ECOLE MAROCAINE DES SCIENCES DE L'INGENIEUR
Membre de HONORIS UNITED UNIVERSITIES

UniWare IT Specialist

Rapport de stage

4eme année
Ingénierie Informatique et Réseaux

Sous le thème

Etude et developpement d'un Système de vérification des composants du système d'information sont à jour en terme de patch de sécurité

Période de stage
du 10 juillet 2025 au 01 septembre 2025

Réaliser par :

AIT EL QADI Anas

IZMAR Mohamed Taha

Encadré par :

Mme. ABBAOUI Wafae (EMSI)

M. MOUSTAKIM Said (Uniware)

Entreprise : Uniware S.A.R.L
Activité : IT
Adresse : Avenue Al Massira, N 1183 Amal 5

Année universitaire 2024/2025

Dédicace

À tous ceux qui ont contribué à la concrétisation de ce projet,nous avons ensemble traversé des épreuves et trouvé des inspirations, partagé des rires et des moments de réflexion.

Ce projet est bien plus qu'une simple réalisation ; il est le reflet de notre dévouement, de notre créativité et de notre esprit de collaboration. Chaque membre de notre groupe a apporté une touche unique, illuminant ce projet de manière spéciale et précieuse.

Que cette dédicace témoigne de notre travail acharné, de notre solidarité et de notre passion commune.Puissions-nous nous souvenir de ce projet comme d'une preuve indélébile de ce que nous sommes capables d'accomplir ensemble.

Remerciements

Nous souhaitons exprimer notre profonde gratitude à tous ceux qui ont contribué à la réalisation de ce projet de fin d'études. Tout d'abord, nous remercions sincèrement Monsieur le Directeur de l'EMSI pour la formation de qualité qu'il nous a offerte. Grâce à son leadership et à sa vision, nous avons bénéficié d'un environnement académique exceptionnel, propice à notre épanouissement et à notre réussite.Nos sincères remerciements vont également à nos professeurs, dont l'expertise, la patience et l'engagement ont été essentiels à notre apprentissage. Leurs enseignements et leurs conseils nous ont guidés tout au long de notre parcours académique, nous permettant de développer les compétences nécessaires pour relever les défis de demain.À nos amis, merci pour votre soutien constant, vos encouragements et pour les moments de camaraderie qui ont rendu cette expérience encore plus mémorable. Votre présence à nos côtés a été une source de motivation et de réconfort.Enfin, nous souhaitons exprimer notre infinie reconnaissance à nos parents. Votre amour, votre soutien inconditionnel et vos sacrifices ont été le pilier sur lequel nous avons construit nos rêves. Sans vous, rien de tout cela n'aurait été possible.Ce projet est le fruit de la collaboration, du travail acharné et du soutien de chacun d'entre vous. Merci de nous avoir permis de réaliser cet accomplissement.

Résumé

Le Security Component Checker est une application web développée en Java (Spring Boot) avec une interface construite en Spring MVC et Thymeleaf, et reposant sur une base de données MySQL. Son rôle principal est de faciliter la détection et la gestion des vulnérabilités grâce à l'intégration d'API spécialisées telles que la NVD (National Vulnerability Database). L'application permet d'authentifier les utilisateurs via un système sécurisé, de gérer différents rôles comme administrateur ou utilisateur classique, et de conserver un historique des actions grâce à des logs et à une traçabilité renforcée.

Sur le plan fonctionnel, l'outil propose l'ajout et la gestion de composants matériels et logiciels, leur analyse automatique ou manuelle, et la classification de leur statut de sécurité (à jour, vulnérable ou nécessitant une mise à jour). Les vulnérabilités détectées sont enregistrées et mises en valeur dans un tableau de bord interactif, qui fournit également des statistiques et des graphiques pour une meilleure visualisation de l'état global du système. Le projet intègre aussi une dimension de reporting, en permettant la génération de rapports sous format PDF ou Excel, avec la possibilité d'envoyer des notifications par email en cas de détection de vulnérabilités critiques.

Enfin, le projet a été conduit selon une méthodologie agile, avec des sprints hebdomadaires incluant la conception, la mise en place de l'authentification, la gestion des composants, l'intégration des vulnérabilités, puis la génération de rapports et le tableau de bord. Les contraintes de développement, liées à un calendrier serré et à la nécessité d'utiliser des outils open-source, ont guidé les choix techniques et méthodologiques. L'ensemble constitue une solution robuste et évolutive pour renforcer la sécurité des systèmes d'information en entreprise.

Abstract

The Security Component Checker is a web application developed in Java (Spring Boot) with an interface built using Spring MVC and Thymeleaf, and relying on a MySQL database. Its main role is to facilitate the detection and management of vulnerabilities through the integration of specialized APIs such as the NVD (National Vulnerability Database). The application allows users to authenticate through a secure system, manage different roles such as administrator or standard user, and maintain an action history thanks to enhanced logging and traceability.

From a functional perspective, the tool offers the addition and management of hardware and software components, their automatic or manual analysis, and the classification of their security status (up-to-date, vulnerable, or requiring an update). Detected vulnerabilities are recorded and highlighted in an interactive dashboard, which also provides statistics and charts for better visualization of the overall system's security posture. The project also includes a reporting feature, enabling the generation of reports in PDF or Excel format, with the option to send email notifications in case of critical vulnerabilities.

Finally, the project was carried out using an agile methodology, with weekly sprints covering design, authentication setup, component management, vulnerability integration, and then report generation and the dashboard. Development constraints, related to a tight schedule and the requirement to use open-source tools, guided the technical and methodological choices. Overall, the solution represents a robust and scalable approach to strengthening information system security within organizations.

Table des matières

Dédicace **2**

Remerciements **3**

Résumé **4**

Abstract **5**

Liste des sigles et acronymes **12**

Introduction générale **13**

1 Contexte général du projet **14**
1.1 Organisme d'accueil 15
1.1.1 Présentation 15
1.1.2 Services et activités de Uniware S.A.R.L 15
1.2 Explication des termes clés 15
1.3 Périmètre du projet 18
1.3.1 But 18
1.3.2 Missions 18
1.3.3 Contraintes 19
1.3.4 Tableau des Sprints 19
1.4 Étude de l'existant 20
1.4.1 Concept métier 20
1.4.2 Problématiques adressées 20
1.4.3 Logiciels similaires 21
1.5 Étude fonctionnelle 21
1.5.1 Objectifs fonctionnels 21
1.5.2 Besoins fonctionnels 22
1.6 Conclusion 24

2 Analyse et conception **25**
2.1 Introduction 26
2.2 Spécification Fonctionnelle Générale 26
2.2.1 Objectif 26
2.2.2 Présentation du système 26
2.2.3 Acteurs du système 26
2.2.4 Modules fonctionnels 27
2.2.5 Relations entre cas d'utilisation 28

2.2.6 Règles de gestion globales . . . 28
2.2.7 Messages standard (globaux) . . . 29
2.2.8 Diagrammes de cas d'utilisation par module . . . 30
2.3 Spécification Fonctionnelle Détaillée . . . 33
2.3.1 Diagramme de classe . . . 33
2.3.2 Cas d'utilisation détaillés . . . 36
2.4 Intégration de la NVD API . . . 39
2.4.1 Présentation générale . . . 39
2.4.2 Composants principaux . . . 39
2.4.3 Flux de fonctionnement . . . 39
2.4.4 Intégrations externes et contraintes . . . 40
2.5 Conclusion . . . 40

3 Etude pratique **41**
3.1 Introduction . . . 42
3.2 Outils . . . 42
3.3 Description détaillée de l'architecture . . . 47
3.3.1 Couche de présentation (Presentation Layer) . . . 48
3.3.2 Couche Web (Web Layer) . . . 48
3.3.3 Couche Service (Service Layer) . . . 48
3.3.4 Couche de persistance (Data Layer) . . . 49
3.3.5 Intégrations externes . . . 49
3.3.6 Fonctionnement global . . . 49
3.4 Réalisation du projet . . . 50
3.4.1 Page de connexion . . . 50
3.4.2 Menu de navigation . . . 51
3.4.3 Tableau de bord statistique . . . 51
3.4.4 Recherche de vulnérabilités . . . 52
3.4.5 Profil utilisateur . . . 54
3.4.6 Gestion des utilisateurs . . . 55
3.4.7 Rapports des utilisateurs . . . 57
3.4.8 Formulaire d'ajout de machine . . . 58
3.4.9 Liste des machines . . . 59
3.4.10 Détails d'une machine . . . 59
3.4.11 Édition d'une machine . . . 61
3.4.12 Détection des vulnérabilités . . . 61
3.4.13 Génération de rapports machines . . . 62
3.4.14 Scan du réseau (Version bêta) . . . 63
3.4.15 Cartographie du réseau (Version bêta) . . . 65
3.4.16 Fonctionnalités implémentées . . . 65
3.4.17 Tableau de bord et visualisation . . . 66
3.4.18 Déploiement et utilisation . . . 66
3.5 Conclusion . . . 66

Conclusion générale **67**

Perspectives **68**

Webographie **69**

Bibliographie **70**

Table des figures

1.1 cvss 16
1.2 le processus d'une vulnérabilité 17

2.1 Diagramme de cas d'utilisation "Gestion des utilisateurs" 30
2.2 Diagramme de cas d'utilisation "Gestion des composants réseau" 31
2.3 Diagramme de cas d'utilisation "Gestion des vulnérabilités" 32
2.4 Diagramme de cas d'utilisation "Gestion des rapports et statistiques" . . . 33
2.5 Diagramme de classe principal 34
2.6 Diagramme de classe - Module d'archivage 35
2.7 Diagramme de séquence - Authentification 36
2.8 Diagramme de séquence - Recherche de vulnérabilités 37
2.9 Diagramme de séquence - Ajout utilisateur 38

3.1 Diagramme de l'architecture du projet 47
3.2 Interface de connexion à l'application Security Component Checker 50
3.3 Menu de navigation de l'application Security Component Checker 51
3.4 Tableau de bord statistique de l'application 52
3.5 Interface de recherche de vulnérabilités CVE 53
3.6 Résultats de recherche de vulnérabilités (partie 1) 53
3.7 Résultats de recherche de vulnérabilités (partie 2) 53
3.8 Interface de consultation du profil utilisateur 54
3.9 Interface de modification du profil utilisateur 54
3.10 Liste des utilisateurs du système 55
3.11 Détails d'un utilisateur spécifique 56
3.12 Formulaire d'ajout d'un nouvel utilisateur 56
3.13 Interface de modification d'un utilisateur 57
3.14 Rapport détaillé des utilisateurs actifs 57
3.15 Archives et historique des modifications des utilisateurs 58
3.16 Métadonnées des rapports utilisateurs 58
3.17 Formulaire d'ajout d'une nouvelle machine à l'inventaire 59
3.18 Liste des machines du système d'information 59
3.19 Section supérieure des détails d'une machine 60
3.20 Section inférieure des détails d'une machine 60
3.21 Interface de modification d'une machine 61
3.22 Tableau des machines vulnérables détectées (partie 1) 61
3.23 Tableau des machines vulnérables détectées (partie 2) 62
3.24 Rapport détaillé des machines actives 62
3.25 Archives et historique des modifications des machines 62
3.26 Métadonnées des rapports machines 63

3.27 Interface de scan réseau - Liste des machines détectées 63
3.28 Backend du scan réseau - Requêtes de vérification 64
3.29 Backend du scan réseau - Insertion des nouvelles machines 64
3.30 Interface de cartographie du réseau d'entreprise 65

Liste des tableaux

1 Tableau des sigles et acronymes 12

1.1 Tableau des sprints et fonctionnalités 19

2.1 Rôles des acteurs 27
2.2 Relations 28
2.3 Règles de gestion 28
2.4 Messages 29

3.1 Tableau des outils utilisés dans le projet 1 42
3.2 Tableau des outils utilisés dans le projet 2 43
3.3 Tableau des outils utilisés dans le projet 3 44

Liste des sigles et acronymes

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
VS	Visual Studio Code
UML	Unified Modeling Language
S.W	Software
H.W	Hardware
M.À.J	Mise À Jour
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
idea	intellij idea

Table 1 – Tableau des sigles et acronymes

Introduction générale

Avec la montée en puissance des cybermenaces et la complexité croissante des environnements numériques en entreprise, il devient indispensable de disposer d'outils modernes, centralisés et automatisés pour renforcer la sécurité des systèmes d'information. C'est dans ce cadre que s'inscrit le projet Security Component Checker, réalisé en 2025 par Anas AIT EL QADI et Mohamed Taha IZMAR dans le cadre d'un stage d'ingénierie.

