	Estimation (Jour)
4	En tant qu'administrateur système, je veux être certain que toutes les actions sont sécurisées et les informations protégées	1	2
5	En tant qu'administrateur système, je veux pouvoir configurer des tâches de récupération automatique des CVE afin de m'assurer que les données sont toujours à jour sans intervention manuelle.	2	1
6	En tant qu'administrateur système, je veux pouvoir afficher un tableau de bord récapitulatif des CVE récupérées pour avoir une vue d'ensemble rapide des vulnérabilités .	3	1

**TABLEAU 3.1** : Backlog du Sprint 1

### 3.1.3 Spécification des besoins fonctionnels

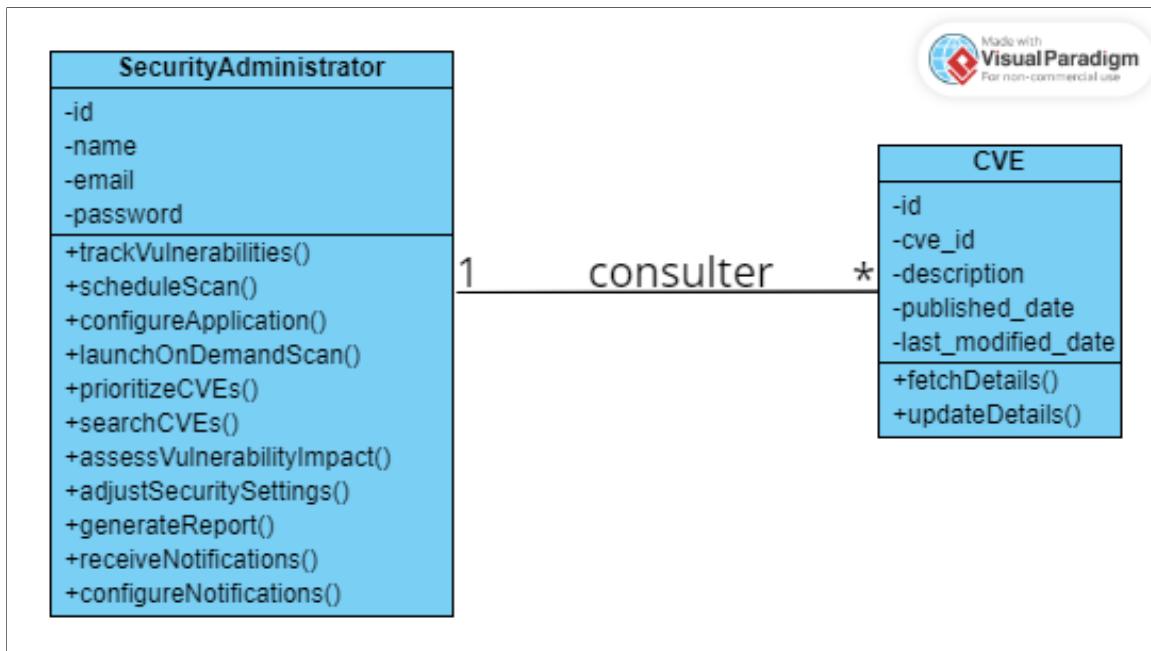
La figure 3.1 représente le diagramme cas d'utilisation du sprint "authentification et intégration de l'API NVD"

**FIGURE 3.1 :** Diagramme de cas d'utilisation "authentification et intégration de l'API NVD"

### 3.1.4 Diagramme de classe

La figure 3.2 représente le diagramme de classe du sprint "authentification et intégration de l'API NVD"

- Le diagramme de classe pour le premier sprint montre l'interaction entre l'administrateur de la sécurité et les CVE. L'administrateur de la sécurité, représenté par la classe SecurityAdministrator, a la capacité de consulter plusieurs CVE, illustrée par la classe CVE. Chaque CVE contient des informations spécifiques telles que l'identifiant, la description et les dates importantes. La relation entre ces deux classes est une association qui permet à l'administrateur de consulter et de suivre les détails des CVE récupérées. Cette configuration est essentielle pour garantir que l'administrateur dispose de toutes les informations nécessaires pour gérer efficacement les vulnérabilités dans le système.



**FIGURE 3.2** : Diagramme de classe "authentification et intégration de l'API NVD"

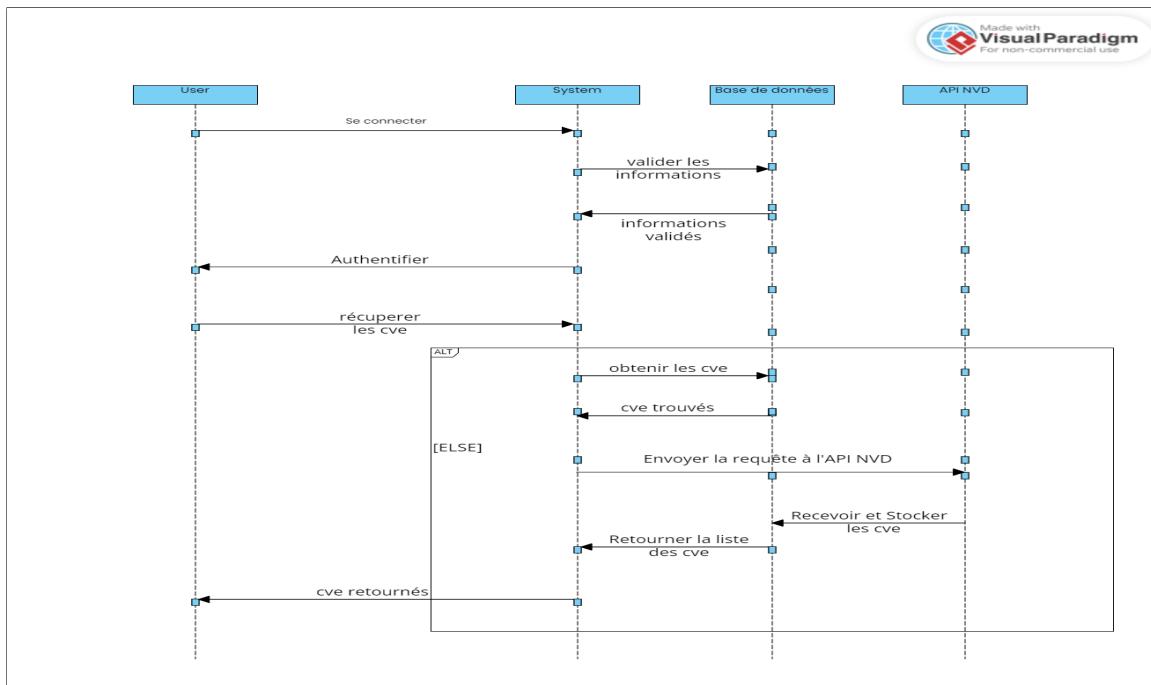
### 3.1.5 Diagrammes dynamiques

Dans cette section nous allons présenter les diagrammes UML dynamique pour ce sprint.

#### 3.1.5.1 Diagramme de séquence objet "authentification et intégration de l'API NVD"

La figure 3.2 illustre le diagramme de séquence objet de l'inscription.

Le diagramme de séquence objet pour le premier sprint montre le processus d'authentification de l'utilisateur et de récupération des CVE. D'abord, l'utilisateur envoie ses informations de connexion au système, qui les valide en les comparant avec celles stockées dans la base de données. Une fois authentifié, l'utilisateur peut demander la récupération des CVE. Le système vérifie si les CVE sont déjà présentes dans la base de données ; sinon, il envoie une requête à l'API NVD(en arrière plan) pour obtenir les dernières CVE. Les CVE récupérées sont ensuite stockées dans la base de données, et la liste des CVE est retornnée à l'utilisateur. Ce processus garantit que les informations de vulnérabilité sont toujours actuelles et accessibles pour l'utilisateur.



**FIGURE 3.3 :** Diagramme de séquence objet "ajout d'entreprise"

### 3.1.6 Réalisation

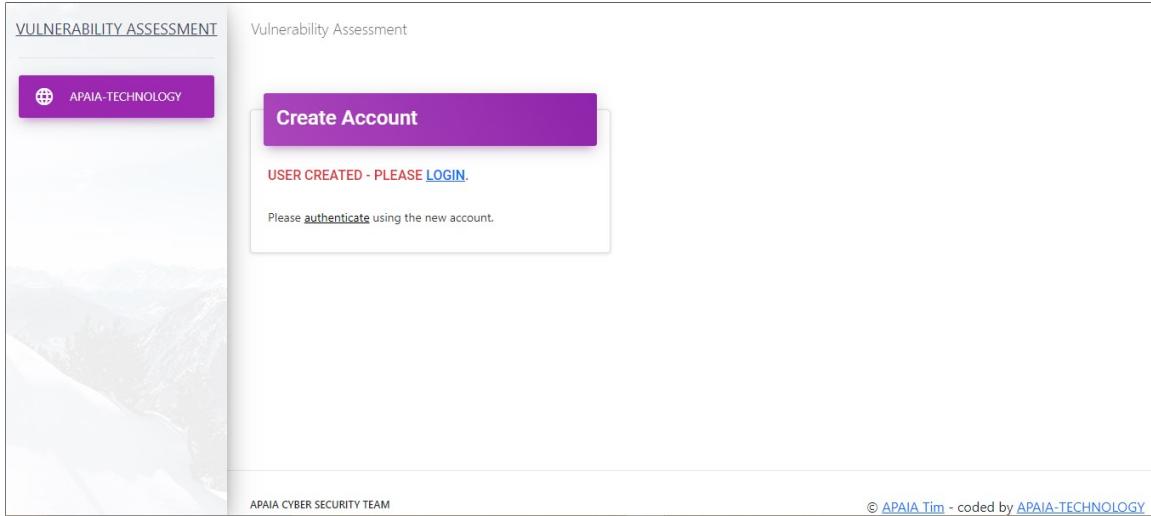
Pour mieux comprendre le fonctionnement de notre projet, nous allons présenter les différents fonctions de l'application en se basant sur un scénario.

Dans ce contexte pour s'inscrire à l'application monsieur "alaeddine" doit remplir le formulaire d'inscription . Les trois premières figures nous montrent le formulaire d'inscription et la redirection vers l'authentification (login).

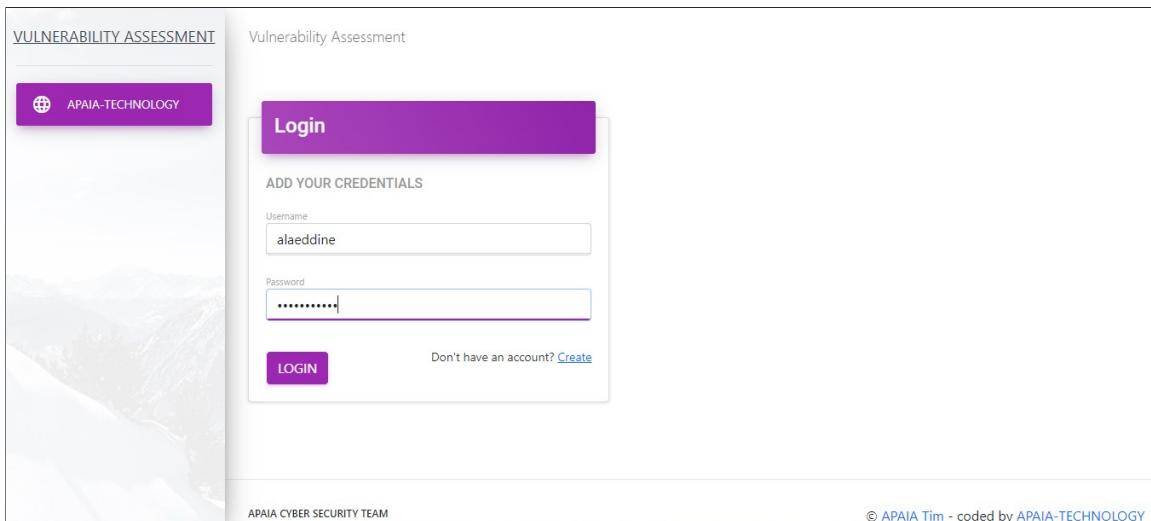
The screenshot shows the 'Create Account' form on a website. The header says 'VULNERABILITY ASSESSMENT' and 'APAIA-TECHNOLOGY'. The form has a purple header 'Create Account' and a purple footer 'REGISTER'. It asks for 'ADD YOUR CREDENTIALS' with fields for 'Username' (alaeddine), 'Email' (alaeddinemansouri2@gmail.com), 'Password', and 'Password Check'. Below the form is a link 'Have an account? [Login](#)'.

**FIGURE 3.4 :** interface de création de compte

Après l'inscription, l'application emmène l'utilisateur vers la page de login comme nous illustre les prochaines figures.



**FIGURE 3.5 :** Page login 1.0



**FIGURE 3.6 :** Page login 1.1

La prochaine capture dans la figure 3.7. nous montrer que la page "CVEs" offre la possibilité de mettre en place une interface utilisateur intuitive pour la consultation des cve. L'administrateur système peut consulter les CVE récupérées depuis l'API NVD et suivre les détails de chaque vulnérabilité, tels que l'identifiant CVE, la description, les dates de publication et de dernière modification, ainsi que les scores de sévérité. La fonctionnalité de déconnexion (logout) a été implémentée pour assurer une sécurité optimale en permettant à l'utilisateur de terminer sa session en toute sécurité. Cette première réalisation établit les bases essentielles pour une gestion efficace et sécurisée des vulnérabilités au sein de l'organisation. .

The screenshot shows a web-based vulnerability management system. On the left, a sidebar menu includes options like Home, Configuration, CVEs (which is selected and highlighted in purple), Dashboard, Analysis, CVE Vulnerability Overview, and APAIA-TECHNOLOGY. The main content area has a purple header titled 'CVE Details' with a sub-header 'Explore the latest Common Vulnerabilities and Exposures to stay informed about potential security threats.' Below this is a table with columns: CVE\_ID, Description, Published, Last Modified, CVSS v2 Score, and CVSS v2 Score. Three rows of CVE details are listed:

CVE_ID	Description	Published	Last Modified	CVSS v2 Score	CVSS v2 Score
CVE-2024-3249	The Zeta Elementor Site Library plugin for WordPress is vulnerable to unauthorized modification of data due to a missing capability ...	2024-06-25	2024-06-25	4.30	0.00
CVE-2024-5431	The WFCafe - Online Food Ordering, Restaurant Menu, Delivery, and Reservations for WooCommerce plugin for WordPress is vulnerable to Local ...	2024-06-25	2024-06-25	8.80	0.00
CVE-2024-4759	The Mime Types Extended WordPress plugin through 0.11 does not sanitise uploaded SVG files, which could allow users with a ...	2024-06-25	2024-06-25	0.00	0.00

**FIGURE 3.7 :** interface CVEs

## 3.2 sprint 2 : filtrage des CVE et implémentation d'une pipeline avancée RAG

Dans cette section nous allons présenter les différents étapes permettant la réalisation du sprint "filtrage des CVE et implémentation d'une pipeline avancée RAG".

### 3.2.1 Objectifs du sprint 2

L'objectif principal de ce sprint est de mettre en place un système de filtrage des CVE et d'implémenter une pipeline avancée RAG (Retrieval-Augmented Generation). Cette pipeline est essentielle pour l'indexation des CVE, les transformant en graphes afin de mieux comprendre les relations sémantiques entre les différentes vulnérabilités. En les organisant sous forme de graphes, le système peut détecter des menaces cachées et établir des connexions entre différentes CVE, facilitant ainsi une analyse plus profonde et pertinente. Ce processus permet non seulement de prioriser les CVE les plus pertinentes pour le système, mais aussi de révéler des vulnérabilités potentiellement liées qui pourraient autrement passer inaperçues.

### 3.2.2 Backlog du sprint 2

<b>Id</b>	<b>Fonctionnalités</b>	<b>Priorité</b>	<b>Estimation (Jour)</b>
-----------	------------------------	-----------------	--------------------------

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant qu'administrateur système, je veux pouvoir filtrer les CVE récupérées et les trier par ordre croissant ou décroissant selon la date de publication, la date de dernière modification, et les scores CVSS v3 et v2.	1	4
2	En tant qu'administrateur système, je veux pouvoir filtrer les CVE affichées en sélectionnant une plage de dates de début et de fin pour affiner les résultats	1	4
3	En tant qu'administrateur système, je veux que la pipeline RAG indexe automatiquement les CVE en arrière-plan et les transforme en graphes pour révéler les relations sémantiques et les menaces cachées.	1	5
4	En tant qu'administrateur système, je veux pouvoir visualiser les relations sémantiques entre les CVE sous forme de graphe, afin d'identifier les clusters de vulnérabilités et comprendre les menaces cachées	3	2

**TABLEAU 3.2 :** Backlog du Sprint 2

### 3.2.3 Technologies utilisées

Dans le cadre de la gestion avancée des CVE, deux technologies clés ont été utilisées : le Graph RAG Framework de Microsoft et Neo4j pour la visualisation :

- **Graph RAG**

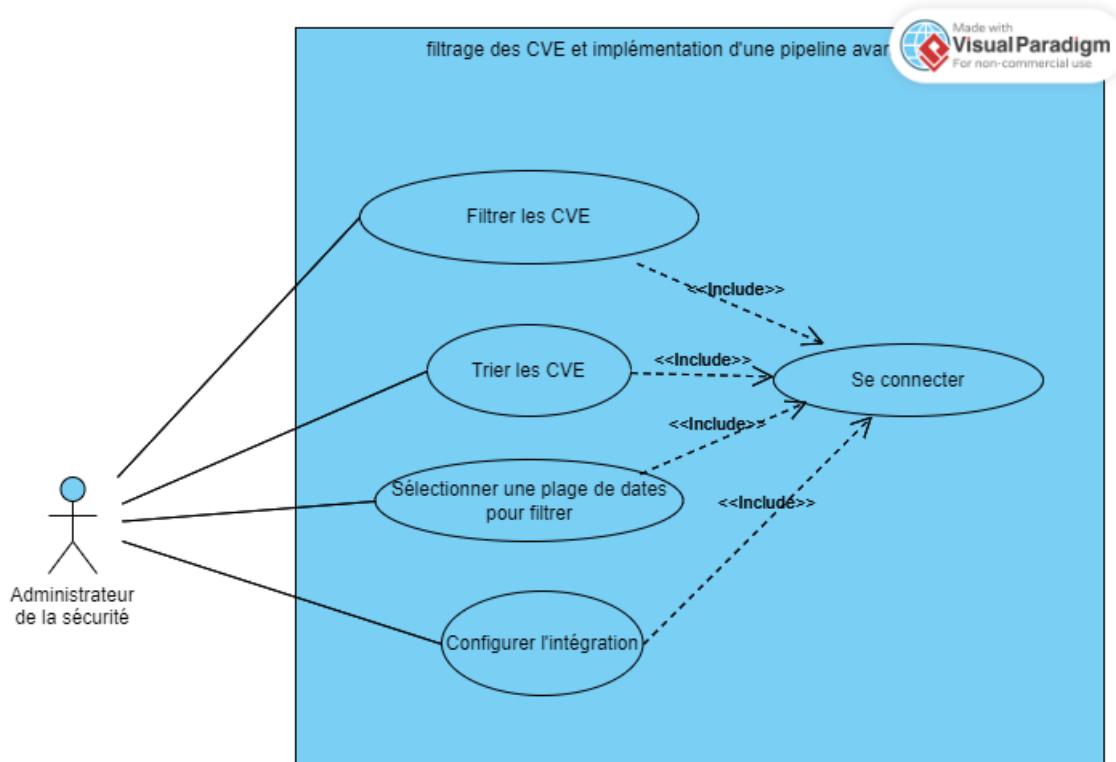
Le Graph RAG Framework de Microsoft est un outil puissant qui permet de transformer les informations textuelles en graphes relationnels. Ce framework est essentiel pour indexer les CVE et comprendre les relations complexes entre les différentes vulnérabilités. En convertissant les données en graphes, il devient plus facile de détecter des connexions cachées et d'identifier des menaces potentielles au sein du système.

- **Neo4j**

Pour la visualisation de ces graphes, Neo4j a été utilisé. Neo4j est une base de données orientée graphes qui excelle dans la modélisation, l'exploration et la visualisation des relations entre les données. En intégrant Neo4j, les administrateurs peuvent visualiser les relations sémantiques entre les CVE, ce qui permet une meilleure compréhension des vulnérabilités et une prise de décision plus éclairée. Cette combinaison d'outils offre une solution robuste pour la gestion proactive des vulnérabilités, en mettant l'accent sur la détection des relations cachées et la visualisation claire des menaces potentielles.

### 3.2.4 Spécification des besoins fonctionnels

La figure 3.8 représente le diagramme cas d'utilisation du sprint "filtrage des CVE et implémentation d'une pipeline avancée RAG"



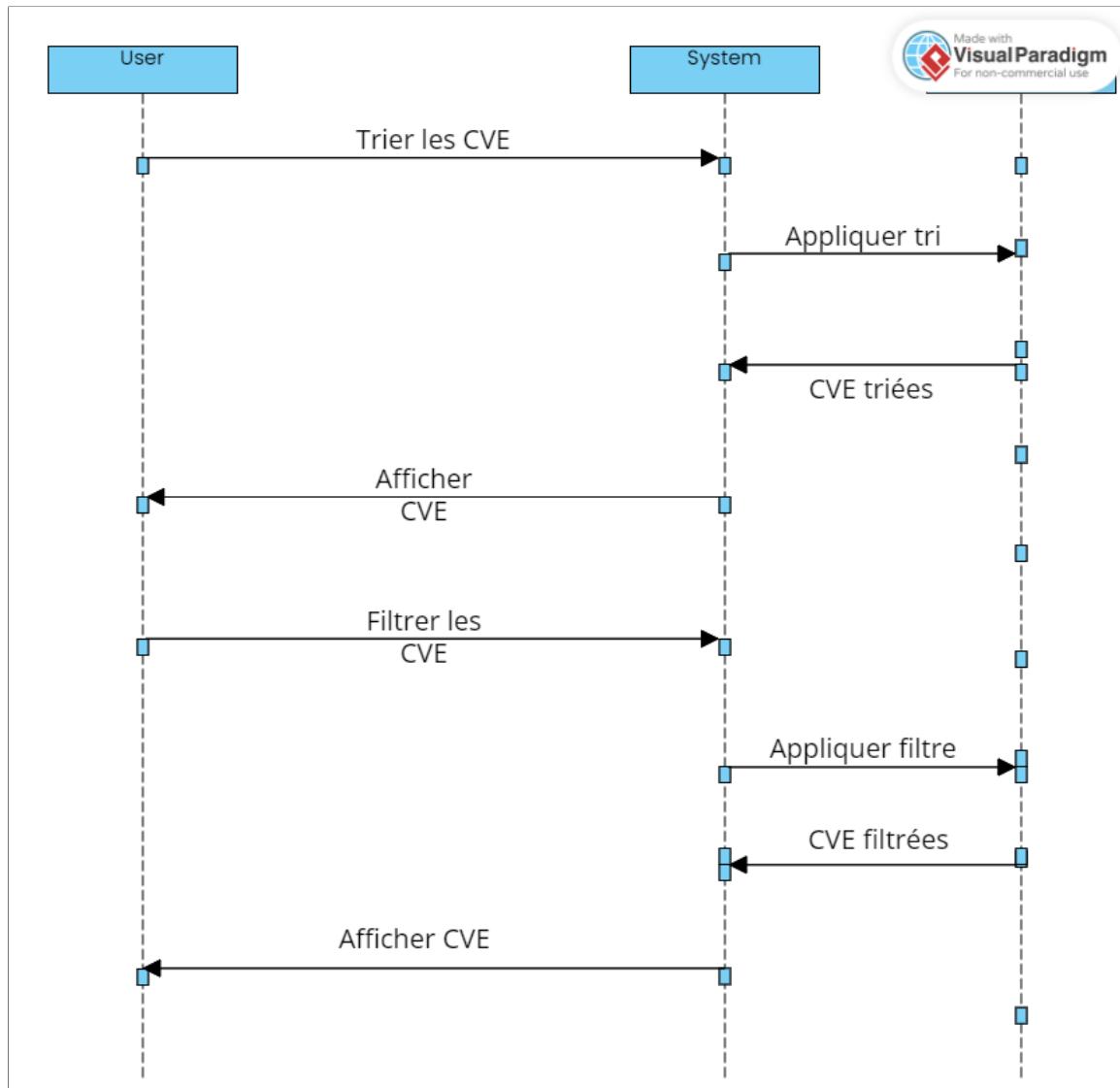
**FIGURE 3.8 :** Diagramme de cas d'utilisation "filtrage des CVE et implémentation d'une pipeline avancée RAG"

### 3.2.5 Diagrammes dynamiques

#### 3.2.5.1 Diagramme de séquence objet "Tri et filtrage des cve"

La figure 3.9 représente le diagramme de séquence objet de Tri et filtrage des cve.

Dans ce diagramme nous illustrons les différents interactions entre l'utilisateur et les composants de notre projet.



**FIGURE 3.9 :** Diagramme de séquence objet "Tri et filtrage des cve"

### 3.2.6 Réalisation

La figure 3.10 représente la page CVE de l'application qui nous permet de Trier les cves par ordre croissant ou décroissant selon la date de publication, la date de dernière modification, et les scores CVSS v3 et v2.

The screenshot shows the 'Vulnerability Assessment' application. On the left is a sidebar with the following navigation items:

- Home
- Configuration
- CVEs** (highlighted in purple)
- Dashboard
- Analysis
- CVE Vulnerability Overview
- APAIA-TECHNOLOGY
- Logout

The main content area is titled 'CVE Details' with the sub-instruction 'Explore the latest Common Vulnerabilities and Exposures to stay informed about potential security threats.' Below this is a table with the following data:

CVE ID	Description	Published	Last Modified	CVEs v3 Score	CVEs v2 Score
CVE-1999-0001	ip_input.c in BSD-derived TCP/IP implementations allows remote attackers to cause a denial of service (crash or hang) via crafted packets.	1999-12-30	2010-12-16	0.00	5.00
CVE-1999-0002	Buffer overflow in NFS mounted gives root access to remote attackers, mostly in Linux systems.	1998-10-12	2009-01-26	0.00	10.00
CVE-1999-0003	Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd).	1998-04-01	2018-10-30	0.00	10.00

**FIGURE 3.10 :** interface CVE

La fonctionnalité de recherche des CVE permet aux utilisateurs de trouver rapidement et efficacement les cve spécifiques dans le système.

The screenshot shows the 'Vulnerability Assessment' application. On the left is a sidebar with the following navigation items:

- Home
- Configuration
- CVEs** (highlighted in purple)
- Dashboard
- Analysis
- CVE Vulnerability Overview
- APAIA-TECHNOLOGY
- Logout

The main content area has a search bar at the top containing the value '2023'. Below it is a table with the following data:

CVE ID	Description	Published	Last Modified	CVEs v3 Score	CVEs v2 Score
CVE-2023-5038	badmonkey, a Security Researcher has found a flaw that allows for a unauthenticated DoS attack on the camera. An attacker ...	2024-06-25	2024-06-25	0.00	0.00
CVE-2023-6198	Use of Hard-coded Credentials vulnerability in Baicells Snap Router BaiCE_EMI on EP3011 (User Passwords modules) allows unauthorized access to the ...	2024-06-25	2024-06-25	9.30	0.00
CVE-2023-5037	badmonkey, a Security Researcher has found a flaw that allows for a authenticated command injection on the camera. An attacker ...	2023-11-13	2024-06-25	0.00	0.00

**FIGURE 3.11 :** interface de recherche des cve

l'utilisateur aussi a la main de filtrer les CVE et les affichées en sélectionnant une plage de dates de début et de fin , afin d'affiner la visualisation .

The screenshot shows a 'Vulnerability Assessment' interface. On the left, a sidebar menu includes 'Home', 'Configuration', 'CVEs' (which is highlighted in purple), 'Dashboard', 'Analysis', 'CVE Vulnerability Overview', 'APAIA-TECHNOLOGY', and 'Logout'. The main area displays 'CVE Details' with three entries listed:

- CVE-1999-0001: ip\_input.c in BSD-derived TCP/IP implementations allows remote attackers to cause a denial of service (crash or hang) via crafted packets.
- CVE-1999-0002: Buffer overflow in NFS mounted gives root access to remote attackers, mostly in Linux systems.
- CVE-1999-0003: Execute commands as root via buffer overflow in Tooltalk database server (rpc.ttdbserverd).

A 'Filters' overlay on the right side of the screen allows users to search by year, start date, and end date, with a button to 'APPLY FILTERS'.

FIGURE 3.12 : Filtrer les CVE par plage de dates

les CVE soient automatiquement indexées en arrière-plan et transformées en graphes pour révéler les relations sémantiques et les menaces cachées, ce qui permet une meilleure compréhension des vulnérabilités et une réponse plus rapide.