Ce projet a pour objectif de fournir aux administrateurs et responsables de la sécurité une solution fiable permettant de vérifier l'état de mise à jour des composants logiciels et matériels, tout en détectant les vulnérabilités connues grâce à l'intégration de bases de données de référence telles que la NVD (National Vulnerability Database). L'application se distingue par son approche combinant efficacité, simplicité d'utilisation et conformité aux normes internationales en matière de cybersécurité, notamment le RGPD et le système de notation CVSS.

Conçue avec une architecture robuste reposant sur Java (Spring Boot), Spring MVC/Thymeleaf et une base de données MySQL, la solution intègre un mécanisme d'authentification sécurisé avec gestion des rôles (administrateur, utilisateur). Elle propose un tableau de bord interactif permettant le suivi en temps réel de l'état des composants critiques, la classification de leur statut (à jour, vulnérable ou nécessitant une mise à jour), ainsi que la génération de rapports détaillés sous formats PDF et Excel. Un système de notifications par e-mail est également intégré pour alerter en cas de vulnérabilités critiques.

Ce rapport présente l'ensemble du processus de conception et de développement, en mettant en lumière les différentes étapes suivies dans un cadre méthodologique agile, incluant l'analyse des besoins, la mise en place de l'authentification, la gestion des composants, l'intégration des vulnérabilités et la création du tableau de bord. Ce projet a permis de mettre en pratique des compétences en développement full stack, en sécurité informatique, ainsi qu'en gestion de projet, tout en apportant une solution concrète et évolutive face aux défis actuels de la cybersécurité en entreprise.

Chapitre 1

Contexte général du projet

1.1 Organisme d'accueil

1.1.1 Présentation

Uniware S.A.R.L est une société marocaine basée à Rabat, spécialisée dans les technologies de l'information et de la communication (TIC) et les solutions de télécommunications. Fondée par des experts dans les domaines de l'informatique et des télécoms, l'entreprise se concentre sur l'innovation et la qualité de ses services.

Uniware propose une gamme complète de prestations incluant l'étude, l'audit, le conseil, l'ingénierie et la réalisation de projets dans les secteurs des réseaux, des systèmes de communication et de la sécurité. La société met l'accent sur l'adaptation aux besoins spécifiques de chaque client et sur l'accompagnement à toutes les étapes des projets.

En complément, Uniware offre des services de conception et d'installation de réseaux, d'infrastructures fibre optique, de télédistribution, de vidéosurveillance et de solutions domotiques pour répondre aux besoins croissants des entreprises et des collectivités en matière de connectivité et de sécurité.

1.1.2 Services et activités de Uniware S.A.R.L

La société Uniware propose une large gamme de services, notamment :

- **Intégration de solutions télécoms :** Câblage structuré VDI, installation et sécurisation des réseaux informatiques et téléphoniques, téléphonie IP et classique, systèmes de vidéosurveillance et sécurité physique.
- **Courants faibles :** Étude, conception et installation de systèmes de sécurité incendie, contrôle d'accès, interphonie, détection d'intrusion, télédistribution, vidéosurveillance, sonorisation et domotique.
- **Fibre optique :** Conception, installation et raccordement ou fusion de câbles fibre optique monomode ou multimode pour des solutions très haut débit.
- **Distribution TV :** Installation d'antennes collectives et systèmes de télédistribution pour hôtels, hôpitaux et résidences, permettant la réception TV terrestre, satellite, numérique ou analogique.
- **Optimisation et conseil :** Étude et réorganisation des installations informatiques et télécoms, réduction des coûts télécoms et accompagnement à toutes les étapes des projets.

1.2 Explication des termes clés

CVE :

Il s'agit d'un système de référencement des vulnérabilités connues, maintenu par l'organisation MITRE et soutenu par le Département de la Sécurité Intérieure des États-Unis. Chaque vulnérabilité publique identifiée reçoit un identifiant unique (CVE), par exemple, CVE-2015-0204 pour la faille FREAK.

CVSS :

C'est un système de notation qui permet d'évaluer la gravité d'une vulnérabilité en fonction de critères mesurables. Le score CVSS va de 0 à 10, où 10 correspond aux vulnérabilités les plus critiques . Le CVSS repose sur trois métriques :

La métrique de base évalue les caractéristiques intrinsèques de la vulnérabilité.

La métrique temporelle prend en compte l'évolution des menaces dans le temps.

La métrique environnementale dépend de l'environnement informatique dans lequel se trouve la vulnérabilité.

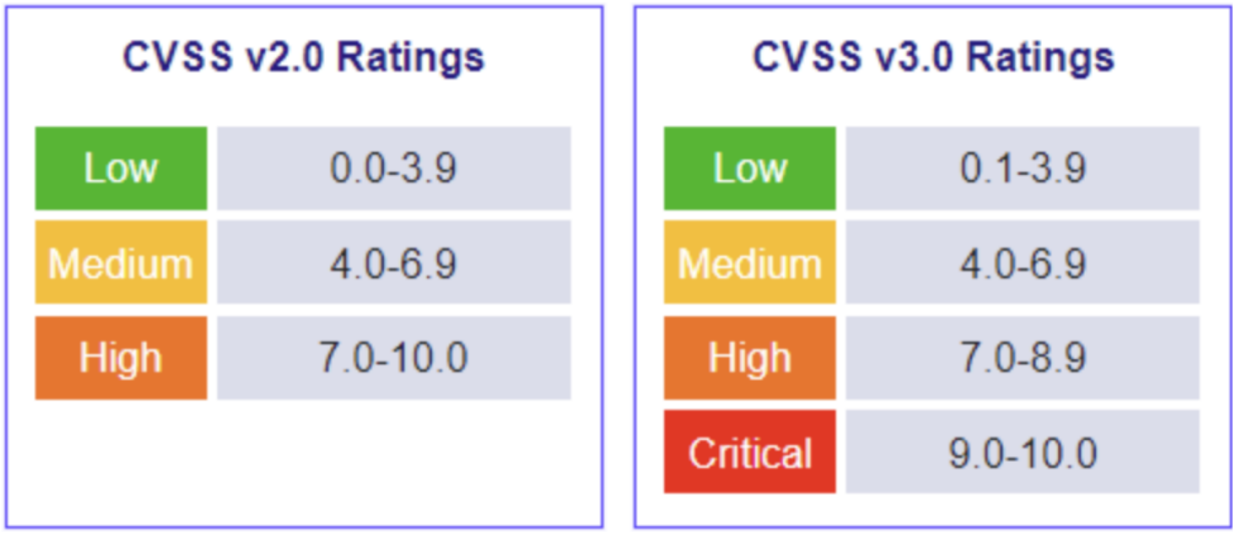


FIGURE 1.1 – cvss

Cette figure compare les classifications de la CVSS pour les versions 2.0 et 3.0 (le niveau de dangerosité) .

CVSS v2.0 Ratings :
Faiblev : 0.0 à 3.9
Moyenne : 4.0 à 6.9
Élevée : 7.0 à 10.0

CVSS v3.0 Ratings :
Faible : 0.1 à 3.9
Moyenne : 4.0 à 6.9
Élevée : 7.0 à 8.9
Critique : 9.0 à 10.0

Exploit :

Un exploit est un code ou un programme utilisé pour exploiter une vulnérabilité informatique. Il permet à un attaquant de tirer parti d'une faille pour accéder aux ressources d'un système, exécuter des commandes malveillantes, ou provoquer un déni de service .

Patch Management :

Il s'agit du processus de gestion des correctifs de sécurité appliqués aux systèmes informatiques. Ce processus comprend l'identification des logiciels obsolètes, la mise à jour des correctifs, et le suivi des installations pour assurer que toutes les machines sont sécurisées contre les failles connues .

Vulnérabilité Zero-day :

C'est une faille de sécurité qui n'a pas encore été rendue publique et pour laquelle aucun correctif n'existe encore. Elle est particulièrement dangereuse car elle peut être exploitée avant qu'un correctif ne soit disponible .

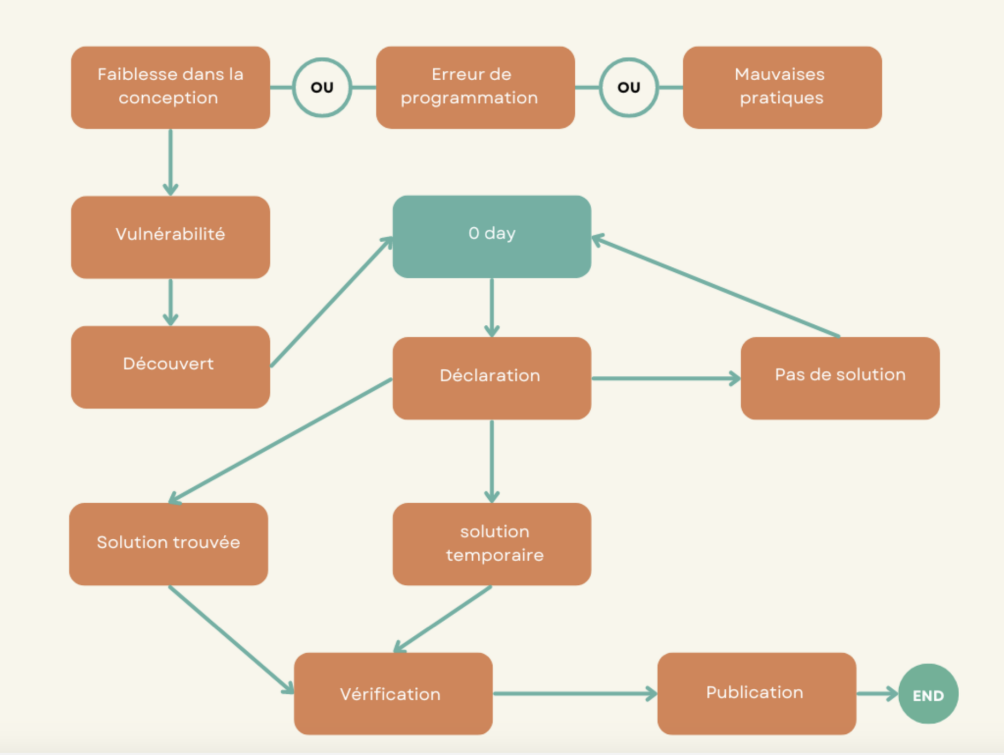


FIGURE 1.2 – le processus d'une vulnérabilité

Cette figure illustre le processus d'une vulnérabilité, elle peut découler de faiblesses de conception, d'erreurs de programmation ou de mauvaises pratiques. Une fois découverte, souvent par un utilisateur ou un chercheur, elle peut être classée comme une vulnérabilité 0 day si elle est inconnue ou sans correctif. La vulnérabilité est ensuite déclarée aux parties concernées. Si aucune solution n'est disponible immédiatement, des mesures temporaires peuvent être prises pour limiter les risques. Une solution définitive est ensuite trouvée et vérifiée avant d'être rendue publique pour informer la communauté.

1.3 Périmètre du projet

1.3.1 But

Le présent projet a pour objectif l'étude, la conception et la réalisation d'un système de vérification des composants du système d'information, permettant de s'assurer que ceux-ci sont à jour en termes de correctifs de sécurité (patchs). Ce système s'adresse principalement aux administrateurs ou responsables de la sécurité informatique, dans le but de renforcer la cybersécurité de l'infrastructure.

Le système intègre plusieurs modules fonctionnels, notamment :

Gestion des utilisateurs et authentification : authentification des utilisateurs et sécurisée et fiable.

Gestion des composants : Suivi et gestion des composants logiciels et matériels

Gestion des vulnérabilités : Identification et retrait des vulnérabilités connues (CVE) via l'intégration d'une API de sécurité (comme NVD API)

Gestion des rapports : Génération des rapports sur l'état des composants et vulnérabilités détectées

Gestion du Tableau de bord : Tableau de bord interactif pour la visualisation des statistiques et alertes de sécurité

1.3.2 Missions

Dans le cadre de ce projet, les missions suivantes ont été menées :

Étude technique : analyse des environnements et technologies adaptés à la cybersécurité (API, formats CVE, frameworks sécurisés).

Étude fonctionnelle : définition des besoins utilisateurs et des scénarios d'usage.

Conception du système : architecture logicielle, base de données, schémas UML.

Développement de la solution : mise en œuvre des différents modules du système.

Déploiement : configuration de l'environnement et publication de l'application.

Documentation : rédaction des guides techniques et d'utilisation.

1.3.3 Contraintes

Le projet a été réalisé en prenant en compte plusieurs contraintes :

Contraintes temporelles : respect des délais fixés pour chaque phase du projet.

Contraintes budgétaires : limitation des ressources matérielles et logicielles.

Contraintes technologiques : intégration avec des API fiables, compatibilité avec les standards de sécurité.

Contraintes légales : respect des normes en matière de protection des données et de sécurité (RGPD, obligations de mise à jour).

1.3.4 Tableau des Sprints

Sprint	**Phase**	**Fonctionnalité**	**Date de livraison**	**Responsables**
0	- initiale	-Création de cahier des charges -Création de Dossier de Spécifications Fonctionnelles Générales/Détaillées(SFG/SFD) -Création de cahier de tests	21/07/2025	Ingénieures IT
1	- conception - réalisation - Test	-Conception du projet -Gestion des utilisateurs et authentification	28/07/2025	Ingénieures dev
2	- conception - réalisation - Test	-Gestion des composants -Gestion des vulnérabilités	11/08/2025	Ingénieures dev
3	- conception - réalisation - Test	-Gestion des rapports -Gestion du Tableau de bord	25/08/2025	Ingénieures dev
4	- conception - réalisation - Test	-amélioration -Test final	01/09/2025	Ingénieures dev

Table 1.1 – Tableau des sprints et fonctionnalités

Le tableau ci-dessus présente l'organisation du projet sous forme de sprints successifs, conformément à la méthodologie agile.

Chaque sprint correspond à une phase de développement bien définie, allant de l'analyse initiale à la livraison finale.

Le sprint 0 est dédié à la préparation et à la documentation, notamment à travers la rédaction du cahier des charges, des spécifications fonctionnelles (SFG/SFD) et du cahier de tests. Les sprints suivants (1 à 4) regroupent des activités de conception, de réalisation et de tests, permettant une progression incrémentale du projet.

Ainsi le sprint 1 introduit la gestion des utilisateurs et l'authentification.

le sprint 2 se focalise sur la gestion des composants et des vulnérabilités.

tandis que le sprint 3 intègre la génération de rapports et la mise en place du tableau de bord.

Enfin, le sprint 4 est consacré à l'amélioration générale et aux tests finaux, garantissant la stabilité et la qualité de l'application avant sa livraison.

1.4 Étude de l'existant

1.4.1 Concept métier

Le concept métier du projet repose sur l'amélioration de la gestion de la sécurité informatique au sein d'un système d'information. L'objectif est de concevoir un système centralisé de vérification de la mise à jour des composants logiciels, notamment en ce qui concerne les patchs de sécurité.

Ce système s'adresse principalement aux administrateurs systèmes ou responsables de la sécurité informatique dans un contexte associatif, institutionnel ou professionnel.

Les principaux besoins métier sont :

- Vérification automatisée des composants installés afin de détecter les vulnérabilités connues (CVE).
- Connexion à une API de vulnérabilités (ex : OSV de Google) pour obtenir les informations actualisées sur les failles de sécurité.
- Suivi de l'état de mise à jour des composants (patch appliqué ou non).
- Gestion des utilisateurs (administrateurs, superviseurs, etc.) avec un système d'authentification sécurisé.
- Production de rapports et tableaux de bord pour assurer le suivi global de la sécurité du système.

1.4.2 Problématiques adressées

Le projet vise à répondre aux problématiques suivantes :

- Difficulté d'identification rapide des composants vulnérables dans un système.
- Manque de centralisation des informations de sécurité provenant de différentes sources (OSV, NVD...).
- Suivi manuel des mises à jour critiques, source d'erreurs humaines.
- Absence de visualisation claire sur l'état global de la sécurité du système d'information.

L'application permettra ainsi d'automatiser, centraliser et visualiser l'état de conformité des composants logiciels face aux vulnérabilités connues.

1.4.3 Logiciels similaires

Une analyse des solutions existantes permet d'identifier des pistes d'amélioration pour le projet :

OSV-Scanner (Google)

- **Description :** outil en ligne de commande pour scanner les dépendances d'un projet logiciel et détecter les CVE connues via l'API OSV.
- **Forces :**
  - Données fiables et mises à jour.
  - Intégration facile avec des projets open-source.
- **Faiblesses :**
  - Pas d'interface graphique utilisateur.
  - Ne propose pas de tableau de bord ou de gestion multi-utilisateur.

VulnWhisperer / OpenVAS

- **Description :** outils de scan de vulnérabilités réseau et système.
- **Forces :**
  - Analyse complète.
  - Génération de rapports automatisés.
- **Faiblesses :**
  - Complexité de configuration.
  - Pas toujours à jour avec les dernières CVE.

Analyse comparative

Ces outils sont souvent puissants, mais peu conviviaux et peu adaptés à une utilisation simplifiée en contexte associatif ou éducatif.

L'application proposée doit donc :

- Fournir une interface web moderne et facile d'utilisation.
- Offrir une vérification simple et rapide des composants système.
- Exploiter une API de vulnérabilités pour garantir des résultats actualisés.
- Proposer des rapports automatisés et des indicateurs visuels.
- Permettre une gestion multi-utilisateur avec des droits spécifiques.

1.5 Étude fonctionnelle

1.5.1 Objectifs fonctionnels

Les fonctionnalités de l'application doivent répondre aux objectifs suivants :

Vérification et surveillance des composants

- Analyser la liste des composants logiciels installés.
- Interroger une base de données publique de vulnérabilités via API.
- Identifier les composants vulnérables et les versions affectées.

Automatisation et alertes

- Lancer automatiquement des vérifications périodiques.
- Notifier les utilisateurs en cas de nouvelle vulnérabilité détectée.
- Générer automatiquement des rapports (PDF, Excel).

Gestion des utilisateurs et sécurité

- Authentification sécurisée.
- Gestion des rôles : administrateur / superviseur.
- Historique des connexions et des actions.

Interface utilisateur intuitive

- Tableau de bord synthétique affichant :
  - (couleur rouge) composants vulnérables
  - (couleur orange) composants à mettre à jour
  - (couleur vert) composants à jour
- Affichage clair des détails de chaque CVE détectée.
- Fonction de recherche et de tri.

Reporting et statistiques

- Génération de statistiques sur l'évolution de la sécurité du système.
- Exportation des rapports en formats standards.
- Visualisation des tendances (graphiques, chronologies).

Archivage et traçabilité

- Conserver l'historique des rapports générés (PDF/Excel) avec leurs métadonnées (date, utilisateur, composants vulnérables détectés).
- Permettre aux administrateurs de consulter ou télécharger les rapports archivés.
- Assurer un stockage sécurisé et horodaté des archives pour audit et contrôle ultérieurs.
- Archiver les entités critiques du système (utilisateurs, composants, alertes, connexions) en cas de suppression ou d'inactivité prolongée.
- Garantir que les données archivées restent accessibles en lecture seule à des fins d'audit ou de récupération.
- Intégrer un système de restauration (rollback) pour réactiver une entité archivée si nécessaire.

1.5.2 Besoins fonctionnels

L'application Web de gestion des vulnérabilités des composants du système d'information est conçue pour répondre aux besoins spécifiques des différents utilisateurs impliqués dans la sécurité informatique. Chaque utilisateur dispose de fonctionnalités adaptées à son rôle.

1.4.2.1 Bloc fonctionnel : Gestion des utilisateurs et authentification

User stories :

- **En tant qu'utilisateur :**
  - Consulter mon profil.
  - Modifier mon profil (données personnelles).
  - Se déconnecter.
- **En tant qu'administrateur :**
  - Lister les profils existants.
  - Ajouter un nouveau utilisateur.
  - Consulter les détails d'un profil.
  - Modifier un profil.
  - Supprimer un profil.
  - Consulter les statistiques.

Règles de gestion :

- RG 01 : Structure d'un utilisateur : Id, Matricule, Nom, Prénom, Photo, Mail, Ville, Numéro de téléphone, Date d'embauche, Role (ADMIN ou USER)
- RG 02 : Un utilisateur peut avoir un seul compte.
- RG 03 : Un e-mail de connexion doit être envoyé à chaque authentification.
- RG 04 : Le mot de passe doit contenir au minimum 8 caractères, incluant au moins une majuscule, une minuscule, un chiffre et un caractère spécial.
- RG 05 : L'administrateur a les mêmes fonctionnalités que l'utilisateur normal.
- RG 06 : Chaque modification ou ajout d'un nouvel utilisateur doit être archivé pour des raisons de traçabilité.

1.4.2.2 Bloc fonctionnel : Gestion des composants logiciels et matériels

User stories :

- En tant qu'utilisateur :
  - Consulter la liste des composants logiciels et matériels.
  - Ajouter manuellement un ou plusieurs composants.
  - Consulter les détails des composants (informations, état, historique des vulnérabilités, etc.).
  - Supprimer les composants.

Règles de gestion :

- RG 07 : L'ajout des composants peut être automatique ou manuel.
- RG 08 : Chaque modification ou ajout d'un nouveau composant doit être archivé.
- RG 09 : Certaines données sont nécessaires pour détecter les vulnérabilités : Nom du composant, Version exacte, Type, OS cible, Architecture.

1.4.2.3 Bloc fonctionnel : Gestion des vulnérabilités (CVE_API)

User stories :

- Consulter les machines vulnérables.
- Consulter les machines dans la période d'attaque.

Règles de gestion :

- RG 10 : Une période d'attaque doit être calculée et stockée (CVSS > 7.0).
- RG 11 : Un mail doit être envoyé en cas de composant vulnérable.
- RG 12 : L'état des machines s'enregistre sous trois formes : composants vulnérables, composants à mettre à jour, composants à jour.

1.4.2.4 Bloc fonctionnel : Gestion des rapports

User stories :

- Imprimer et télécharger des rapports.