The screenshot shows the Neo4j browser interface. On the left, there's a sidebar with 'Database Information' showing 'neo4j' as the database, 'Node labels' (Community, Entity, Node, Relationship, TextUnit), 'Relationship types' (HAS\_TEXT\_UNIT, RELATES\_TO, REPORTS\_ON), and 'Property keys' (community, degree, description, document\_ids, entity\_ids, entity\_type, findings, full\_content, full\_content\_json, human\_readable\_id, id, level, n\_tokens, name, rank, rank\_explanation, raw\_community). The main area shows a graph visualization with many red nodes and connections. The Neo4j shell at the bottom has a command history with '\$ :play start' and a status bar indicating 'Getting started with Trv Neo4i with live data' and 'Cypher basics'.

FIGURE 3.13 : visualisation en graph après l'indexation des cve

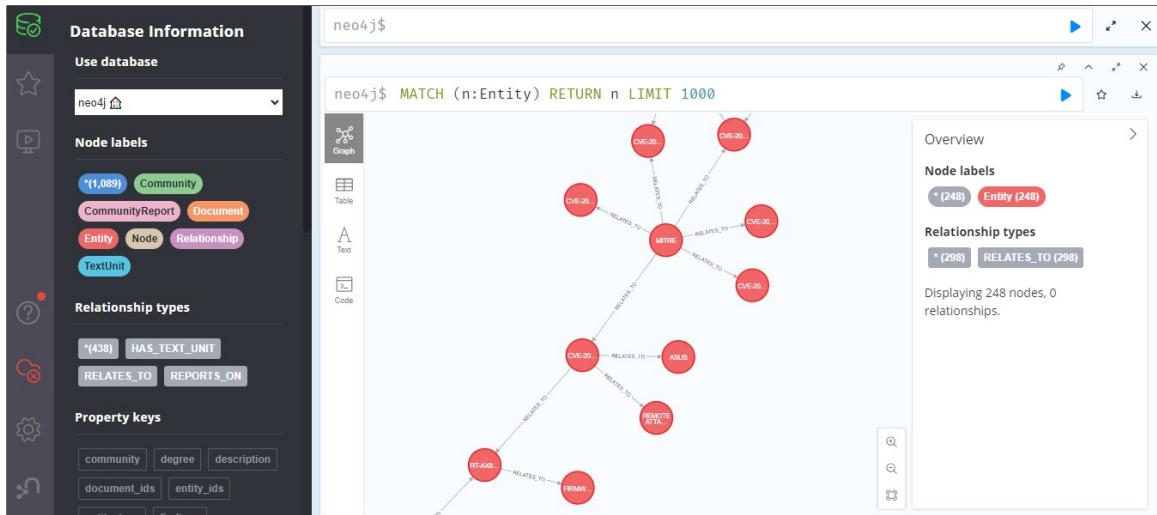


FIGURE 3.14 : visualisation en graph après l'indexation des cve(Zoom)

## Conclusion

L’implémentation du filtrage avancé et de la pipeline RAG a permis non seulement de trier et d’analyser les cve de manière plus précise, mais aussi de révéler des relations cachées entre les CVE grâce à l’indexation en graphes. Ces fonctionnalités offrent désormais une meilleure visibilité sur les menaces . Ce sprint a ainsi posé les bases d’une gestion des vulnérabilités plus proactive et efficace, essentielle pour la sécurité continue de l’infrastructure.

# RELEASE 2 : RECUPERATION DE LA CONFIGURATION UTILISATEUR ET DES SCANS AVANCÉS

---

## Plan

- 1   sprint 3 : Module Récupération de la configuration système et scans avancés utilisant NMAP et les outils de gestion de configuration . . . . . 43
- 2   Sprint 4 : Module gestion des paramètres de configuration utilisateur . . 51

## Introduction

Dans ce chapitre, nous allons présenter les différentes étapes de réalisation du troisième sprint "Récupération de la configuration système et scans avancés utilisant NMAP et les outils de gestion de configuration", et du quatrième sprint "gestion des paramètres de configuration utilisateur".

### 4.1 sprint 3 : Module Récupération de la configuration système et scans avancés utilisant NMAP et les outils de gestion de configuration

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Récupération de la configuration système et scans avancés utilisant NMAP et les outils de gestion de configuration".

#### 4.1.1 Objectifs du sprint 3

L'objectif principal de ce sprint est de permettre à l'application de récupérer la configuration du système en offrant trois méthodes distinctes. La première méthode consiste à utiliser un fichier YAML fourni par l'administrateur réseau ou système, décrivant les configurations spécifiques du système. La deuxième méthode repose sur un scan agressif réalisé automatiquement en arrière-plan à l'aide de NMAP, qui explore le réseau pour identifier les configurations et services actifs. Enfin, la troisième méthode intègre des outils de gestion de configuration comme Ansible et Puppet, permettant de récupérer automatiquement les configurations système via ces outils, offrant ainsi une flexibilité et une automatisation accrues dans la gestion des configurations.

#### 4.1.2 Backlog du sprint 3

Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant qu'administrateur système, je veux pouvoir charger un fichier YAML contenant les configurations système afin que l'application puisse les utiliser pour la gestion des vulnérabilités.	1	5

Id	Fonctionnalités	Priorité	Estimation (Jour)
2	En tant qu'administrateur système, je veux que l'application lance automatiquement des scans agressifs en arrière-plan en utilisant NMAP pour détecter et récupérer les configurations système	1	5
3	En tant qu'administrateur système, je veux pouvoir intégrer des outils de gestion de configuration comme Ansible pour récupérer automatiquement les configurations système à partir de cet outil.	1	5

TABLEAU 4.1 : Backlog du Sprint 3

#### 4.1.3 Spécification des besoins fonctionnels

La figure 4.1 représente le diagramme de cas d'utilisation du sprint " Recuperation de la configuration utilisateur et des scans avancés"

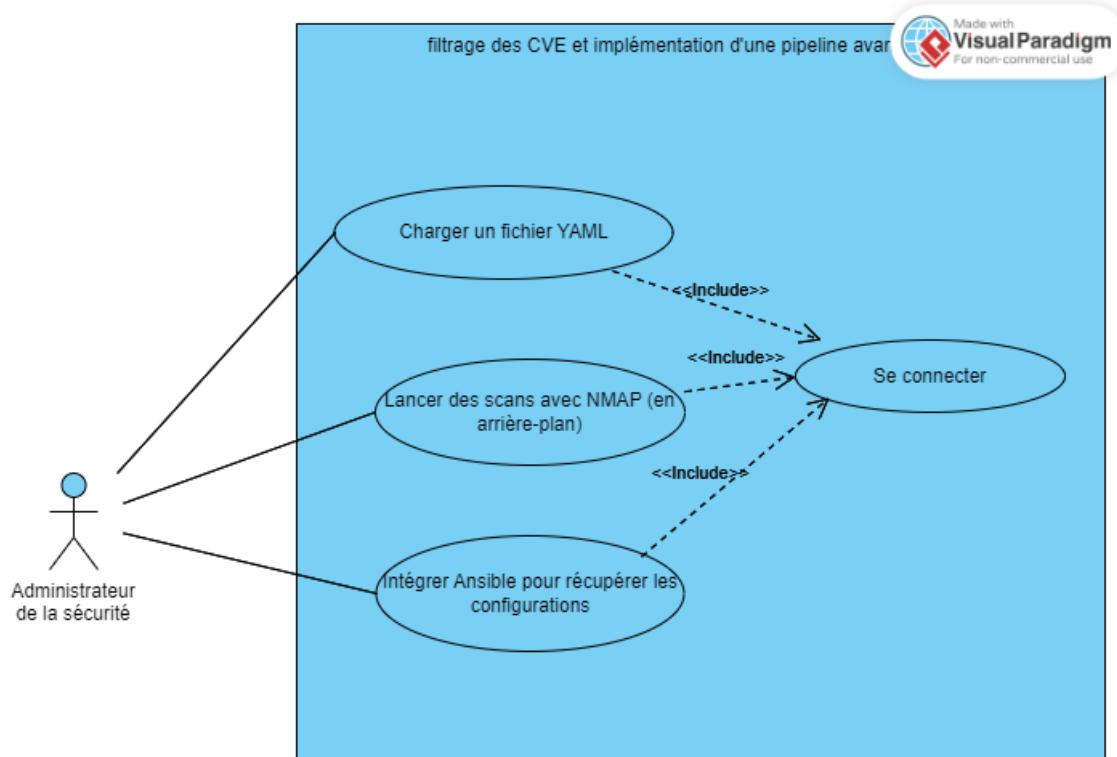
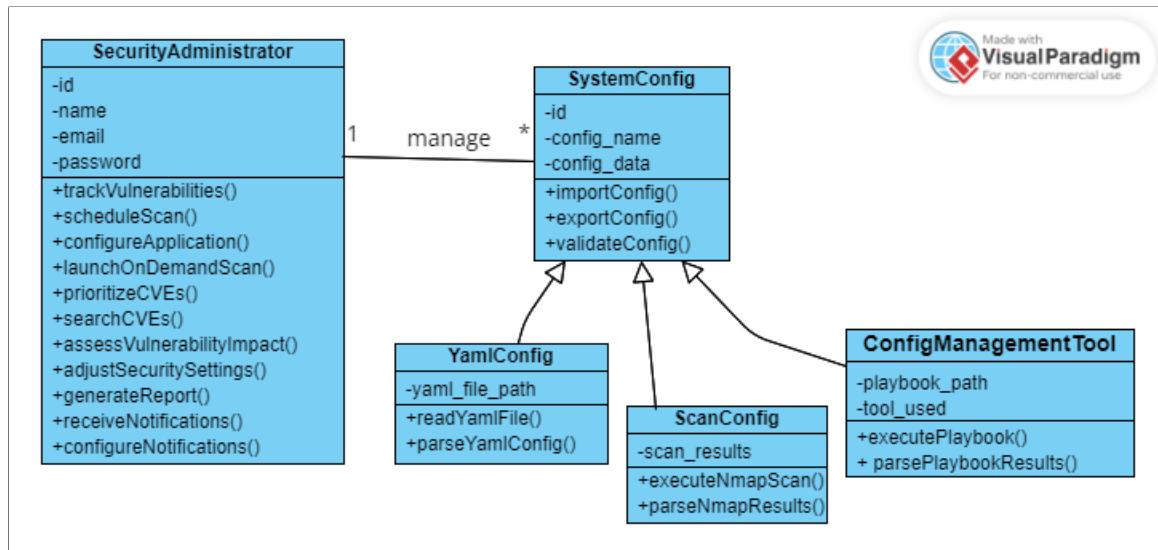


FIGURE 4.1 : Diagramme de cas d'utilisation " Recuperation de la configuration utilisateur et des scans avancés"

#### 4.1.4 Diagramme de classe

La figure 4.2 représente le diagramme de classe du sprint " Recuperation de la configuration utilisateur et des scans avancés".

- Le diagramme de classe présenté illustre la gestion des configurations système dans le cadre du Sprint 3. Au centre du diagramme se trouve la classe SystemConfig, qui représente la configuration globale du système, avec des méthodes pour importer, exporter, et valider les configurations. Cette classe est héritée par trois sous-classes : YamlConfig pour gérer les configurations via des fichiers YAML, ScanConfig pour les configurations récupérées par des scans NMAP, et ConfigManagementTool pour les configurations obtenues à l'aide d'outils de gestion comme Ansible .
- L'administrateur de sécurité, représenté par la classe SecurityAdministrator, a la responsabilité de gérer ces configurations en interagissant avec ces différentes classes, permettant ainsi une gestion flexible et automatisée des configurations système.



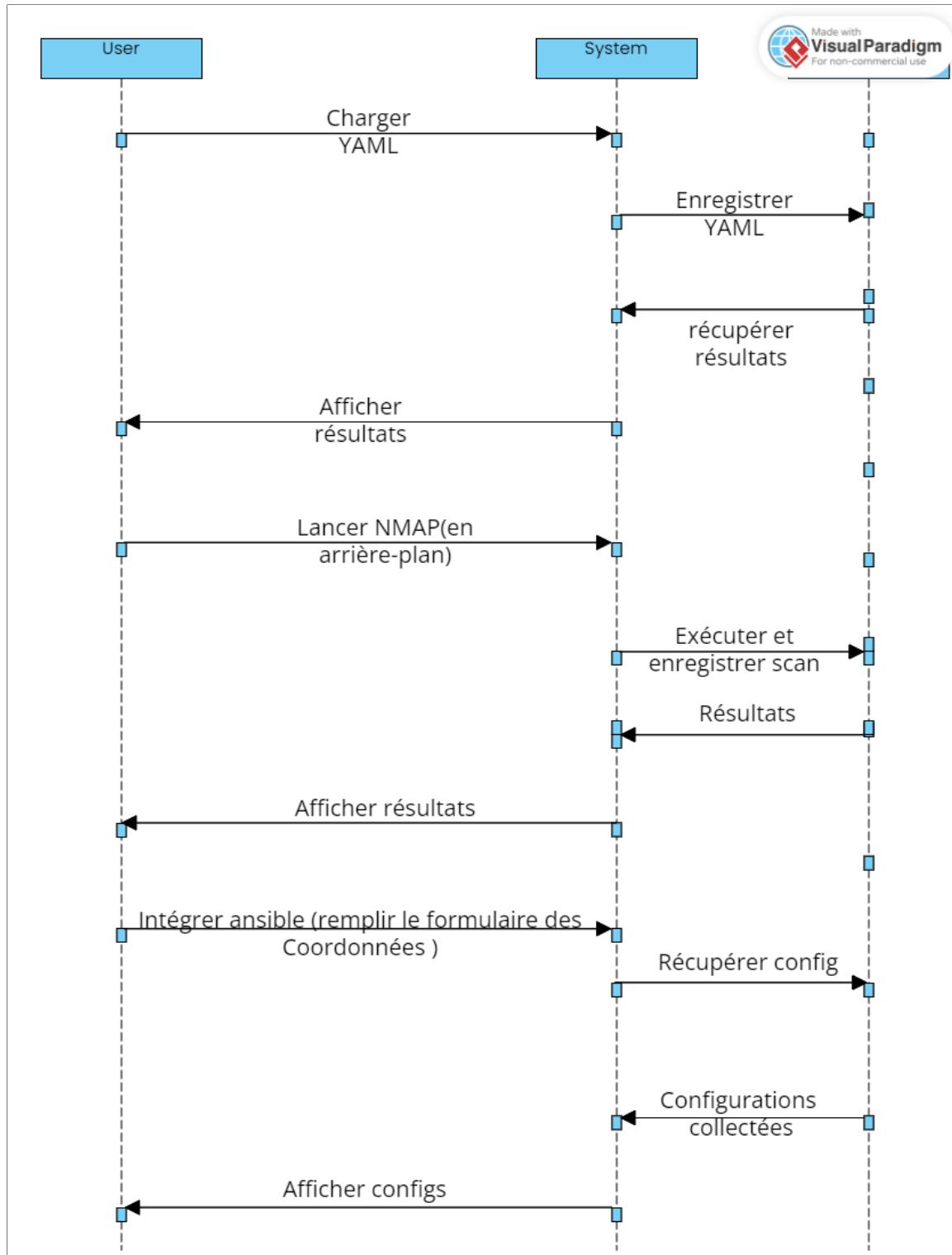
**FIGURE 4.2 :** Diagramme de classe " Recuperation de la configuration utilisateur et des scans avancé

#### 4.1.5 Diagrammes dynamiques

Dans cette partie nous allons présenter un diagramme séquence système pour la procédure de Recuperation de la configuration utilisateur et des scans avancé.