Règles de gestion :

- RG 13 : Les rapports doivent être disponibles sous plusieurs formats (PDF, Excel, etc.).
- RG 14 : Les rapports doivent inclure des données de suivi (date d'extraction, nom et matricule de l'utilisateur).

1.4.2.5 Bloc fonctionnel : Gestion des statistiques et tableau de bord

User stories :

- Suivre le nombre total de membres inscrits et de composants.
- Suivre les activités récentes (vulnérabilités, composants vulnérables, composants à mettre à jour, etc.).
- Suivre les machines dans la période d'attaque.

Règles de gestion :

- RG 15 : Les données doivent être automatiquement mises à jour (temps réel ou périodique).
- RG 16 : Le tableau de bord doit présenter les informations de manière synthétique et claire.
- RG 17 : Le tableau de bord peut contenir des graphiques et des tableaux.

1.6 Conclusion

Pour résumer, notre projet sera une réponse adaptée aux besoins grâce à une solution moderne, sécurisée et conviviale. Ce chapitre a posé les fondements de notre approche et des fonctionnalités attendues, établissant ainsi les bases sur lesquelles nous allons développer et mettre en œuvre notre solution innovante.

Chapitre 2

Analyse et conception

2.1 Introduction

La conception joue un rôle essentiel dans la réalisation de solutions innovantes, particulièrement dans le domaine de la cybersécurité où l'architecture et la planification sont cruciales pour assurer la robustesse et l'efficacité d'un système de surveillance réseau. Que ce soit dans le domaine de l'ingénierie réseau, de l'analyse de vulnérabilités ou de la gestion des incidents de sécurité, la conception est le processus clé qui permet de transformer une idée abstraite en une réalité opérationnelle. Ce chapitre se concentre sur l'importance de la conception et explore les différentes étapes et approches nécessaires pour concevoir une plateforme complète de gestion de la sécurité réseau.

2.2 Spécification Fonctionnelle Générale

2.2.1 Objectif

Cette partie a pour objectif de spécifier les fonctionnalités principales de la plateforme de gestion de la sécurité réseau. Il sert de référence pour la conception fonctionnelle, la validation des exigences et la coordination entre les parties prenantes. Cette spécification définit le cadre fonctionnel complet du système Security Component Checker, détaillant les interactions entre les différents acteurs, les modules fonctionnels, les règles de gestion et les messages standards qui assurent le bon fonctionnement de l'application dans un environnement de production.

2.2.2 Présentation du système

La plateforme permet de gérer la sécurité d'un réseau avec trois profils utilisateurs :

- Utilisateur normal : peut consulter les composants, rechercher des vulnérabilités et générer des rapports.
- Administrateur : gère les utilisateurs, les composants réseau, supervise les vulnérabilités et génère des rapports détaillés.
- Système : effectue des analyses automatiques, scanne le réseau et interagit avec les API externes (NVD).

2.2.3 Acteurs du système

Ce tableau présente les différents acteurs du système Security Component Checker et leurs responsabilités respectives. L'Utilisateur normal représente les opérationnels sécurité qui ont besoin d'accéder aux informations sans privileges d'administration. L'Administrateur incarne les responsables sécurité ayant des droits étendus pour configurer et superviser l'ensemble de la plateforme. Le Système représente les processus automatisés qui exécutent les tâches planifiées telles que les scans réseau et l'analyse des vulnérabilités. Enfin, l'API NVD constitue un acteur externe crucial qui fournit les données actualisées des vulnérabilités connues (CVE) essentielles au fonctionnement du système.

Acteur	Description
Utilisateur normal	Consulte les composants réseau, recherche des vulnérabilités, génère des rapports basiques
Administrateur	Gère les utilisateurs et leurs profils, administre les composants réseau, supervise les vulnérabilités, génère des rapports détaillés
Système	Scan automatique du réseau, analyse des vulnérabilités via API NVD, gestion des composants
NVD API	API externe fournissant les données de vulnérabilités (CVE)

Table 2.1 – Rôles des acteurs

2.2.4 Modules fonctionnels

Gestion des utilisateurs et profils

- Consulter, modifier son profil (utilisateur) - Ajouter, consulter, modifier, supprimer des profils (administrateur) - Authentification et gestion des sessions

Gestion des composants réseau

- Scanner et identifier les machines actives sur le réseau - Consulter la liste des composants détectés - Ajouter, modifier, supprimer des composants manuellement - Analyser les ports ouverts et informations système

Gestion des vulnérabilités

- Rechercher des vulnérabilités par mot-clé via l'API NVD - Consulter les machines vulnérables - Suivre l'état des vulnérabilités et périodes d'attaque - Consulter les détails des CVE identifiées

Gestion des rapports et statistiques

- Consulter les statistiques sur les utilisateurs - Générer des rapports sur les composants à mettre à jour - Produire des rapports sur les composants vulnérables - Exporter des rapports en formats PDF ou Excel - Suivi des activités récentes et état du système

Ces modules fonctionnels structurent l'ensemble des capacités offertes par la plateforme Security Component Checker. Le module de gestion des utilisateurs et profils assure la sécurité d'accès et la personnalisation des expériences utilisateur. Le module de gestion des composants réseau permet l'inventaire dynamique du parc informatique avec des capacités de découverte automatique. Le module de gestion des vulnérabilités constitue le cœur métier du système en permettant l'identification et le suivi des failles de sécurité. Enfin, le module de rapports et statistiques offre des capacités avancées de reporting et d'analyse pour supporter la prise de décision stratégique.

2.2.5 Relations entre cas d'utilisation

Cas Principal	**«include» / «extend»**
Se connecter	Validation des identifiants
Modifier profil	Vérification des données
Rechercher vulnérabilités	Interrogation API NVD
Scanner le réseau	Détection des machines actives
Générer rapports	Export PDF/Excel

Table 2.2 – Relations

Ce tableau décrit les relations entre les différents cas d'utilisation du système, mettant en évidence les dépendances fonctionnelles et les extensions de comportement. La relation "Se connecter" inclut la validation des identifiants qui est une condition préalable à toute authentification réussie. La modification de profil étend le comportement de base par une vérification avancée des données saisies. La recherche de vulnérabilités inclut nécessairement l'interrogation de l'API NVD pour obtenir les données actualisées. Le scan réseau inclut la détection des machines actives comme étape fondamentale du processus. Enfin, la génération de rapports étend les fonctionnalités de base par des capacités d'export dans différents formats.

2.2.6 Règles de gestion globales

RG1	Le mot de passe doit contenir au minimum 8 caractères, incluant au moins une majuscule, une minuscule, un chiffre et un caractère spécial
RG2	Un scan réseau automatique doit être exécuté quotidiennement
RG3	Les résultats de vulnérabilités doivent être mis en cache pour optimiser les performances
RG4	Toute modification critique doit être journalisée

Table 2.3 – Règles de gestion

Ces règles de gestion globales définissent les contraintes et politiques qui régissent le fonctionnement de la plateforme Security Component Checker. La règle RG1 impose une politique de mots de passe robuste conforme aux standards de sécurité modernes. La règle RG2 garantit une surveillance continue du réseau grâce à des scans automatiques quotidiens. La règle RG3 optimise les performances du système grâce à une stratégie de caching des résultats de vulnérabilités. La règle RG4 assure la traçabilité des actions critiques grâce à un système complet de journalisation (logging) qui enregistre toutes les modifications importantes pour audit et troubleshooting.

2.2.7 Messages standard (globaux)

Code	Message
MSG 001	Accès refusé
MSG 002	Opération réalisée avec succès
MSG 003	Erreur : données incomplètes
MSG 004	Vulnérabilités trouvées
MSG 005	Aucune vulnérabilité détectée

Table 2.4 – Messages

Ce tableau présente les messages standards utilisés dans l'ensemble de la plateforme Security Component Checker pour garantir une expérience utilisateur cohérente et informative. Le message MSG 001 signale un refus d'accès lorsqu'un utilisateur tente d'accéder à une ressource non autorisée. Le message MSG 002 confirme le succès d'une opération importante. Le message MSG 003 alerte l'utilisateur lorsque des données obligatoires sont manquantes dans un formulaire. Le message MSG 004 informe de la détection de vulnérabilités lors d'un scan ou d'une recherche. Le message MSG 005 indique qu'aucune vulnérabilité n'a été détectée, ce qui peut être tout aussi important pour l'utilisateur que la détection de vulnérabilités. Ces messages standards contribuent à la qualité de l'expérience utilisateur en fournissant un feedback clair et cohérent sur les actions entreprises.

2.2.8 Diagrammes de cas d'utilisation par module

Diagramme de cas d'utilisation module : Gestion des utilisateurs

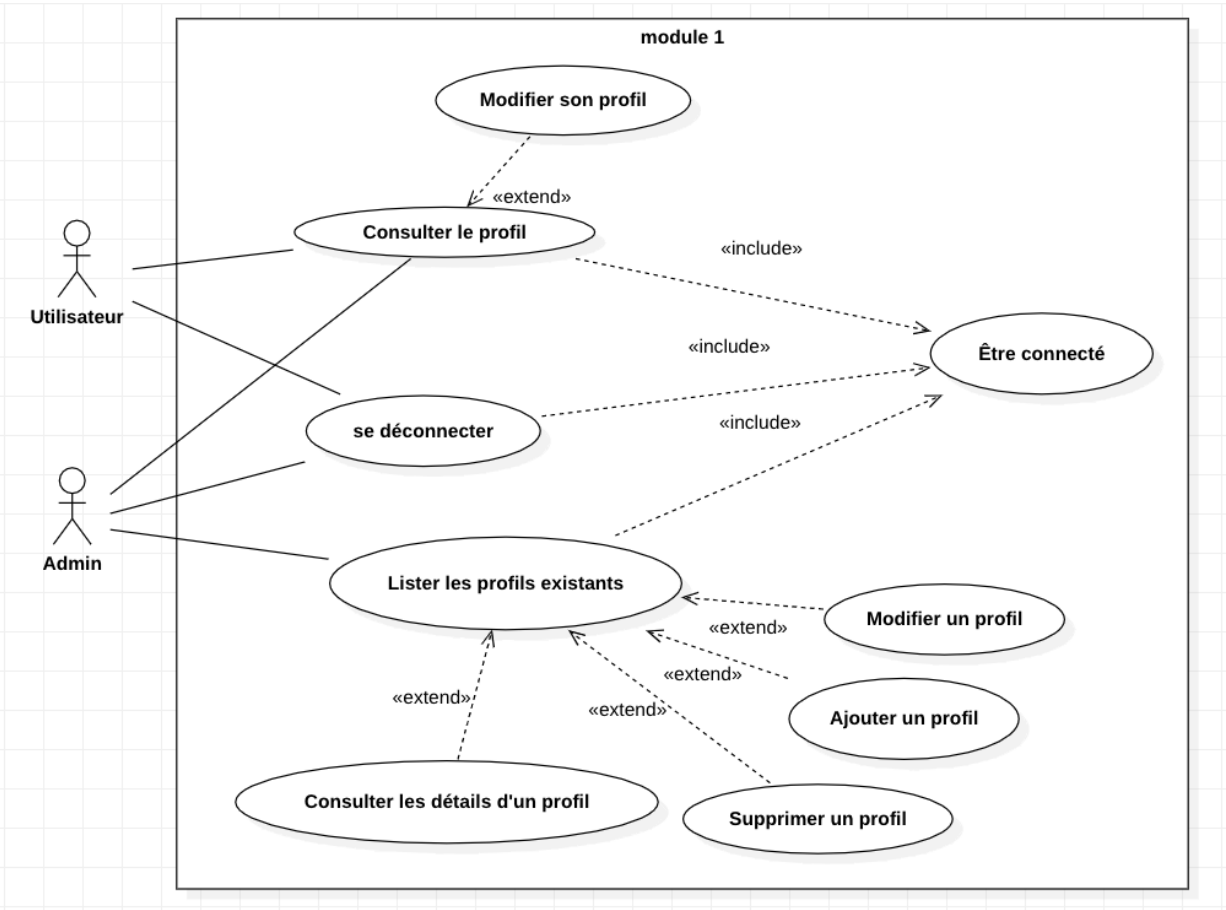


FIGURE 2.1 – Diagramme de cas d'utilisation "Gestion des utilisateurs"

Utilisateur : Rôle générique pour toute personne interagissant avec le système.

Utilisateur normal : Peut consulter son profil, modifier ses informations et se déconnecter.
Administrateur : Gère tous les profils utilisateurs (ajout, modification, suppression).

Relations entre cas :

« Include » pour des sous-étapes obligatoires (ex : être connecté pour modifier un profil).
« Extend » pour des extensions optionnelles (ex : se déconnecter).

Diagramme de cas d'utilisation module : Gestion des composants réseau

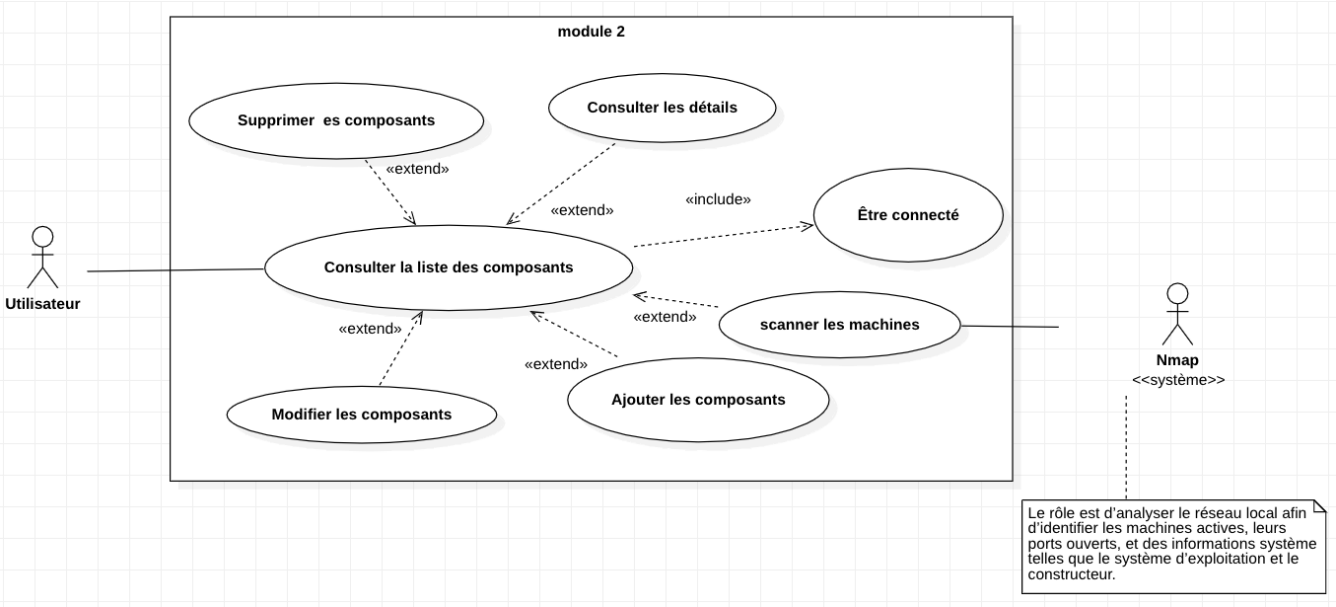


FIGURE 2.2 – Diagramme de cas d'utilisation "Gestion des composants réseau"

onctionnalités principales :

Scan réseau : Détection automatique des machines actives et de leurs caractéristiques.
Gestion manuelle : Ajout, modification et suppression de composants.
Consultation : Liste des composants avec leurs détails techniques.

Relations entre cas :

« Include » pour des étapes obligatoires (ex : être connecté pour scanner le réseau).
« Extend » pour des fonctionnalités avancées (ex : analyse détaillée après scan).

Diagramme de cas d'utilisation module : Gestion des vulnérabilités

Pour les Utilisateurs :

Recherche de vulnérabilités par mot-clé via l'API NVD.
Consultation des machines vulnérables.

Pour l'Administrateur :

Supervision de l'état des vulnérabilités.
Suivi des périodes d'attaque.

Interactions externes :

« System » pour l'interaction avec l'API NVD externe.

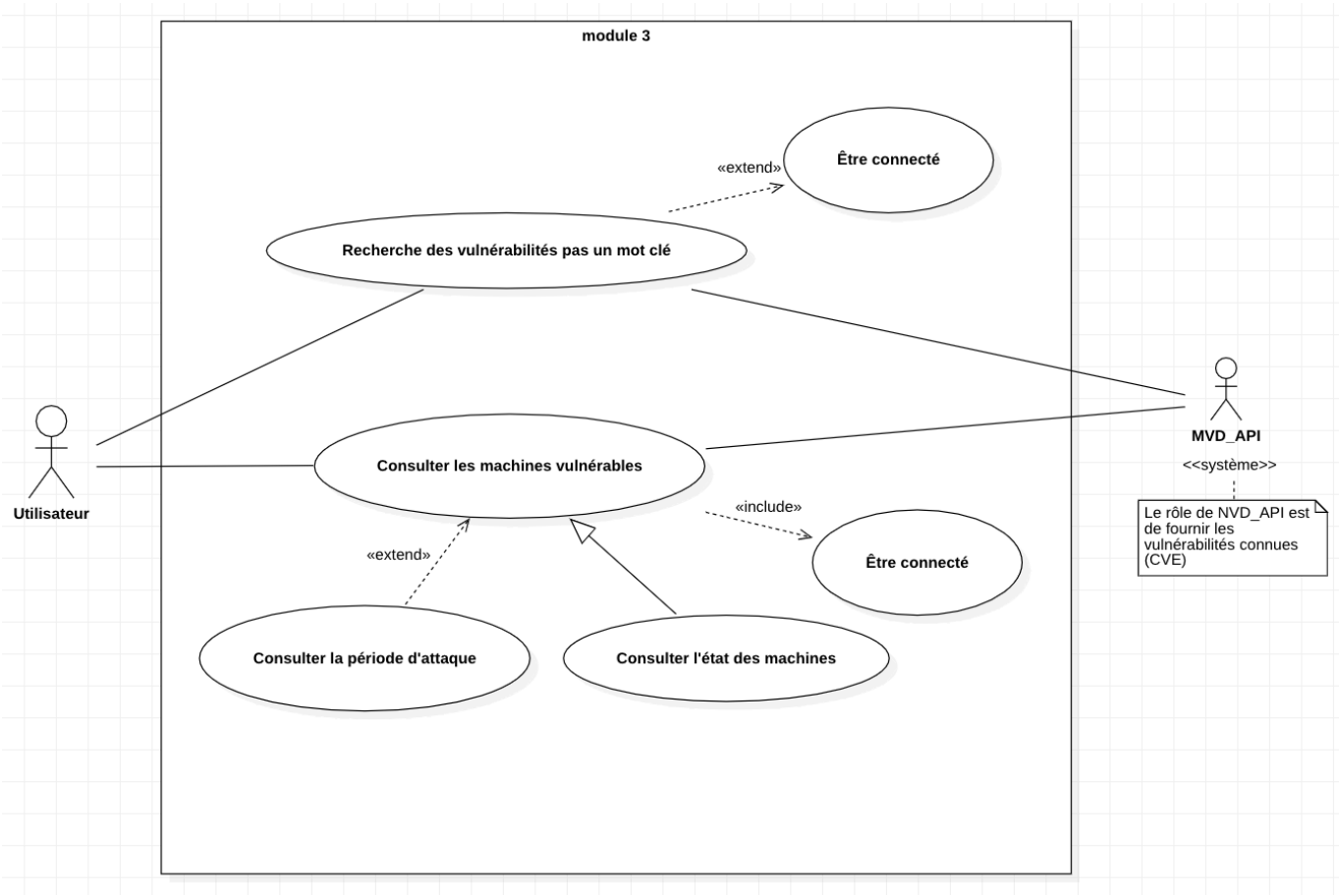


FIGURE 2.3 – Diagramme de cas d'utilisation "Gestion des vulnérabilités"

Diagramme de cas d'utilisation module : Gestion des rapports et statistiques

Participants : Utilisateur normal, Administrateur.

Actions clés :

Consulter les statistiques utilisateurs et système.
Générer des rapports sur les vulnérabilités et composants.
Exporter en PDF/Excel.

Relations :

« Include » pour des étapes obligatoires (ex : être connecté pour générer un rapport).

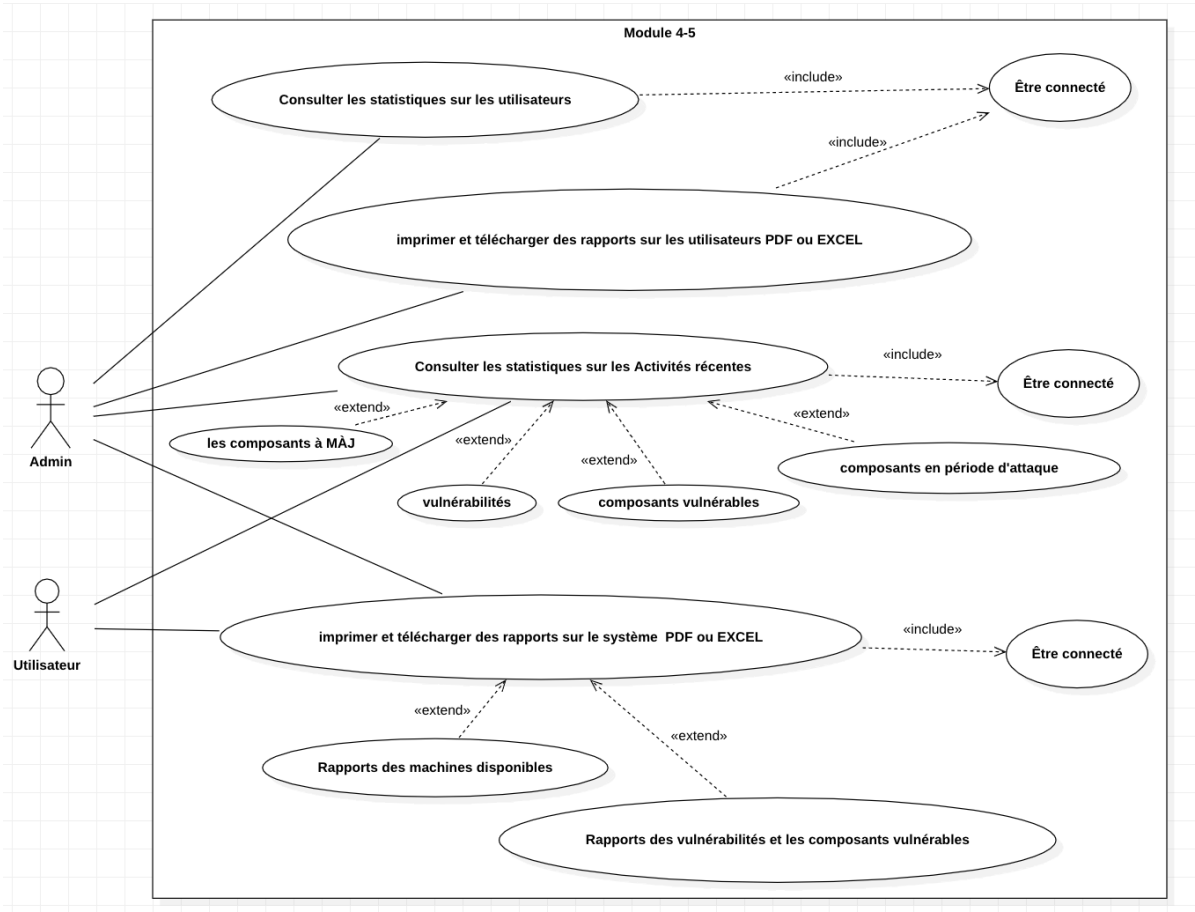


FIGURE 2.4 – Diagramme de cas d'utilisation "Gestion des rapports et statistiques"

2.3 Spécification Fonctionnelle Détaillée

2.3.1 Diagramme de classe

Le schéma de base de données de l'application de gestion de sécurité réseau VScanner comprend plusieurs tables principales interconnectées organisées autour de trois entités fondamentales : les utilisateurs, les machines et les vulnérabilités.