#### 4.1.5.1 Diagramme de séquence système " Recuperation de la configuration utilisateur et des scans avancés"

La figure 5.3 représente le diagramme de séquence système de la procédure de Recuperation de la configuration utilisateur et des scans avancés avec le système.

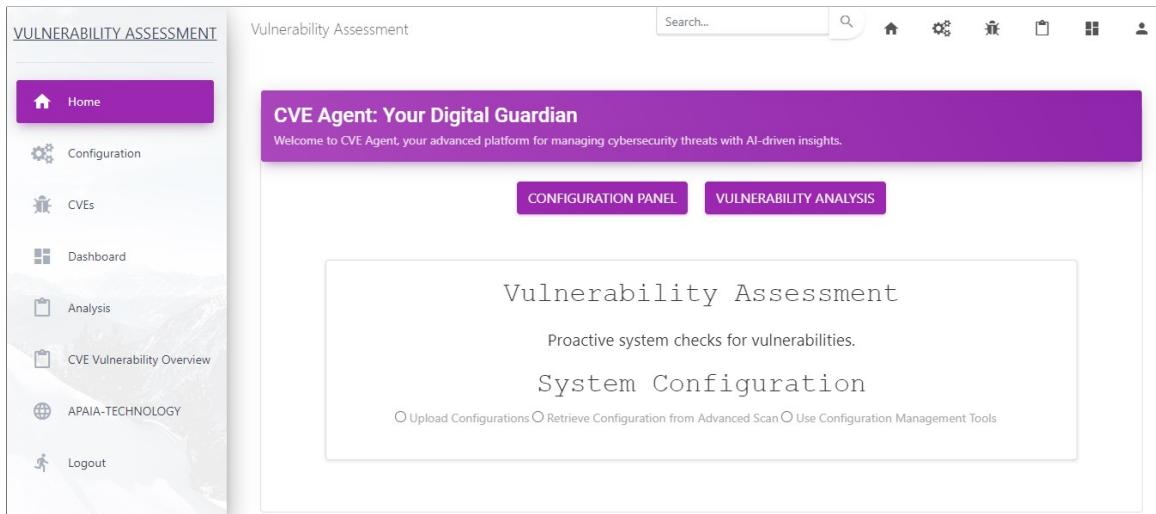


**FIGURE 4.3 :** Diagramme de séquence système " Recuperation de la configuration utilisateur et des scans avancés"

### 4.1.6 Réalisation

Dans ce sprint, la réalisation s'est concentrée sur l'intégration et l'automatisation de la récupération des configurations système. Trois approches distinctes ont été développées : l'importation de fichiers YAML, l'exécution de scans NMAP en arrière-plan, et l'utilisation d'outils de gestion de configuration tels qu'Ansible et Puppet. Le diagramme de classe montre comment la classe SystemConfig centralise ces différentes méthodes, avec YamlConfig gérant les configurations définies manuellement via des fichiers YAML, ScanConfig prenant en charge les résultats des scans NMAP, et ConfigManagementTool automatisant la récupération des configurations via des playbooks spécifiques. Chaque méthode de récupération a été implémentée pour offrir une flexibilité maximale, permettant aux administrateurs de choisir l'approche la mieux adaptée à leurs besoins spécifiques. L'ensemble de ces réalisations permet d'automatiser et de sécuriser efficacement la gestion des configurations, assurant ainsi une base solide pour l'évaluation continue des vulnérabilités.

La figure 4.4 représente l'interface Home. propose trois méthodes pour récupérer les configurations système : le chargement manuel de fichiers de configuration, l'exécution de scans avancés en arrière-plan, et l'utilisation d'outils de gestion de configuration comme Ansible et Puppet. L'utilisateur peut facilement naviguer entre ces options via une interface intuitive, garantissant une gestion proactive et automatisée des configurations de sécurité. Cette interface est conçue pour simplifier l'intégration et l'analyse des configurations système dans le cadre de la gestion des vulnérabilités..



**FIGURE 4.4 :** interface Home

Les figure 4.5 et 4.6 représente le premier choix de configuration yaml.

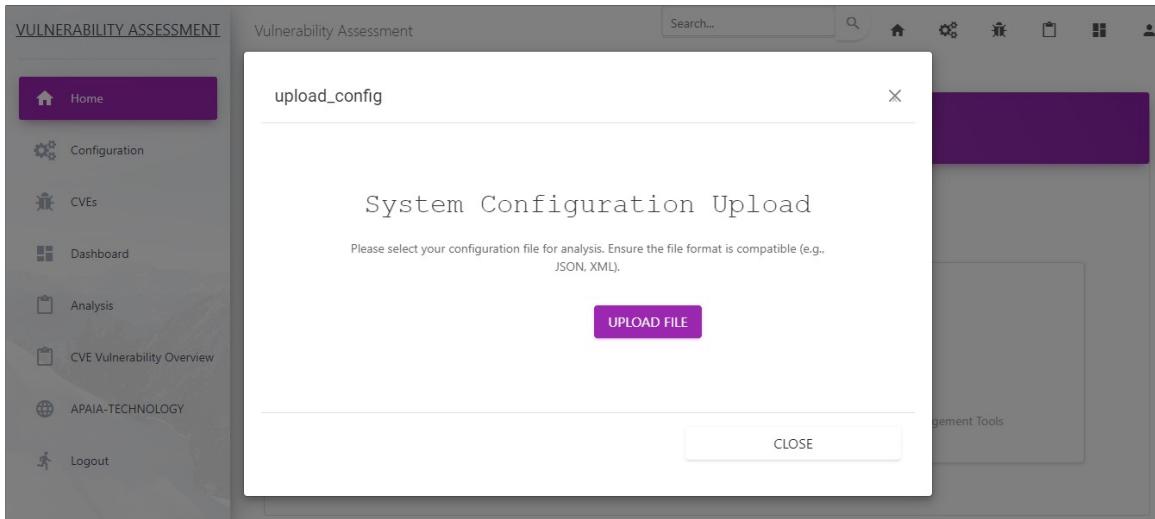


FIGURE 4.5 : interface de téléchargement de configuration

Configuration ID: 26					
uploaded Date: Aug. 9, 2024, 12:15 p.m.					
Device ID	Device Type	Operating System	Applications	Network Settings	Security Settings
node1	LINUX HOST CLIENT	Ubuntu 22.04.3 LTS (x86_64) 22.04.3	SSH vOpenSSH 8.2p1 (Ports: 22) DNS vbIND 9.16.22 (Ports: 53) HTTP vApache 2.4.41 (Ports: 80) vsftpd vvsftpd 2.3.4 (Ports: 21)	IP: 10.10.10.101 Subnet: 255.255.255.0 Gateway: 10.10.10.10	Firewall: Enabled Default Password Changed: No Encryption: Enabled
node2	Windows host client	Microsoft Windows 10 Pro 10.0.19041 N/A Build 19041	SSH vOpenSSH 8.2p1 (Ports: 22) DNS vbIND 9.16.22 (Ports: 53) Systlog vRsyslog 8.1912 (Ports: 514) 7-zip v7-zip 17.01 beta (Ports: ) SIEmail vSIEmail 5.5 (Ports: 110)	IP: 10.10.10.102 Subnet: 255.255.255.0 Gateway: 10.10.10.10	Firewall: Disabled Default Password Changed: Yes

FIGURE 4.6 : la configuration yaml afficher après le process système

Les figures 4.7 et 4.8 représentent la résultat du scan aggressif après son lancement en arrière plan .

**FIGURE 4.7 :** interface de resultat du scan aggressive

La figure 4.8

Port	Protocol	State	Name	Product	Version
21	tcp	open	ftp		
23	tcp	open	tcpwrapped		
53	tcp	open	domain		
80	tcp	open	http	RTK Web 0.9	
443	tcp	open	https		
5431	tcp	open	upnp	MiniUPnP	

**FIGURE 4.8 :** interface de resultat du scan aggressive développée

Maintenant, après avoir entré les informations de connexion nécessaires dans l'interface de "Configuration Management Tools" (comme illustré dans la première figure 4.9), nous avons accès aux configurations système à partir du playbook spécifié. En accédant à l'onglet suivant, représenté dans la deuxième figure, on peut voir une liste détaillée des dispositifs identifiés, avec leurs systèmes d'exploitation et adresses IP. Enfin, en sélectionnant un dispositif spécifique, la troisième figure nous montre les applications installées sur ce nœud, offrant ainsi une vue détaillée de l'environnement logiciel pour chaque machine.

**Configuration Management Tools**  
Connect SSH && Get System Configuration

Hostname: 10.10.239.113

Username: admin

Password: \*\*\*\*

Play Path: /home/ansible/Ansible/Test6.yml

**CONNECT**

**FIGURE 4.9 :** formulaire d’Outils de gestion de configuration

Après la récupération des configurations, cette interface 4.10 affiche une liste des dispositifs identifiés, incluant le type de dispositif, le système d’exploitation, et l’adresse IP de chaque noeud. Cela permet à l’administrateur de visualiser rapidement l’état de chaque machine dans l’infrastructure. Cette vue consolidée facilite la gestion des configurations et la prise de décision pour la sécurité.

Device : node1",		
Device Type:	OS:	IP Address:
VMware",	Ubuntu 22.04", Version: 22.04",	192.168.80.135"
Device : node3",		
Device Type:	OS:	IP Address:
VMware",	Ubuntu 22.04", Version: 22.04",	192.168.80.131"
Device : node2"		

**FIGURE 4.10 :** Liste des dispositifs

Cette vue 4.11 détaille les applications installées sur chaque noeud, incluant le nom de l’application et sa version. Ces informations sont cruciales pour identifier les logiciels vulnérables et prendre les mesures nécessaires pour leur mise à jour ou leur sécurisation. L’intégration de ces données avec les CVE récupérées permet une gestion proactive des vulnérabilités.

The screenshot shows a web interface for a 'VULNERABILITY ASSESSMENT' tool. On the left, a sidebar menu includes 'Home', 'Configuration', 'CVEs', 'Dashboard', 'Analysis', 'CVE Vulnerability Overview', 'APAIATECHNOLOGY', and 'Logout'. The main content area has a purple header 'Device : node1', showing 'Device Type: VMware', 'OS: Ubuntu 22.04, Version: 22.04', and 'IP Address: 192.168.80.135'. Below this is a section titled 'Applications:' with a table:

Application Name	Version
accountsservice	22.07.5-2ubuntu1.5
acl	2.3.1-1
acpi-support	0.144
acpid	1:2.0.33-1ubuntu1

**FIGURE 4.11 :** Détails des applications

## 4.2 Sprint 4 : Module gestion des paramètres de configuration utilisateur

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Gestion des paramètres de configuration utilisateur".

### 4.2.1 Objectifs du sprint 4

L'objectif principal de ce sprint est de permettre aux administrateurs système de configurer les paramètres de l'analyse des CVE, en définissant la profondeur et la fréquence des analyses, ainsi que de sélectionner et configurer les modèles LLM ou GPT utilisés pour ces analyses. Cela vise à offrir une flexibilité maximale dans la gestion des vulnérabilités, en permettant une personnalisation adaptée aux besoins spécifiques de chaque infrastructure.

### 4.2.2 Backlog du sprint 4

<b>Id</b>	<b>Fonctionnalités</b>	<b>Priorité</b>	<b>Estimation (Jour)</b>
1	En tant qu'administrateur système, je veux pouvoir choisir entre l'utilisation d'un modèle GPT ou un modèle LLM local pour l'analyse des CVE, afin d'ajuster l'approche en fonction des ressources disponibles	1	6

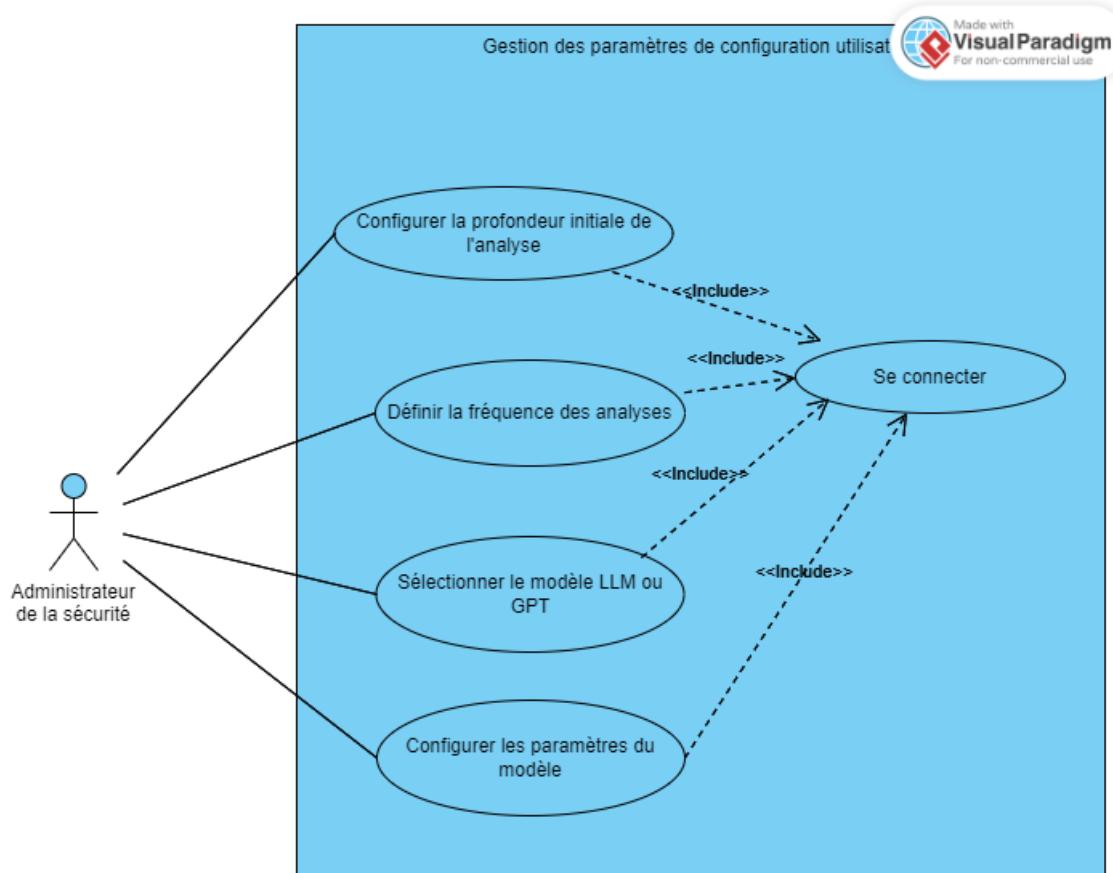
Id	Fonctionnalités	Priorité	Estimation (Jour)
2	En tant qu'administrateur système, je veux pouvoir configurer les paramètres spécifiques du modèle LLM, tels que le nom du modèle et l'URL du serveur, pour assurer une intégration fluide et efficace.	2	5
3	En tant qu'administrateur système, je veux pouvoir configurer la profondeur initiale de l'analyse des CVE en spécifiant une période de début et de fin, afin de concentrer les analyses sur des dates spécifiques.	2	5
4	En tant qu'administrateur système, je veux pouvoir définir la fréquence des analyses des CVE (quotidienne, hebdomadaire, etc.) et l'heure de leur exécution pour automatiser les tâches de manière optimale.	3	5

**TABLEAU 4.2 :** Backlog du Sprint 4

#### 4.2.3 Spécification des besoins fonctionnels

La figure 4.12 représente le diagramme cas d'utilisation du sprint "Gestion des paramètres de configuration utilisateur".