Structure principale

La table **user** constitue le cœur du système d'authentification et de gestion des accès avec les champs suivants :

- `id` : Identifiant unique (bigint)
- `email` et `password` : informations d'authentification
- `nom`, `prenom`, `adresse`, `ville`, `pays` : informations personnelles
- `telephone`, `date_naissance`, `date_embauche` : coordonnées et dates importantes
- `matricule` et `role` : informations professionnelles et niveau de privilège
- `date_creation` : historique de création du compte
- `entreprise_id` : lien vers l'entreprise (clé étrangère)

La table **machine** recense l'ensemble des équipements réseau monitorés avec une riche description technique :

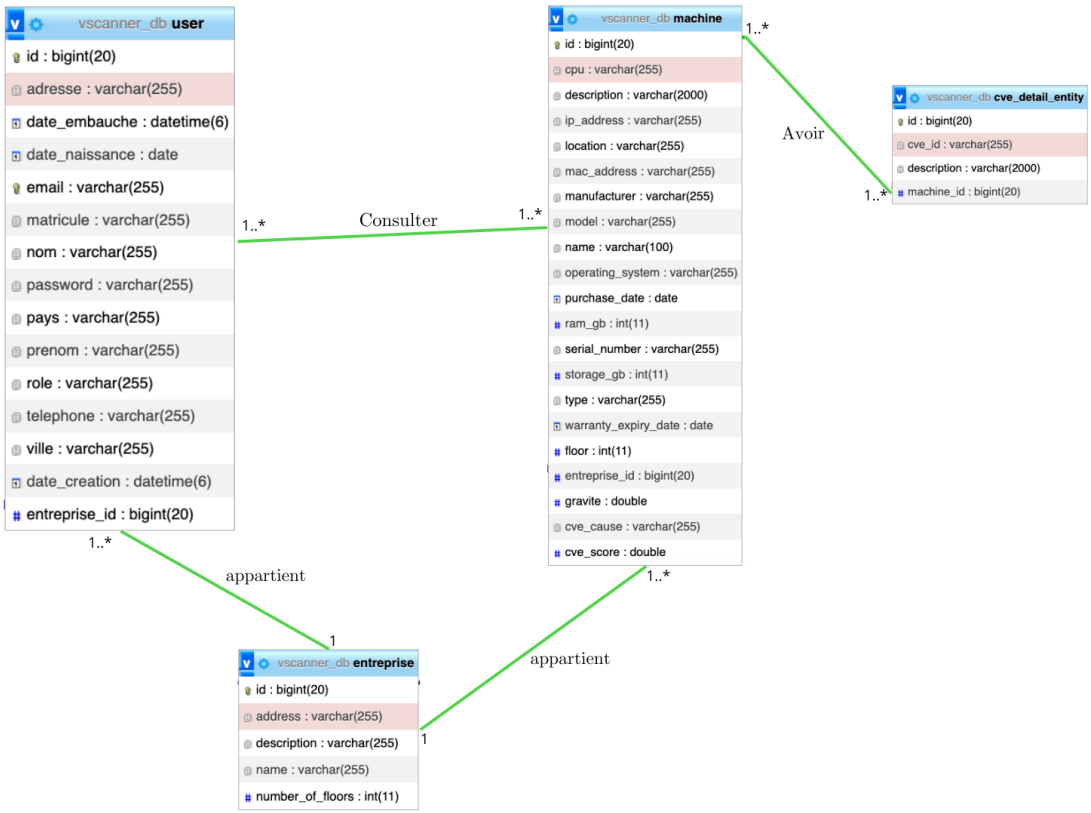


FIGURE 2.5 – Diagramme de classe principal

— `id` : Identifiant unique (bigint)
— `ip_address`, `mac_address` : identifiants réseau
— `name`, `description` : informations descriptives
— `cpu`, `ram_gb`, `storage_gb` : caractéristiques techniques
— `operating_system`, `manufacturer`, `model` : informations système
— `serial_number`, `type` : identification matérielle
— `purchase_date`, `warranty_expiry_date` : informations d'acquisition
— `location`, `floor` : localisation physique
— `gravite`, `cve_cause`, `cve_score` : évaluation des vulnérabilités
— `entreprise_id` : lien vers l'entreprise (clé étrangère)

La table **cve_detail_entity** gère le recensement des vulnérabilités spécifiques :
— `id` : Identifiant unique (bigint)
— `cve_id` : identifiant standard de la vulnérabilité
— `description` : description détaillée de la vulnérabilité
— `machine_id` : lien vers la machine affectée (clé étrangère)

Module d'archivage

Le système intègre également un mécanisme complet d'archivage avec les tables : **machine_archive** pour l'historique des modifications machines :
— `id`, `machine_id` : référence à la machine

— `action_type` : type de modification (création, modification, suppression)
— `archived_at` : date de l'archivage
— Tous les champs techniques de la table machine pour conservation historique

user_archive pour l'historique des modifications utilisateurs :
— `id`, `user_id` : référence à l'utilisateur
— `action_type` : type de modification
— `archived_at` : date de l'archivage
— Tous les champs de la table user pour conservation historique

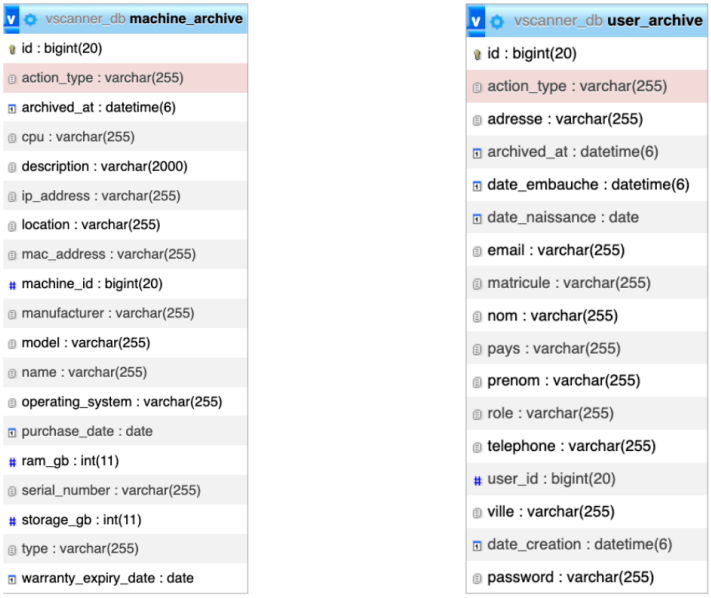


FIGURE 2.6 – Diagramme de classe - Module d'archivage

Relations entre tables

L'architecture relationnelle repose sur des clés étrangères qui créent des liens cohérents entre ces tables :
— Relation entre `user` et `entreprise` via `entreprise_id`
— Relation entre `machine` et `entreprise` via `entreprise_id`
— Relation entre `cve_detail_entity` et `machine` via `machine_id`
— Relations d'archivage entre `machine_archive` et `machine`
— Relations d'archivage entre `user_archive` et `user`

Cette structure normalisée offre une base solide pour les fonctionnalités de monitoring réseau, la gestion des vulnérabilités, l'audit de sécurité et la génération de rapports détaillés.

2.3.2 Cas d'utilisation détaillés

Cas d'utilisation 1 : Authentification utilisateur

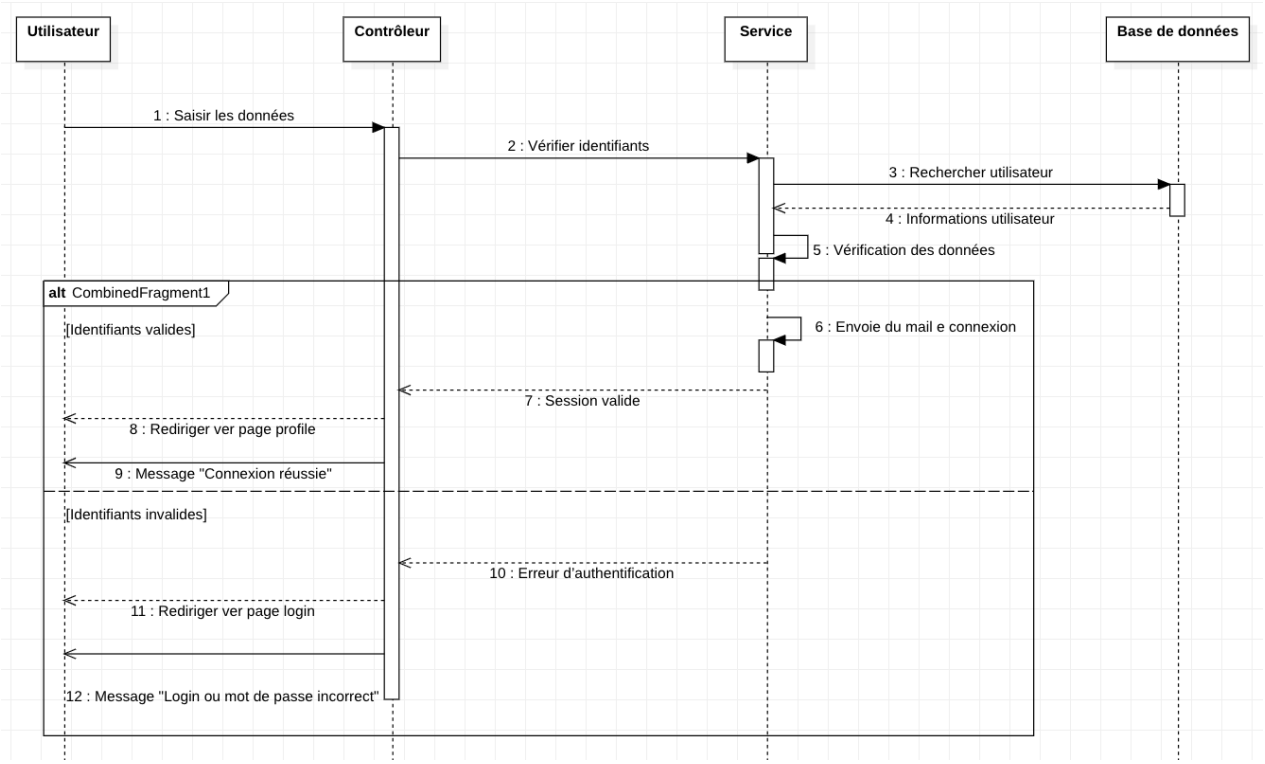


FIGURE 2.7 – Diagramme de séquence - Authentification

Acteur principal : Utilisateur

Description : Permet à un utilisateur de s'authentifier sur la plateforme.