Ce diagramme montre les différents cas d'utilisation pour la gestion des paramètres de configuration utilisateur dans le cadre de l'analyse des CVE. L'administrateur de la sécurité peut configurer la profondeur initiale des analyses, définir la fréquence des analyses, sélectionner le modèle LLM ou GPT à utiliser, et configurer les paramètres spécifiques du modèle choisi. Chaque cas d'utilisation inclut une connexion préalable, soulignant l'importance de la sécurisation des accès avant la gestion des configurations.

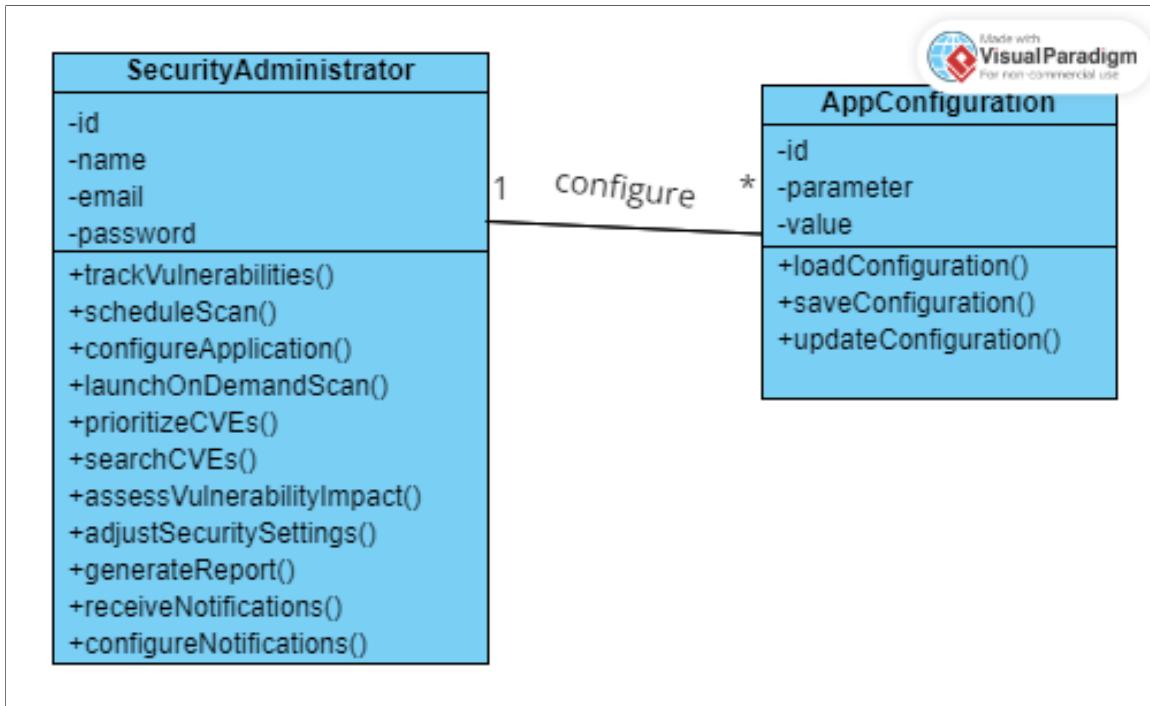


**FIGURE 4.12 :** Diagramme de cas d'utilisation "Gestion des paramètres de configuration utilisateur"

#### 4.2.4 Diagramme de classe

La figure 4.13 représente le diagramme de classe du sprint "Gestion des paramètres de configuration utilisateur".

- Ce diagramme met en évidence la relation entre la classe SecurityAdministrator et la classe AppConfiguration. L'administrateur de la sécurité configure les paramètres de l'application, qui sont stockés et gérés par la classe AppConfiguration. Les méthodes de cette classe permettent de charger, sauvegarder, et mettre à jour les configurations, assurant une gestion centralisée et flexible des paramètres de l'application.



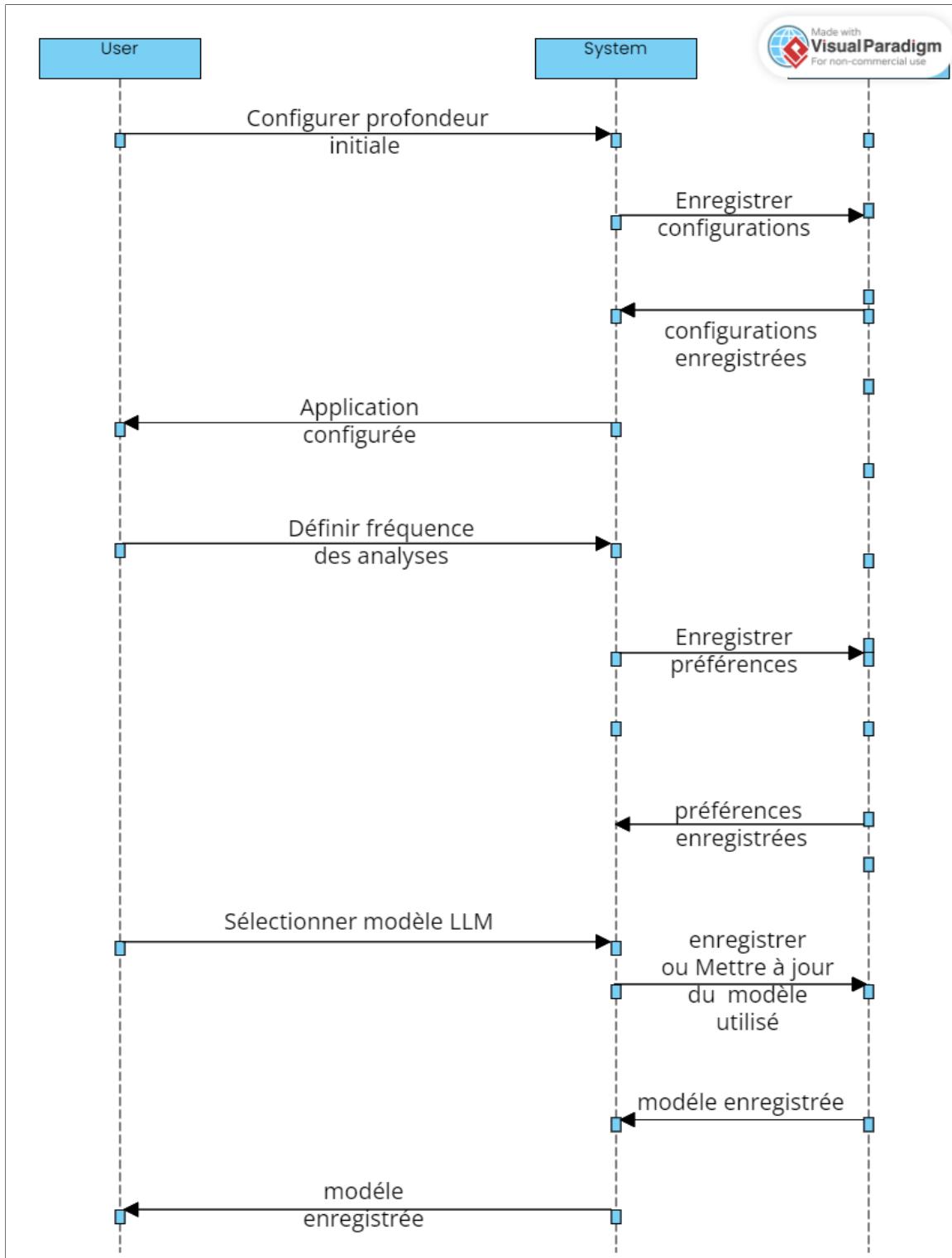
**FIGURE 4.13 :** Diagramme de classe "Gestion des paramètres de configuration utilisateur"

#### 4.2.5 Diagrammes dynamiques

##### 4.2.5.1 Diagramme de séquence objet "Gestion des paramètres de configuration utilisateur"

La figure 5.3 représente le diagramme de séquence objet du cas d'utilisation "marquer l'heure d'entrée".

Ce diagramme illustre le flux d'interactions entre l'administrateur et le système lors de la configuration des paramètres d'analyse des CVE. L'administrateur configure la profondeur initiale de l'analyse et la fréquence des scans, puis sélectionne et configure le modèle LLM à utiliser pour l'analyse. Chaque action entraîne une mise à jour et un enregistrement des configurations dans le système, garantissant que toutes les préférences de l'administrateur sont correctement enregistrées et appliquées.

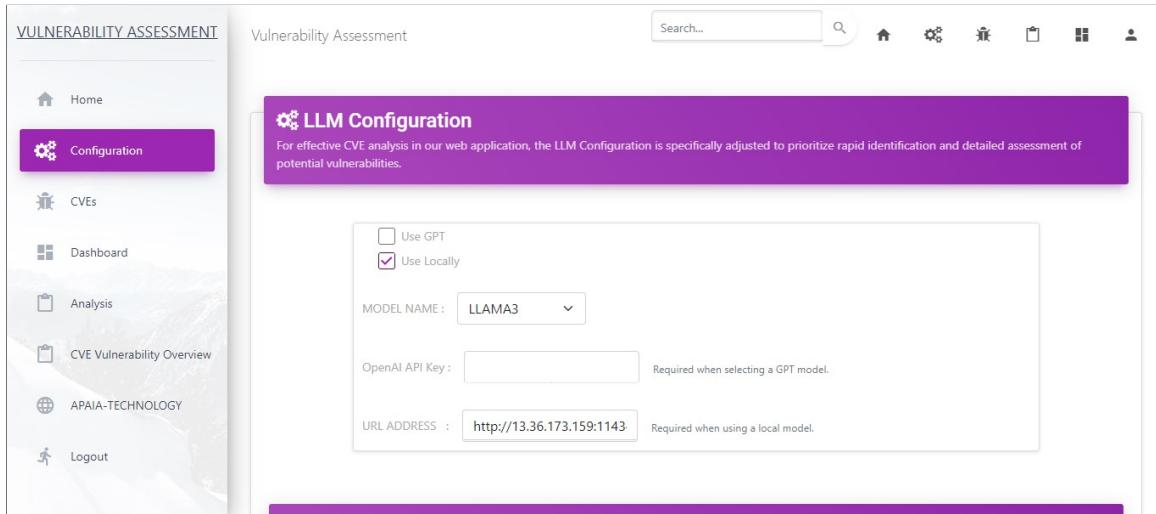


**FIGURE 4.14 :** Diagramme de séquence objet "Gestion des paramètres de configuration utilisateur"

#### 4.2.6 Réalisation

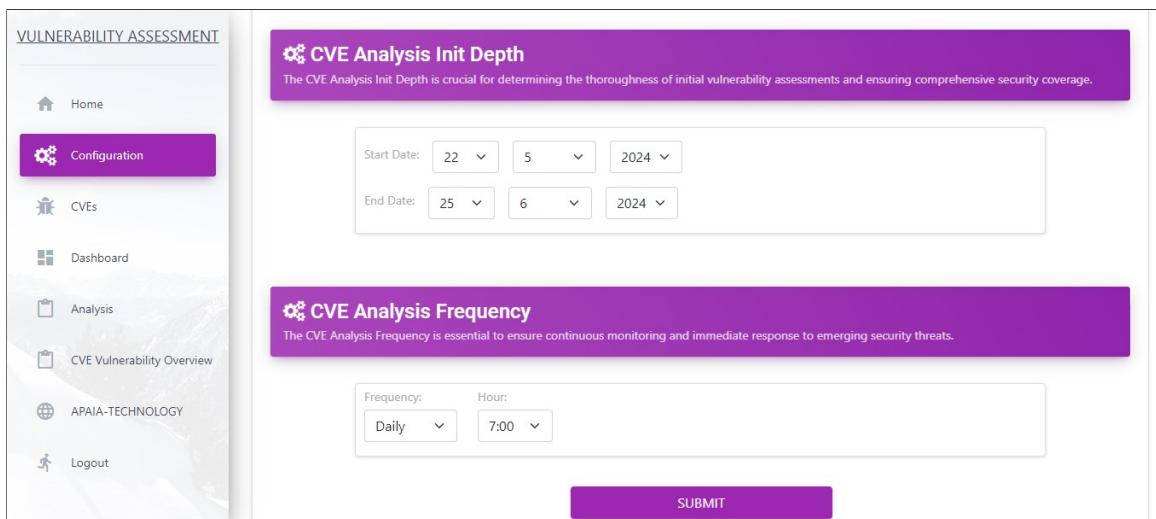
Pour la réalisation du quatrième sprint nous montrons l'interface de configuration du modèle LLM (Large Language Model), où l'administrateur peut choisir d'utiliser un modèle local ou un modèle GPT pour l'analyse des CVE. Les options incluent la sélection du modèle (comme LLAMA3), la saisie de la clé API pour les modèles GPT, et l'adresse URL pour l'accès au modèle local. Cette configuration

permet de personnaliser l'approche d'analyse des vulnérabilités en fonction des ressources disponibles et des préférences en matière de traitement des données. La figure 4.15 représente la page configuration utilisateur.



**FIGURE 4.15 :** interface de configuration LLM

La figure 4.16 nous montre l'interface de configuration utilisateur a été enrichie pour offrir une gestion plus granulaire des paramètres liés à l'analyse des CVE. La première capture d'écran présente les options permettant de définir la profondeur initiale de l'analyse des CVE, en sélectionnant une période spécifique grâce aux champs "Start Date" et "End Date", ainsi que la fréquence des analyses, configurée selon une base quotidienne ou à une heure précise. Cela permet aux administrateurs de planifier et d'automatiser les analyses de vulnérabilités selon les besoins spécifiques de leur infrastructure.



**FIGURE 4.16 :** interface de configuration Utilisateur

## Conclusion

Ce sprint final a permis de compléter les fonctionnalités essentielles de l'application en permettant une gestion détaillée des paramètres d'analyse des CVE. Grâce à la configuration de la profondeur d'analyse, de la fréquence des scans, et au choix du modèle LLM ou GPT, l'application offre désormais une personnalisation complète pour s'adapter aux besoins spécifiques des administrateurs de sécurité. Cette étape marque la fin d'une phase cruciale de développement, positionnant l'application comme un outil flexible et robuste pour la gestion proactive des vulnérabilités..

# RELEASE 3 : ANALYSE APPROFONDIE, TABLEAU DE BORD, RECOMMANDATIONS ET DASHBOARD

---

## Plan

1	Sprint 5 : Module d'Analyse approfondie et Tableau de bord . . . . .	59
2	sprint 6 : Recommandations et Dashboard . . . . .	65

## Introduction

Dans ce chapitre, nous allons présenter la réalisation du Sprint 5 "Analyse approfondie et Tableau de bord", et sprint 6 "Recommandations et dashboard". Nous allons commencer par l'identification des objectifs du Sprint, le backlog du sprint, la spécification des besoins et la réalisation.

### 5.1 Sprint 5 : Module d'Analyse approfondie et Tableau de bord

Dans cette section nous allons présenter les différents étapes de la réalisation du sprint "Analyse approfondie et Tableau de bord".

#### 5.1.1 Objectifs du sprint 5

L'objectif principal de ce sprint est de développer une fonctionnalité d'analyse approfondie des CVE, qui permet aux administrateurs de sécurité de réaliser des analyses détaillées en utilisant des modèles LLM avancés, comme Llama 3. Ces analyses doivent être présentées de manière claire et visuelle via un tableau de bord interactif, offrant une vue d'ensemble des vulnérabilités identifiées et de leur impact potentiel sur l'infrastructure. L'objectif est également d'obtenir des résultats précis et compréhensibles sur l'état de sécurité de leur système. Le module doit être capable de traiter en arrière-plan une vaste quantité de CVE (Common Vulnerabilities and Exposures), tout en permettant aux utilisateurs de configurer et de personnaliser les critères de filtrage et d'affichage des résultats..

#### 5.1.2 Backlog du sprint 5

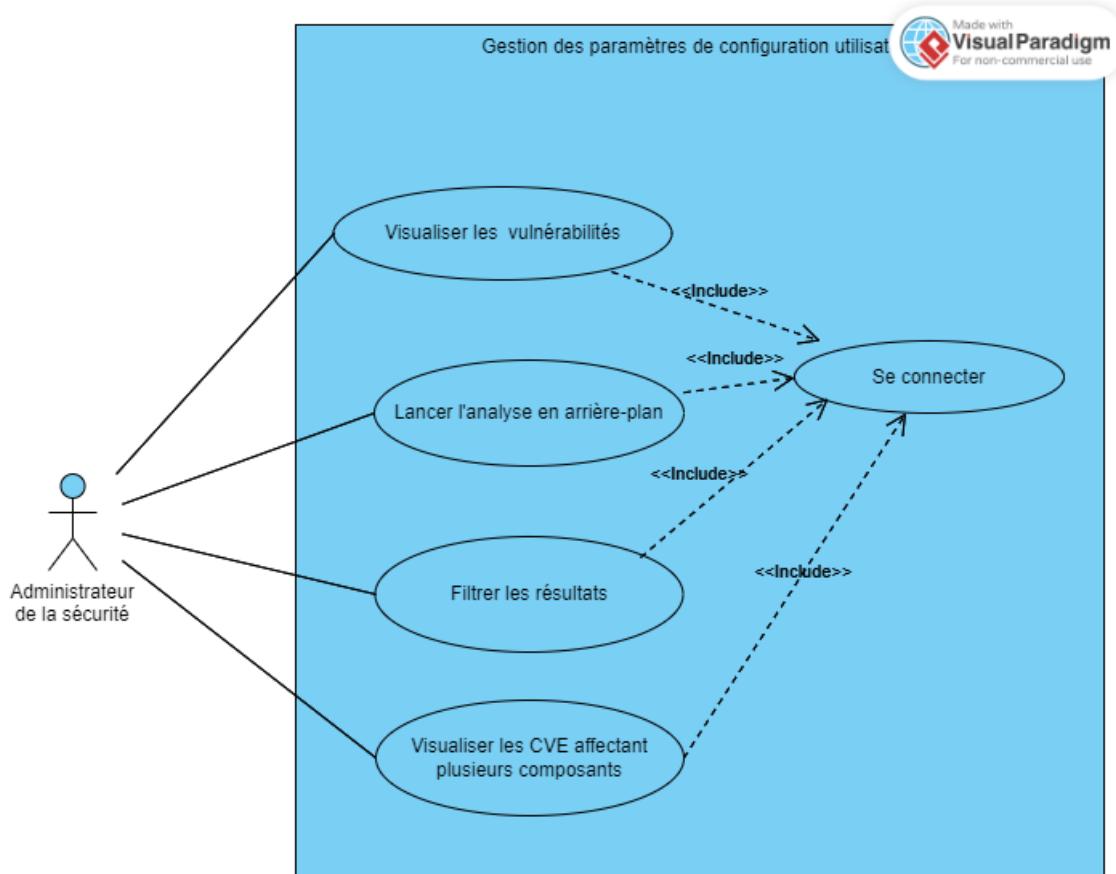
Id	Fonctionnalités	Priorité	Estimation (Jour)
1	En tant qu'administrateur système, je veux que les analyses de vulnérabilités soient exécutées automatiquement en arrière-plan pour garantir que les résultats sont toujours à jour sans affecter l'expérience utilisateur.	1	5
2	En tant qu'administrateur système, je veux pouvoir filtrer les résultats d'analyse par impact et par type de vulnérabilité afin de prioriser les menaces les plus critiques.	2	5

Id	Fonctionnalités	Priorité	Estimation (Jour)
3	En tant qu'administrateur système, je veux pouvoir visualiser les vulnérabilités affectant plusieurs composants dans un onglet dédié pour une compréhension claire de l'ampleur des menaces.	3	3
4	En tant qu'administrateur système, je veux recevoir une vue détaillée des résultats d'analyse, comprenant une description complète des vulnérabilités, des nœuds affectés, et des actions recommandées.	4	5

**TABLEAU 5.1 :** Backlog du Sprint 5

### 5.1.3 Spécification des besoins fonctionnels

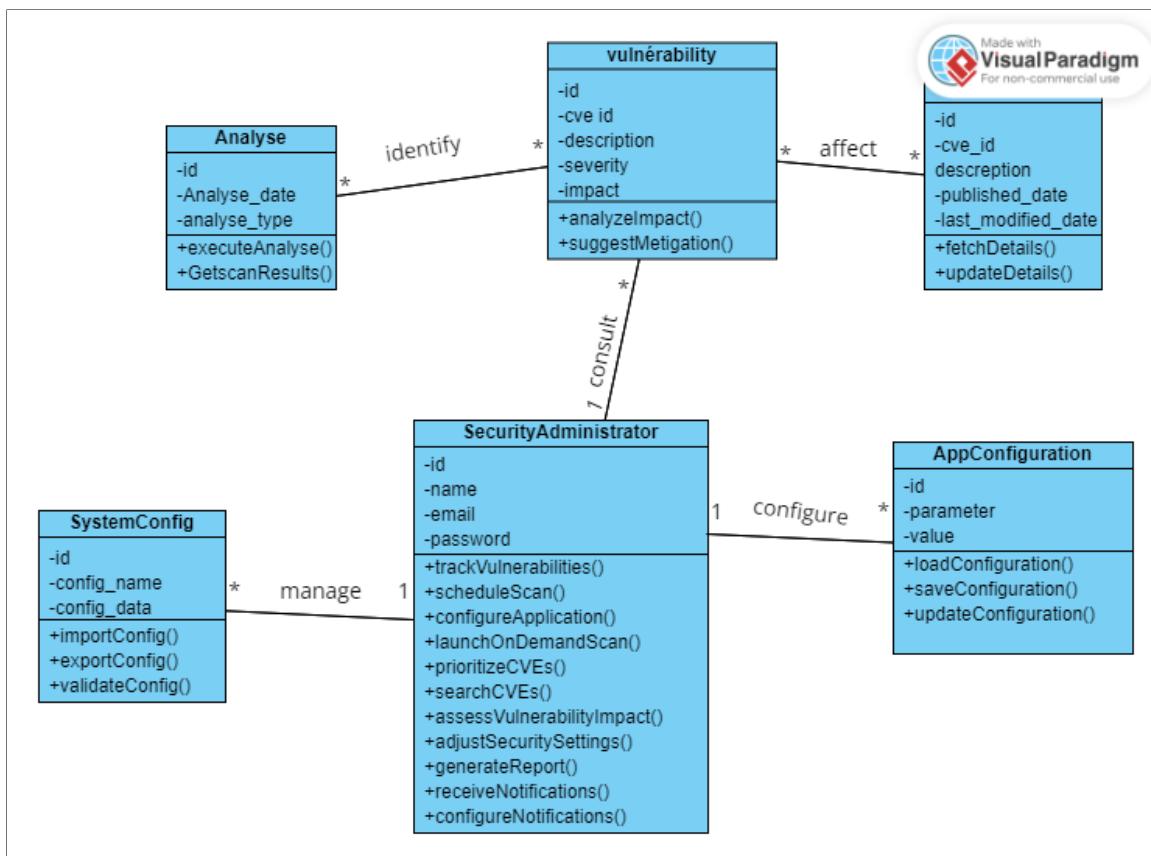
La figure 5.1 représente le diagramme de cas d'utilisation du sprint "Gestion d'achats et dépenses".

**FIGURE 5.1 :** Diagramme de cas d'utilisation "Analyse approfondie et Tableau de bord"

### 5.1.4 Diagramme de classe

Ce diagramme 5.2 illustre les différentes classes impliquées dans le processus d'analyse des CVE, ainsi que les relations entre elles, comme l'interface utilisateur, les tâches en arrière-plan, et le tableau de bord des résultats..

- Analyse : Une analyse est réalisée pour évaluer les vulnérabilités présentes dans la configuration du système. Chaque analyse est caractérisée par un identifiant, une date, et un type d'analyse. Une analyse peut identifier plusieurs vulnérabilités.
- Vulnérabilité : Une vulnérabilité représente un risque potentiel pour la sécurité du système. Elle est caractérisée par un identifiant, une description, un niveau de sévérité, un impact potentiel, et les dates de publication et de dernière modification. Une vulnérabilité peut affecter plusieurs analyses.
- SystemConfig : La configuration système regroupe les paramètres spécifiques d'un système à analyser. Elle est caractérisée par un identifiant et des données de configuration. Une configuration système peut être associée à plusieurs analyses.

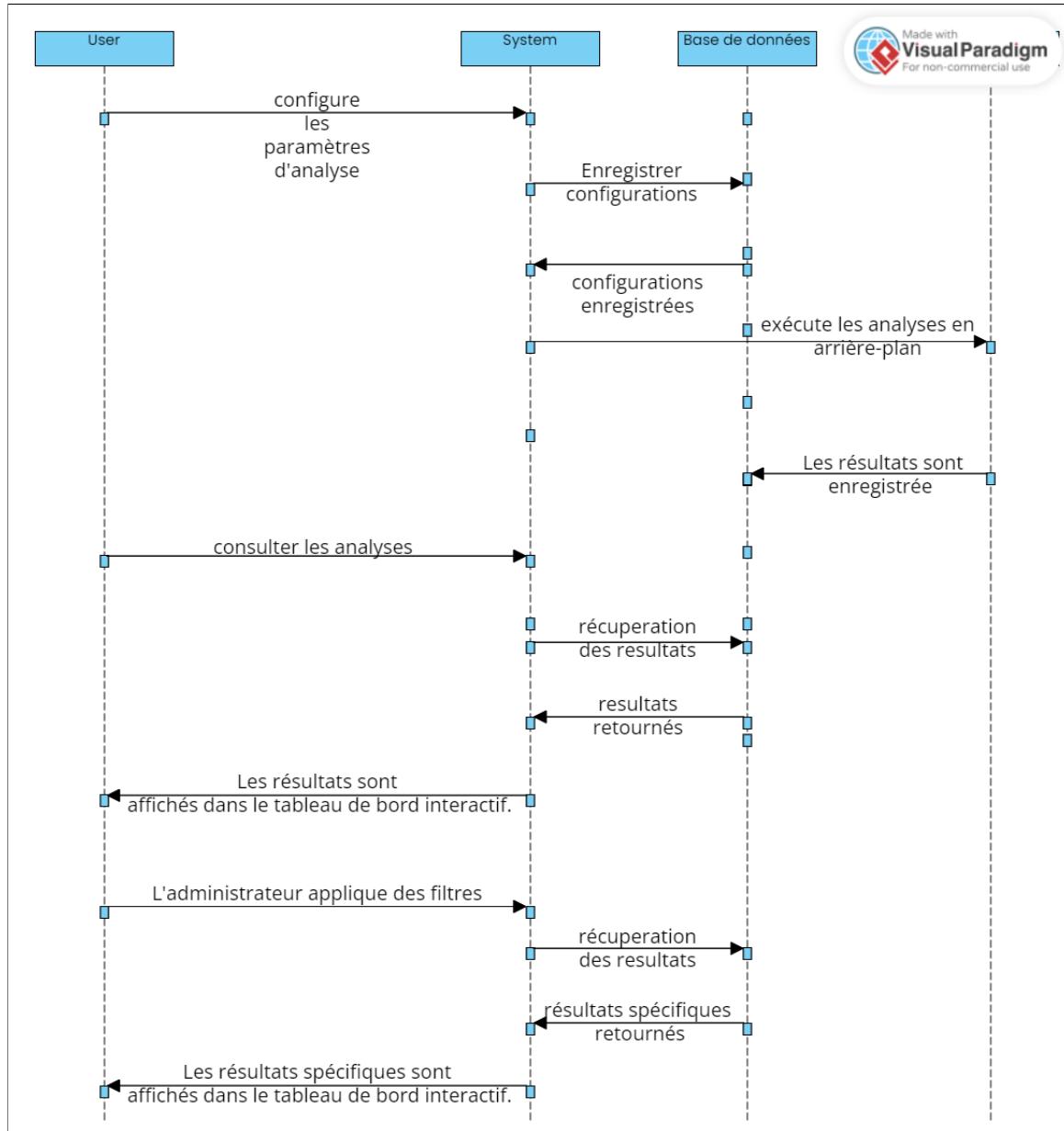


**FIGURE 5.2 :** Diagramme de classe "Analyse approfondie et Tableau de bord"

### 5.1.5 Diagrammes dynamiques

#### 5.1.5.1 Diagramme de séquence système "Analyse approfondie et Tableau de bord"

Ce diagramme 5.3 détaille les étapes suivies par l'utilisateur pour configurer et exécuter les analyses, suivies de l'affichage des résultats dans un tableau de bord interactif.