Préconditions : L'utilisateur dispose d'un compte valide.

Postconditions : L'utilisateur est connecté et redirigé vers la page d'accueil.

Déroulement nominal :

1. L'utilisateur saisit ses identifiants.
2. Le système vérifie les credentials.
3. Si valides : création d'une session et redirection.
4. Si invalides : message d'erreur et retour au login.

Cas d'utilisation 2 : Recherche de vulnérabilités

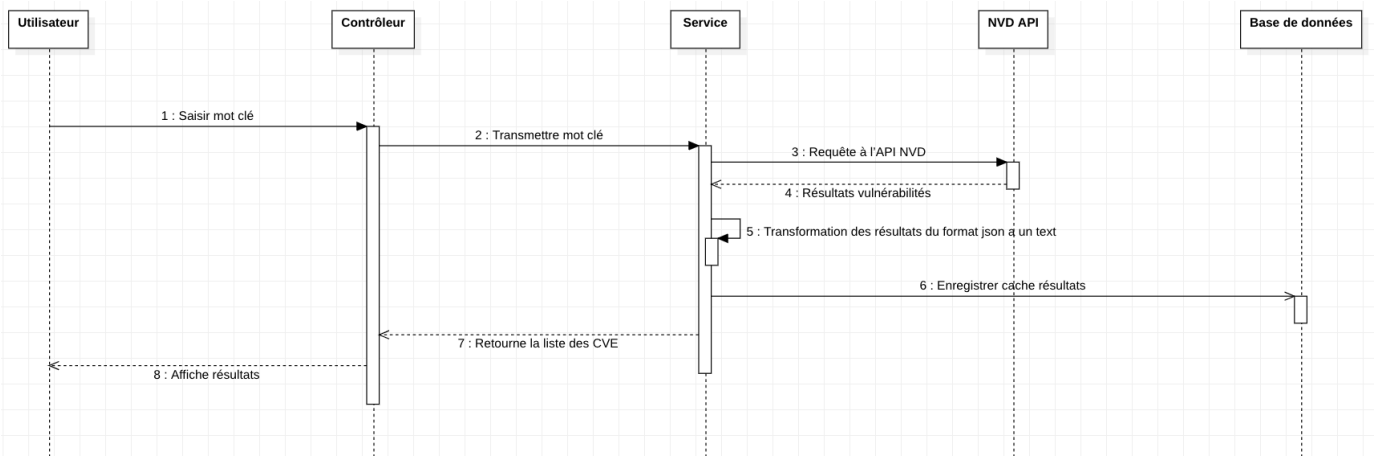


FIGURE 2.8 – Diagramme de séquence - Recherche de vulnérabilités

Acteur principal : Utilisateur/Administrateur

Description : Permet de rechercher des vulnérabilités via l'API NVD.

Préconditions : L'utilisateur est connecté.

Postconditions : Les résultats sont affichés et mis en cache.

Déroulement nominal :

1. Saisie du mot-clé de recherche.
2. Transmission à l'API NVD.
3. Récupération et transformation des résultats JSON.
4. Mise en cache des résultats.
5. Affichage à l'utilisateur.

Cas d'utilisation 3 : Ajout d'un utilisateur (admin)

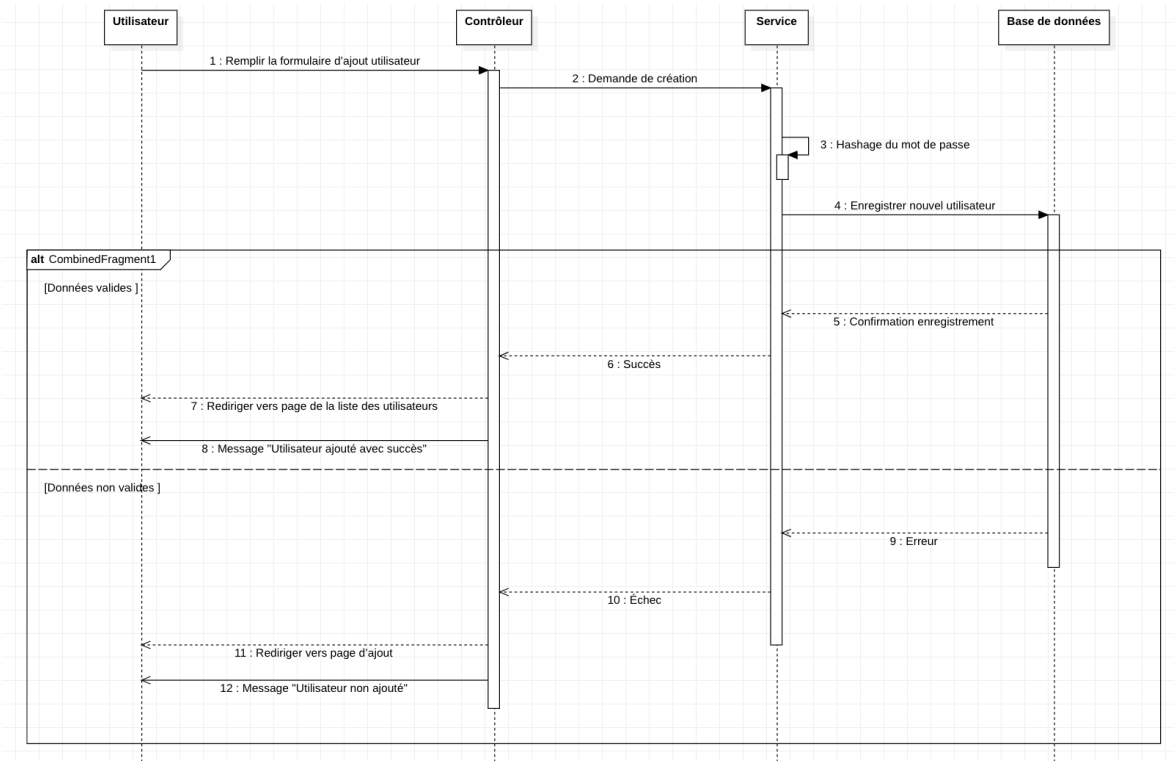


FIGURE 2.9 – Diagramme de séquence - Ajout utilisateur

Acteur principal : Administrateur

Description : Permet à un administrateur d'ajouter un nouvel utilisateur.

Préconditions : L'administrateur est connecté.

Postconditions : Le nouvel utilisateur est créé dans le système.

Déroulement nominal :

1. Remplissage du formulaire d'ajout.
2. Hashage du mot de passe.
3. Enregistrement en base de données.
4. Confirmation et redirection.
5. Si erreur : message d'échec et retour au formulaire.

2.4 Intégration de la NVD API

2.4.1 Présentation générale

L'architecture suit le schéma suivant :

Client/Navigateur → (via contrôleurs) → **NvdService** → NVD API (REST/JSON)
NvdService → transforme les données reçues → CveDetailRepository → Base MySQL

Les résultats sont utilisés ensuite par **VulnerabilityController**, **StatisticsService** et le **Dashboard**.

2.4.2 Composants principaux

NvdService : Connexion à la NVD API via REST.
Utilisation d'un client HTTP (RestTemplate ou WebClient).
Interrogation des endpoints, ex. : /rest/json/cves/2.0.
Désérialisation des données JSON en DTO Java.
Filtrage et extraction : CVE ID, description, score CVSS, niveau de sévérité.

DTOs et Mappers : Conversion JSON → DTO.
Mappers : DTO → entités métiers pour le stockage.

Repositories : CveDetailRepository : détails d'une CVE.
MachineScanRepository : liaison CVE ↔ machine.

Base de données MySQL : Historique des vulnérabilités.
Réduction des appels API et traçabilité.

Utilisation par les autres services : VulnerabilityController : consultation.
StatisticsService : statistiques et graphiques.
EmailService : alertes vulnérabilités critiques.
DashboardController : affichage synthétique.

2.4.3 Flux de fonctionnement

1. L'utilisateur demande une analyse de vulnérabilités.
2. Le contrôleur appelle **NvdService**.
3. NvdService interroge l'API NVD.
4. Conversion JSON → objets métiers.
5. Persistance des vulnérabilités dans MySQL via les repositories.
6. Exploitation par les services : rapports, graphiques, alertes.

2.4.4 Intégrations externes et contraintes

NVD API (REST/JSON) : clé API recommandée.
Rate limiting : mise en cache et archivage local.
Sécurité : HTTPS et protection des données sensibles.

2.5 Conclusion

Nous avons structuré notre application autour de compétences architecturales avec une distinction nette, et l'aide fut fournie par des diagrammes de conception pour donner le développement clair et consistant. A travers plusieurs schémas de cahier de charges, séquences de classe et autres diagrammes, nous avons pu tracer des fondations solides pour démarrer et réussir notre projet.

Chapitre 3

Etude pratique

3.1 Introduction

Idée d'amélioration pour faciliter la vérification des composants du système d'information, la création d'une application web dédiée à la vérification et à la gestion des patches de sécurité est une solution efficace. solution, afin de rendre les processus de sécurité automatisés et simplifiés. L'application il est il est possible de la développer en utilisant des technologies informatiques appropriées. Parmi les exemples figurent l'HTML et à CSS (cascading style sheets).

3.2 Outils

Image	Nom	Description
	HTML	Langage de balisage standard développé par le W3C et le WHATWG. Il permet de structurer le contenu des pages web à l'aide de balises et d'y insérer du texte, des images, des liens, des vidéos et d'autres médias.
	CSS	Langage de styles en cascade permettant de définir l'apparence visuelle des pages HTML : couleurs, polices, marges, grilles, animations et responsive design.
	Bootstrap	Framework front-end open-source facilitant la création d'interfaces réactives grâce à un système de grilles et des composants prédéfinis (boutons, formulaires, menus, etc.).
	JavaScript	Langage de script côté client permettant d'ajouter de l'interactivité aux pages web : animations, formulaires dynamiques, manipulation du DOM et applications web en temps réel.

TABLE 3.1 – Tableau des outils utilisés dans le projet 1

Image	Nom	Description
	MySQL	Système de gestion de bases de données relationnelles (SGBDR) utilisé pour stocker les utilisateurs, documents et événements du projet. Compatible avec Spring Data JPA.
	MAMP	Environnement de développement local (Apache, MySQL, PHP/Python/Perl) permettant de tester les applications et bases de données. Alternative à XAMPP.
	Git / GitHub	Outils de gestion de versions et de collaboration. Git permet de suivre l'évolution du code, et GitHub offre hébergement, gestion de branches, CI/CD et releases.
	Postman	Outil de test et documentation d'API REST. Permet d'envoyer des requêtes HTTP, valider les réponses JSON, automatiser les tests et générer de la documentation.
	IntelliJ IDEA Ultimate	IDE principal du projet, optimisé pour Java et Spring Boot. Offre un refactoring intelligent, une intégration avec MySQL, Git/GitHub et Postman, ainsi qu'un débogage avancé.
	Mailtrap.io	Service de test d'e-mails capturant les envois sortants dans une sandbox sécurisée. Permet d'éviter le spam, d'analyser le rendu HTML et de simuler des scénarios de délivrabilité.