**FIGURE 5.3 :** Diagramme de séquence système "Analyse approfondie et Tableau de bord"

### 5.1.6 Réalisation

Pour la réalisation du sprint La figure 5.4 représente les résultats de l'analyse des vulnérabilités sous forme de tableau. Elle inclut le niveau d'impact, l'ID CVE, le type de vulnérabilité, le nom du noeud, et une description, ainsi que toutes les actions d'atténuation disponibles. Ce tableau de bord

permet à l'administrateur d'évaluer rapidement la posture de sécurité globale du système.

Impact Level	CVE ID	Vulnerability Type	Node	Description	Action
Low	<a href="#">CVE-2024-29785</a>	Local Information Disclosure	Nkhiel Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.100 Microsoft Office Professionnel Plus 2019 - fr-fr 16.0.17628.20144	This system running Microsoft Office Professionnel Plus 2019 might be vulnerable to local information disclosure due to uninitialized data. Although the exact component involved in the CVE is not specified, the potential compromise could allow an attacker to gather sensitive system or application-specific information.	No mitigation or recommendation available for this analysis.
High	<a href="#">CVE-2024-29784</a>	Local escalation of privilege	Nkhiel Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.100 Microsoft Visual C++ 2015-2019 Redistributable (x86) - 14.23.27820 14.23.27820.0	The Microsoft Visual C++ 2015-2019 Redistributable application in this Windows 10 Professional system is vulnerable to an out-of-bounds write vulnerability, which could lead to a local escalation of privilege attack. This highlights the need for prompt patching or mitigation measures to prevent attackers exploiting this vulnerability and compromising system integrity.	Patch Available: Yes Patch Release Link: <a href="https://www.microsoft.com/">https://www.microsoft.com/</a> Show More Last Update: 2024-06-11 UTC Recommendations: To secure your ... Show More

FIGURE 5.4 : Interface des Résultats de l'Analyse des Vulnérabilités

La figure 5.5 fournit un aperçu de la progression de la tâche d'analyse des vulnérabilités en cours d'exécution en arrière-plan. Elle indique combien de dispositifs ont été analysés et le nombre de CVE vérifiés par rapport aux configurations. Cet indicateur de progression en temps réel aide l'administrateur à surveiller l'analyse sans avoir besoin de vérifier manuellement l'état.

Critical	<a href="#">CVE-2024-29781</a>	Improper input validation	Visual C++ 2019 X64 Minimum Runtime - 14.23.27820 14.23.27820	attackers to disclose sensitive information without requiring additional execution privileges, posing a significant threat to the confidentiality of the system.	available for this analysis.
Critical	<a href="#">CVE-2024-29781</a>	Remote Information Disclosure	Nkhiel Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.100 Microsoft Edge 126.0.2592.56	The system's Microsoft Edge browser (version 126.0.2592.56) is directly affected by the CVE-2024-29781, allowing an attacker to remotely disclose sensitive information with no additional execution privileges needed.	No mitigation or recommendation available for this analysis.

Analysis background Progress

Completed  
Devices in your configurations: 4  
CVEs Checked: 1 of 3198

FIGURE 5.5 : Interface de Progression en Arrière-Plan de l'Analyse des Vulnérabilités

La figure 5.6 nous montre une fonctionnalité de recherche permettant de filtrer les résultats des analyses de vulnérabilités selon des critères spécifiques. Ici, l'utilisateur a recherché les composants affectés par une CVE critique concernant un nœud spécifique nommé "Gazalla". La recherche permet de réduire le nombre de résultats visibles pour se concentrer uniquement sur les éléments pertinents. Cette capacité de filtrage avancé aide les administrateurs à identifier rapidement et à gérer les vulnérabilités les plus critiques affectant plusieurs composants du système.

The screenshot shows the 'Vulnerability Assessment' interface. On the left, a sidebar menu includes 'Home', 'Configuration', 'CVEs', 'Dashboard', 'Analysis' (which is selected), 'CVE Vulnerability Overview', 'APAIA-TECHNOLOGY', and 'Logout'. The main area displays 'Vulnerability Analysis Results' with a table. The table has columns: Impact Level, CVE ID, Vulnerability Type, Node, and Description. One row is shown for 'Gazalla Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.16'. The 'Impact Level' is 'Critical' and the 'CVE ID' is 'CVE-2024-29786'. The 'Vulnerability Type' is 'Out of bounds write'. The 'Node' is 'Gazalla Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.16'. The 'Description' column contains a detailed technical note about an update for Windows 10 for x64-based Systems (KB5001716) version 8.94.0.0 being vulnerable to a critical out-of-bounds write issue allowing remote code execution. A 'Filters' panel on the right shows a search for 'Gazalla' and an 'Impact Level' filter set to 'All', with a 'SEARCH' button at the bottom.

**FIGURE 5.6 :** Interface de recherche

anssi que La figure 5.7 permet à l'utilisateur de filtrer les vulnérabilités par niveau d'impact, tel que "Critique", facilitant ainsi la priorisation des vulnérabilités nécessitant une attention immédiate. Cette capture montre les résultats du filtrage, affichant uniquement les vulnérabilités qui répondent aux critères spécifiés.

This screenshot shows the same 'Vulnerability Assessment' interface as Figure 5.6, but with a different filter applied. In the 'Filters' panel on the right, the 'Impact Level' dropdown is set to 'Critical'. This results in six items being listed in the main table. The first item is for 'Ennasr Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.191 NVIDIA Pilote graphique 462.59 462.59', and the second item is for 'Gazalla Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.16 Update for Windows 10 for x64-based Systems (KB5001716) 8.94.0.0'. Both entries show 'Impact Level: Critical' and 'CVE ID: CVE-2024-29786'.

**FIGURE 5.7 :** Interface des Résultats du Filtrage par Niveau d'Impact

Les deux prochaines captures vont nous montrer le Rapport d'Analyse Détaillée des CVE(présente une liste complète des CVE identifiées lors de l'analyse. Elle inclut des détails comme l'ID CVE, le type de vulnérabilité, son impact, et des détails techniques. La liste est interactive, permettant à l'utilisateur d'étendre chaque CVE pour voir des informations plus approfondies. Ce rapport est essentiel pour comprendre la gravité et les spécificités de chaque vulnérabilité.) et le Composant Principal Affecté(Cette capture d'écran montre les détails des composants du système affectés après l'analyse effectuée. Elle met en évidence des informations critiques telles que le nom du noeud, le système d'exploitation, l'adresse IP, le composant impacté, ainsi qu'une description de la vulnérabilité.

Cela permet à l'administrateur d'identifier les parties vulnérables du système et de prendre les mesures nécessaires).

Vulnerability ID	Description	Impact
CVE-2024-29781	Information Disclosure	Medium
CVE-2024-26657	Unknown (page fault due to direct access to a bad page)	Low
CVE-2024-29785	Local Information Disclosure	Low
CVE-2024-29784	Local escalation of privilege	High
CVE-2024-29786	Remote Code Execution	High
CVE-2024-5157	Use after free	High
CVE-2024-5158	Type Confusion in V8	High
CVE-2024-5159	Heap buffer overflow	High

**FIGURE 5.8 :** Interface de Rapport d'Analyse Détailée des CVE

Node Name	Operating System	IP Address	Impacted Component	Description
Gazalla	Microsoft Windows 10 Professionnel 10.0.19045	192.168.80.16	Google Chrome 125.0.6422.176	The vulnerability CVE-2024-5158 in Google Chrome prior to 125.0.6422.76 allows a remote attacker to potentially perform arbitrary read/write via a crafted HTML page. With the given system configuration running Google Chrome 125.0.6422.176, it is possible for an attacker to exploit this vulnerability and compromise the security of this application.
Nkhilet	Microsoft Windows 10 Professionnel 10.0.19045	192.168.80.100	Google Chrome 125.0.6422.176	An attacker could potentially exploit this vulnerability by crafting a malicious HTML page to perform arbitrary read/write operations. In the specified configuration, where Google Chrome is version 125.0.6422.176, an attacker could leverage this vulnerability to compromise the system, compromising user data and potentially allowing further attacks.

**FIGURE 5.9 :** Interface des Composants Principaux Affectés

## 5.2 sprint 6 : Recommandations et Dashboard

Le Sprint 6 se concentre sur la finalisation des fonctionnalités critiques de l'application de gestion des vulnérabilités, notamment la génération de recommandations automatisées pour chaque vulnérabilité identifiée et la mise en place d'un tableau de bord interactif. Ces outils visent à améliorer la prise de décision des administrateurs de sécurité en leur fournissant des informations pertinentes et en temps réel sur les menaces potentielles et les actions correctives nécessaires. Le tableau de bord permet également une visualisation globale des systèmes vulnérables, facilitant ainsi la gestion proactive de la sécurité au sein de l'infrastructure.

### 5.2.1 Objectifs du sprint 6

L'objectif de ce sprint est de développer un système robuste de recommandations pour chaque vulnérabilité identifiée, ainsi qu'un tableau de bord interactif permettant une visualisation claire et concise du système étudiée.

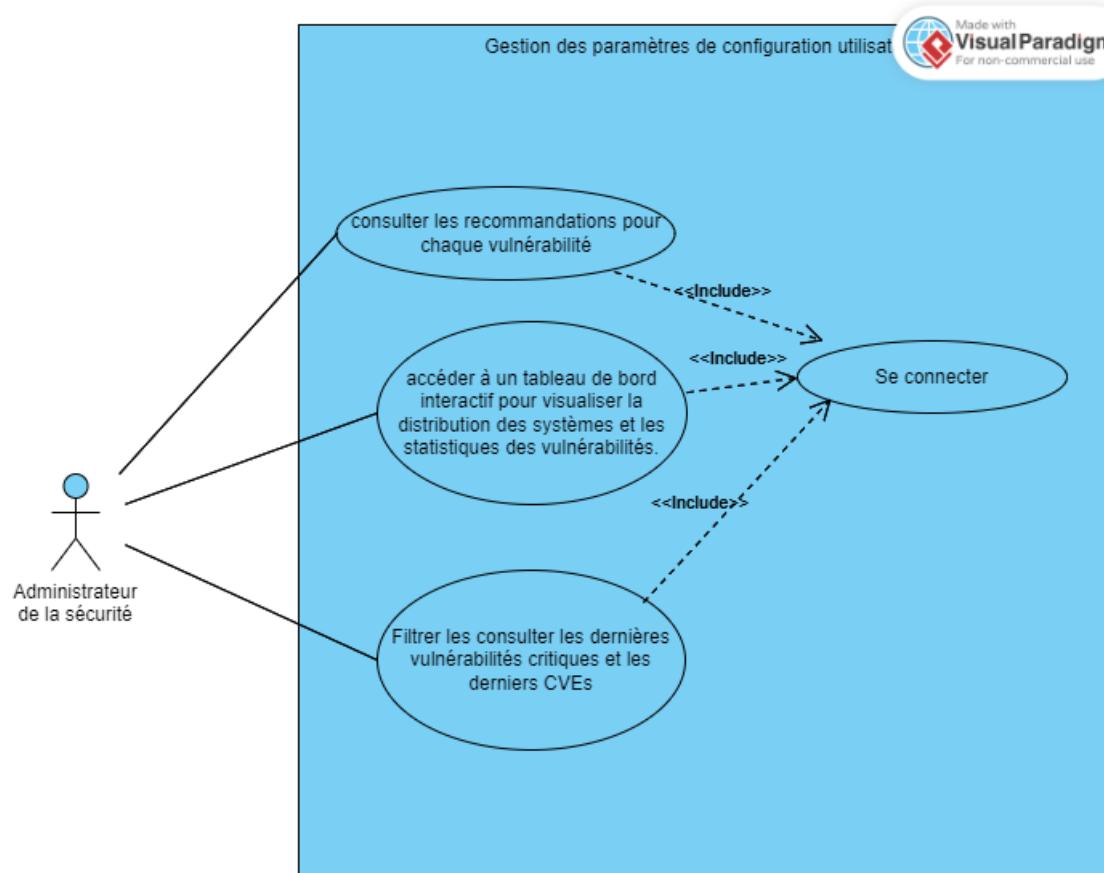
### 5.2.2 Backlog du sprint 6

<b>Id</b>	<b>Fonctionnalités</b>	<b>Priorité</b>	<b>Estimation (Jour)</b>
1	En tant qu'administrateur de sécurité, je veux recevoir des recommandations automatiques pour chaque vulnérabilité identifiée, afin de pouvoir appliquer rapidement des correctifs ou des mesures de mitigation.	1	5
2	En tant qu'administrateur de sécurité, je veux un tableau de bord interactif qui me montre la distribution des systèmes et les statistiques des vulnérabilités, pour une gestion proactive des menaces.	2	5
3	En tant qu'administrateur de sécurité, je veux pouvoir consulter rapidement les derniers CVEs et vulnérabilités identifiées, pour rester informé des nouvelles menaces.	1	5

**TABLEAU 5.2 :** Backlog du Sprint 6

### 5.2.3 Spécification des besoins fonctionnels

La figure 5.10 représente le diagramme de cas d'utilisation du sprint "Recommandations et Dashboard"



**FIGURE 5.10 :** Diagramme de cas d'utilisation "Recommandations et Dashboard"

#### 5.2.4 Recommandations pour les Vulnérabilités

- **Lien de Patch** : Lorsqu'un patch est disponible pour une vulnérabilité, le système génère automatiquement un lien vers le correctif correspondant. Ce lien est fourni en utilisant les références CVE associées, ce qui permet aux administrateurs de sécurité de corriger rapidement les failles.
- **Mitigations** : Dans le cas où aucun patch n'est disponible, des mesures de mitigation sont proposées. Celles-ci sont générées à partir d'une analyse approfondie par LLM (Large Language Model), qui suggère des pratiques pour minimiser les risques associés.
- **Intégration avec LLM pour Recommandations** : Le système exploite un LLM pour traiter les informations extraites des CVEs et fournir des recommandations personnalisées. Ce modèle est utilisé pour interpréter les descriptions techniques des vulnérabilités et pour générer des recommandations pertinentes basées sur les meilleures pratiques en cybersécurité.

Tableau de Bord Interactif :

- **Vue Globale sur la Distribution des Systèmes** : Le tableau de bord offre une vue d'ensemble sur la distribution des systèmes vulnérables au sein de l'infrastructure de l'utilisateur. Il permet

de filtrer et de visualiser les systèmes par type, version, et nombre de vulnérabilités détectées.

- **Statistiques des Vulnérabilités :** Le tableau de bord présente les statistiques des vulnérabilités, y compris les niveaux d'impact (critique, élevé, moyen, faible), offrant ainsi une compréhension immédiate des risques. De plus, les cinq dernières vulnérabilités détectées sont mises en avant pour une action rapide.
- **Derniers CVEs et Vulnérabilités :** Les cinq dernières vulnérabilités critiques sont listées, avec des détails sur les systèmes concernés, leur impact, et les recommandations disponibles.
- De plus, les cinq derniers CVEs sont affichés avec une option de visualisation détaillée, permettant aux utilisateurs de se tenir informés des nouvelles menaces

### 5.2.5 Réalisation

Pour la partie réalisation du sprint "Recommandations et Dashboard" nous allons représenter sur La figure 5.11 comment les vulnérabilités critiques sont triées et les recommandations générées, avec des détails sur les actions possibles (application de patch, mitigation ou aucune action requise).

VULNERABILITY ASSESSMENT		29781	Disclosure	16 Click-to-Run Extensibility Component 16.0.17628.20110	vulnerability to steal sensitive information from the system, posing a significant risk to its security and integrity.	analysis.
Home Configuration CVEs Dashboard <b>Analysis</b> CVE Vulnerability Overview APAIA-TECHNOLOGY Logout	High Critical High	CVE-2024-29781 CVE-2024-29781 CVE-2024-29781	Out of bounds read due to improper input validation	Soukra Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.198 Microsoft Edge 125.0.2535.92	The system is running Microsoft Edge 125.0.2535.92, which is vulnerable to a high-level information disclosure attack. This could allow attackers to remotely access sensitive system files or data without needing additional privileges or user interaction.	<b>Patch Available:</b> Yes <b>Patch Release Link:</b> <a href="https://developer.android.com/security/bulletin/2024-06">https://developer.android.com/security/bulletin/2024-06</a> <b>Last Update:</b> June 11, 2024 <b>Recommendations:</b> Please refer to the Android Security Blog for detailed recommendations on best practices for securing your systems.
				Soukra Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.198 Google Chrome 125.0.6422.176	The system is running Google Chrome version 125.0.6422.176, which contains a critical vulnerability in ss_OssAsnManagement.c, allowing attackers to perform remote information disclosure with no additional execution privileges needed. This could potentially expose sensitive user data or allow malicious code injection.	<b>Mitigation Measures:</b> For devices not yet updated, consider implementing network segregation, disabling unnecessary services and restricting access to vulnerable interfaces.
				Ennas Microsoft Windows 10 Professionnel 10.0.19045 192.168.80.191	The system's Microsoft Edge browser (126.0.2592.56) is vulnerable to a high-level out-of-bounds read vulnerability. This could allow attackers to remotely disclose information without requiring	<b>No mitigation or recommendation available for this analysis.</b>

FIGURE 5.11 : Recommandations pour les Vulnérabilités

Les captures suivantes illustrent le tableau de bord avec des graphiques montrant la répartition des systèmes affectés, ainsi que les types de vulnérabilités les plus fréquentes. Ces informations sont essentielles pour la gestion proactive des menaces.

La figure 5.12 représente le dashboard .

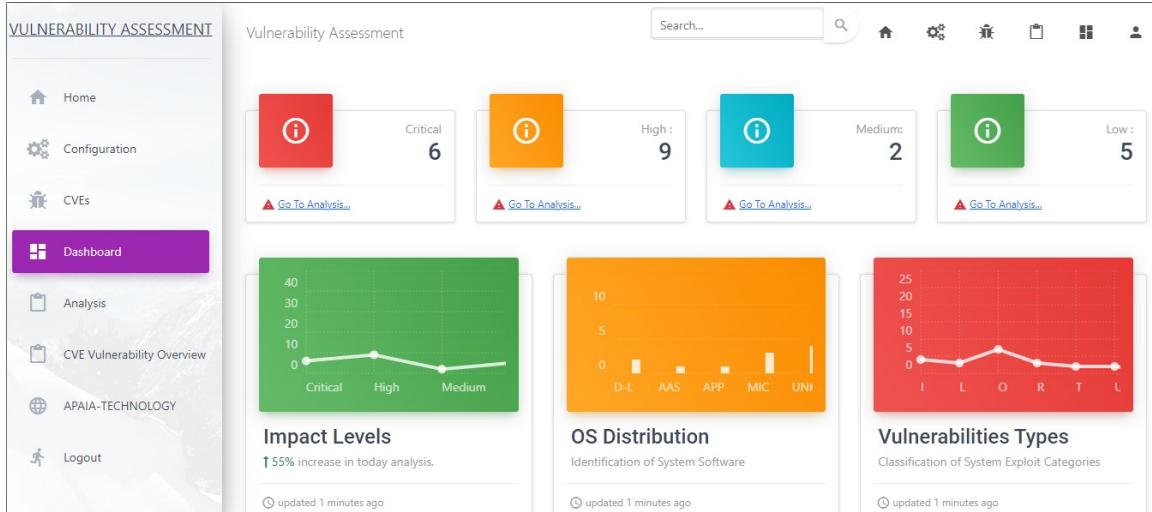


FIGURE 5.12 : Dashboard Employé

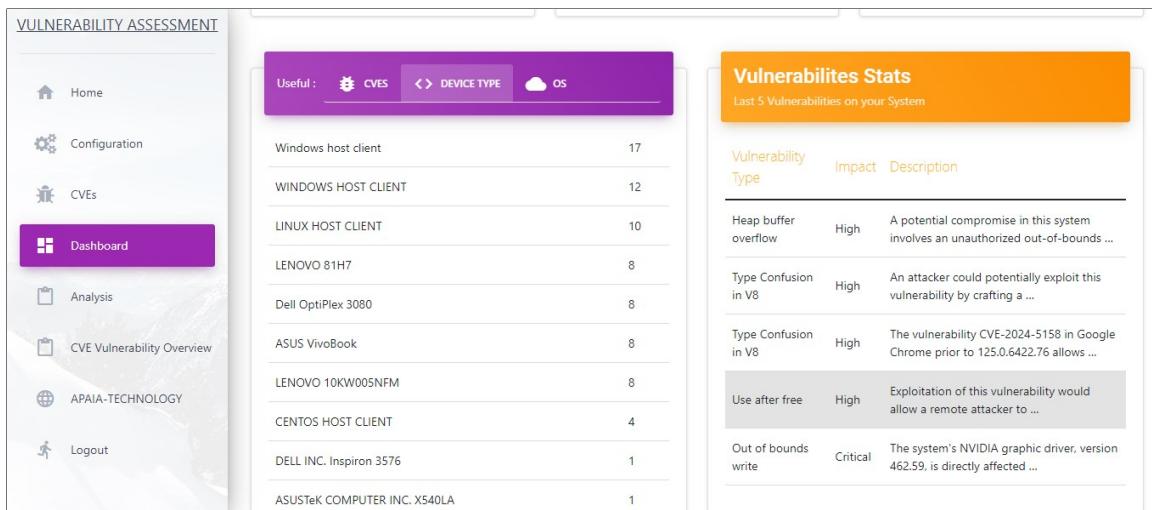


FIGURE 5.13 : Type des équipements

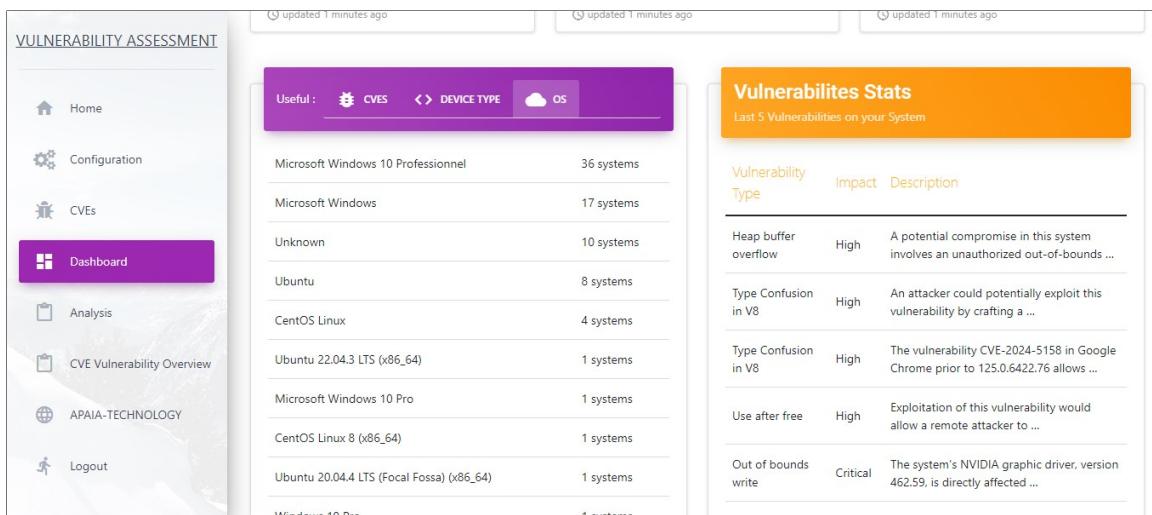


FIGURE 5.14 : Distribution des systèmes d'exploitation

The screenshot displays a web-based vulnerability assessment interface. On the left, a sidebar titled 'VULNERABILITY ASSESSMENT' includes links for Home, Configuration, CVEs, Dashboard (which is highlighted in purple), Analysis, CVE Vulnerability Overview, APAIA-TECHNOLOGY, and Logout. The main content area has a purple header bar with 'Useful:' followed by filters for 'CVEs', 'DEVICE TYPE', and 'OS'. Below this, a section titled 'Vulnerabilities Stats' shows 'Last 5 Vulnerabilities on your System'. A table lists five vulnerabilities with columns for Type, Impact, and Description.

Type	Impact	Description
Heap buffer overflow	High	A potential compromise in this system involves an unauthorized out-of-bounds ...
Type Confusion in V8	High	An attacker could potentially exploit this vulnerability by crafting a ...
Type Confusion in V8	High	The vulnerability CVE-2024-5158 in Google Chrome prior to 125.0.6422.76 allows ...
Use after free	High	Exploitation of this vulnerability would allow a remote attacker to ...
Out of bounds write	Critical	The system's NVIDIA graphic driver, version 462.59, is directly affected ...

**FIGURE 5.15 :** Derniers CVEs et Vulnérabilités

## Conclusion

Le sprint 6 se termine avec la mise en place d'un système de recommandations sophistiqué et d'un tableau de bord interactif. Grâce à l'intégration du LLM pour l'analyse des vulnérabilités et la génération de recommandations, les utilisateurs sont désormais mieux équipés pour gérer les menaces de sécurité. Le tableau de bord fournit une vue d'ensemble des systèmes vulnérables et des statistiques essentielles pour une prise de décision rapide et informée.

# Conclusion générale

Le projet de développement de l'application d'évaluation des vulnérabilités, structuré en plusieurs sprints, a permis de concevoir un système robuste et efficace pour la gestion proactive des menaces de sécurité. À travers ces sprints, nous avons progressivement élaboré et amélioré les différentes composantes de l'application, allant de la configuration initiale des paramètres de sécurité à la génération de rapports détaillés et la présentation interactive des résultats via un tableau de bord.

Chaque sprint a apporté une valeur ajoutée essentielle à l'application, en mettant l'accent sur la configuration personnalisée, l'intégration des modèles de langage pour l'analyse des vulnérabilités, et la génération automatique de recommandations basées sur les CVEs. Le tableau de bord interactif, quant à lui, offre une visualisation claire et intuitive de l'état de sécurité de l'infrastructure, facilitant ainsi la prise de décision rapide pour les administrateurs.

L'intégration d'un LLM pour l'analyse et la génération de recommandations a particulièrement renforcé l'efficacité de l'application, en permettant de traiter des descriptions techniques complexes et de proposer des mesures pertinentes pour atténuer les risques. La flexibilité du système, qui permet d'adapter les recommandations en fonction de la disponibilité des patches ou des stratégies de mitigation, garantit une réponse appropriée à chaque type de vulnérabilité.

En somme, le projet se termine avec la mise en place d'un outil complet qui, non seulement répond aux besoins actuels de gestion des vulnérabilités, mais est également prêt à évoluer avec les nouvelles menaces et technologies. Ce système offre aux utilisateurs un moyen sophistiqué et intuitif de surveiller, analyser et agir sur les vulnérabilités de leur infrastructure, renforçant ainsi leur posture de sécurité de manière significative.

# Bibliographie

[B1] Pierre Pezziardi, Référentiel des Pratiques Agiles, édition ebook.2013

# Résumé

Le présent rapport synthétise le travail effectué dans le cadre du projet de fin d'études pour l'obtention du diplôme national d'ingénieur en informatique au sein de l'entreprise APAIA-TECHNOLOGY. L'objectif de ce travail est la conception et l'implémentation d'une application web d'évaluation des CVE. Ce projet vise à développer une application robuste pour l'évaluation automatisée des CVE, en utilisant Django et des technologies avancées telles que Llama 3 et une pipeline RAG pour une analyse approfondie..

**Mots clés :** Évaluation des CVE, Django, LLM, Application web, Graph RAG

# Abstract

This report summarizes the work carried out as part of the final year project for obtaining the national engineering degree in computer science at APAIA-TECHNOLOGY.

The objective of this work is the design and implementation of a web application for CVE assessment. This project aims to develop a robust application for the automated evaluation of CVEs, using Django and advanced technologies such as Llama 3 and a RAG pipeline for in-depth analysis.

**Keywords :** CVE Assessment, Django, LLM, Web Application, Graph RAG