TABLE 3.2 – Tableau des outils utilisés dans le projet 2

Image	Nom	Description
	Microsoft Azure	Plateforme cloud proposant des services de calcul, bases de données, IA et DevOps. Utile pour le déploiement d'applications, la scalabilité et la sécurité.
	Java	Langage orienté objet, portable et typé, fonctionnant sur la JVM. Supporte un large écosystème (Spring, Hibernate, Maven) et favorise la productivité et la portabilité.
	Spring Boot	Framework Java basé sur Spring, facilitant le développement d'applications d'entreprise avec auto-configuration, Spring Data JPA, Spring Security, Spring Cloud et Spring REST Docs.
	StarUML	Outil de modélisation UML supportant divers types de diagrammes : cas d'utilisation, séquence, activité, classes, SysML et ERD.
	NVD API	API donnant accès à la base de données des vulnérabilités (NVD). Utilisée pour vérifier les dépendances et composants logiciels contre des failles de sécurité connues (CVE).
	JaCoCo	Outil d'analyse de couverture de code Java. Intégré avec Maven/Gradle et CI/CD, il fournit des rapports détaillés pour améliorer la qualité et la fiabilité du code.

TABLE 3.3 – Tableau des outils utilisés dans le projet 3

La sélection des outils pour le projet **Security Component Checker** résulte d'une analyse stratégique conciliant performance, sécurité, maintenabilité et évolutivité. Plusieurs critères ont guidé ce choix : maturité technologique, support communautaire, robustesse architecturale, conformité aux standards de sécurité et intégration fluide dans un écosystème cohérent.

Backend : Java / Spring Boot

L'écosystème **Java/Spring Boot** a été retenu comme socle principal. Java, grâce à la JVM, garantit portabilité et gestion mémoire avancée, tandis que Spring Boot accélère le développement via son approche *convention over configuration*. L'usage de **Spring Security** a permis d'intégrer :

— Authentification multifacteur (MFA) ;
— Gestion des rôles (RBAC) ;
— Protection native contre les attaques OWASP (CSRF, injections).

Avec **Spring Data JPA** et **Hibernate**, la couche de persistance bénéficie de transactions distribuées et du cache de second niveau via Ehcache.

Intégration de l'API NVD

L'intégration de l'API **NVD (National Vulnerability Database)** constitue la pierre angulaire du projet. L'architecture repose sur :

— **Spring WebFlux** et **Project Reactor** pour un traitement asynchrone non bloquant ;
— Un cache distribué avec **Redis** pour réduire la latence ;
— Des mécanismes de résilience avec **Resilience4j** (retry, circuit breaker).

Le traitement parallèle des composants est optimisé via les `Streams Java` et `CompletableFuture`.

Développement et CI/CD

Les outils utilisés sont :

— **IntelliJ IDEA Ultimate** : refactoring avancé, analyse statique (SonarLint) ;
— **Git/GitHub** : workflow GitFlow, hooks pré-commit ;
— **GitHub Actions** : CI/CD incluant build Maven multi-modules, tests unitaires (**JUnit 5, Mockito**), intégration (**Testcontainers**), scans de sécurité (**OWASP Dependency-Check, Snyk**), et déploiements *blue-green*.

Testing

— **Postman/Newman** : tests d'API avec validation de schémas JSON ;
— **JaCoCo** : couverture minimale exigée à 80% ;
— **JMeter** : tests de charge avec milliers d'utilisateurs simultanés ;
— **OWASP ZAP** et **Burp Suite** : tests de sécurité (XSS, injections, désérialisation).

Frontend

— **HTML5, CSS3 (Grid, Flexbox, animations)**, **Bootstrap 5** customisé ;
— **JavaScript ES2022** modulaire et **Thymeleaf** pour un rendu sécurisé ;
— **Chart.js** et **D3.js** pour la visualisation des données.

Base de Données et Environnement

- **MySQL 8.0** : support JSON, CTE, window functions, cluster Galera pour haute disponibilité ;
- **MAMP Pro** : environnement local avec Apache et MySQL tunés ;
- **Mailtrap.io** : sandbox d'emails avec analyse et webhooks.

Déploiement sur Microsoft Azure

L'architecture cloud exploite plusieurs services :

- **Azure App Service** : hébergement avec autoscaling et slots de déploiement ;
- **Azure Database for MySQL** : HA, sauvegardes automatiques, chiffrement TDE ;
- **Azure Cache for Redis** : cache distribué, persistance et géo-réplication ;
- **Azure Application Gateway** : load balancing + WAF (OWASP 3.1) ;
- **Azure Monitor** & **Application Insights** : tracing distribué et alertes KQL ;
- **Azure Security Center** & **Defender for Cloud** : analyse de vulnérabilités et détection des menaces ;
- **Azure Active Directory** : SSO, provisioning JIT, accès conditionnel ;
- **Azure Key Vault** : gestion sécurisée des secrets et certificats ;
- **Azure DevOps** : CI/CD avancée (canary releases, feature flags).

Outils Complémentaires

- **StarUML** : modélisation UML (composants, séquences, déploiement) ;
- **Nmap** : scanning réseau automatisé (export XML) ;

3.3 Description détaillée de l'architecture

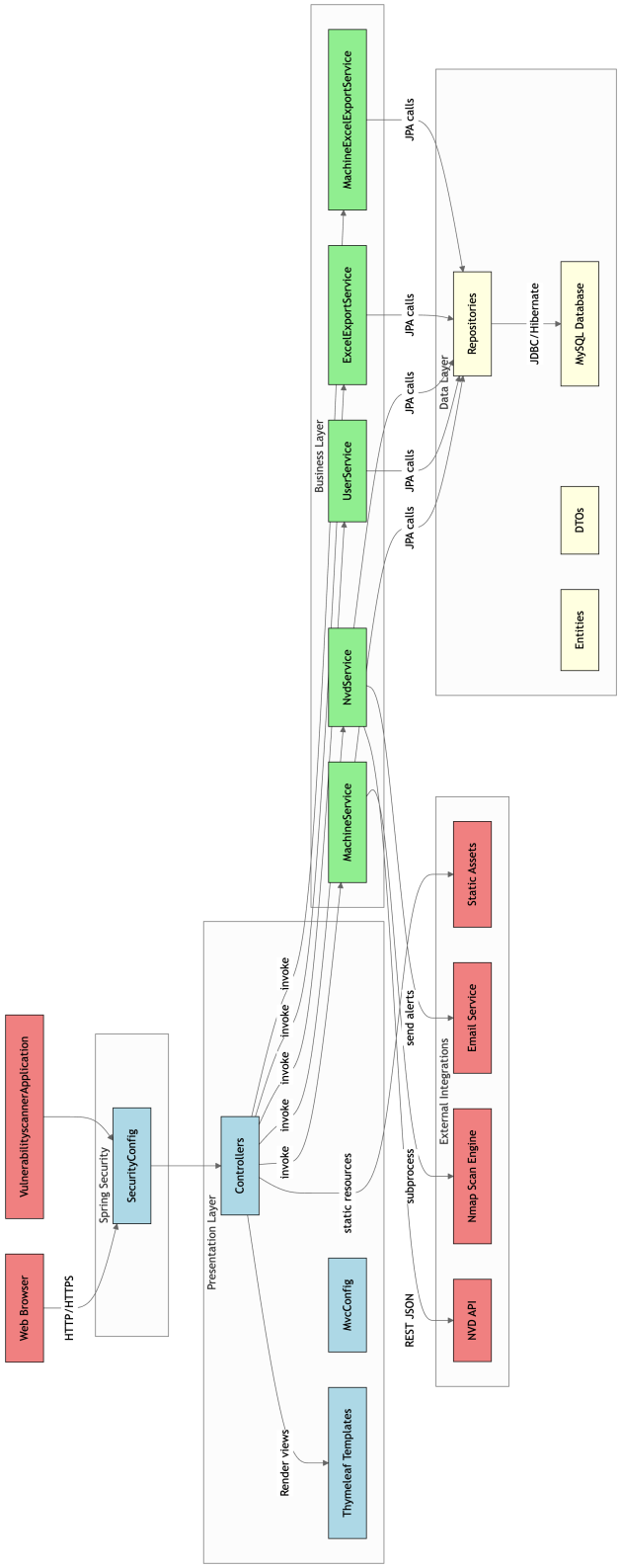


FIGURE 3.1 – Diagramme de l'architecture du projet

L'architecture du projet **Security Component Checker** suit une approche multicouche (multi-tier), inspirée du modèle MVC (Model-View-Controller) et des bonnes pratiques de conception logicielle dans les applications Spring Boot. Elle se compose principalement de quatre niveaux : la couche de présentation, la couche web, la couche service et la couche de persistance (data layer), avec des intégrations externes pour enrichir les fonctionnalités.

3.3.1 Couche de présentation (Presentation Layer)

La couche de présentation est représentée par le navigateur/client (Browser/Client), qui interagit avec l'application via des requêtes HTTP/HTML. Elle inclut également les gabarits **Thymeleaf** pour la génération dynamique de pages web et les ressources statiques (CSS, JS, images). Ces éléments permettent d'assurer une interface utilisateur ergonomique et responsive.

Cette couche constitue le point d'entrée visuel du système et offre une expérience fluide aux utilisateurs lors de la navigation.

3.3.2 Couche Web (Web Layer)

Cette couche repose sur une série de contrôleurs Spring MVC, chacun étant responsable d'une partie fonctionnelle spécifique :

LayoutController et DashboardController : gestion de l'affichage général et des tableaux de bord.

EnterpriseController : gestion des informations liées aux entreprises enregistrées.

VulnerabilityController et ScanController : détection et analyse des vulnérabilités.

ProfileController et AuthController : gestion des profils utilisateurs et du processus d'authentification.

MachineController et StatisticsController : gestion des machines et génération des statistiques globales.

Chaque contrôleur reçoit les requêtes du client. Il appelle les services appropriés, puis renvoie les résultats vers les vues (Thymeleaf templates).

3.3.3 Couche Service (Service Layer)

La couche service contient la logique métier de l'application. Elle regroupe plusieurs services spécialisés, chacun responsable d'une fonctionnalité précise :

UserService et MachineService : gestion des utilisateurs et des machines.

VulnService et StatisticsService : analyse des vulnérabilités et calcul des statistiques.

NetworkScanService et EmailService : scan réseau (via Nmap) et envoi de notifications par email.

ExcelExportService, MachineExcelExportService et UserArchiveService : export des données et archivage des résultats.

MachineArchiveService et EnterpriseService : archivage et gestion des entités associées.

Cette couche constitue le cœur de l'application. Elle contient les règles de gestion, la coordination entre les composants et les appels aux intégrations externes.

3.3.4 Couche de persistance (Data Layer)

La couche de persistance repose sur Spring Data JPA et des repositories spécialisés pour chaque entité. Elle assure la communication avec la base de données **MySQL** via JDBC.

Parmi les repositories principaux :

UserRepository et MachineRepository : gestion des utilisateurs et des machines.

CveDetailRepository et MachineScanRepository : stockage des résultats liés aux vulnérabilités et aux scans.

UserArchiveRepository, MachineArchiveRepository et EnterpriseRepository : archivage et gestion des données associées.

Ces repositories manipulent les entités métiers (**User, Machine, CveDetail, etc.**). Ils garantissent une structure claire et normalisée de la base de données.

3.3.5 Intégrations externes

L'application est enrichie par plusieurs outils et services externes :

NVD API (REST/JSON) : consultation en temps réel de la base officielle des vulnérabilités (CVE).
Nmap CLI : scan réseau pour détecter les machines et analyser leur état de sécurité.

SMTP/Email Server : envoi automatisé de notifications aux administrateurs en cas de vulnérabilité critique.

Report Libraries (PDF/Excel) : génération de rapports d'audit exploitables par les équipes de sécurité.

3.3.6 Fonctionnement global

Le processus global suit le flux suivant :

L'utilisateur interagit via l'interface web (navigateur, templates Thymeleaf).

Les requêtes sont traitées par les contrôleurs (Web Layer).

Les contrôleurs invoquent les services métiers (Service Layer) qui appliquent les règles de gestion et coordonnent les actions.

Les services communiquent avec les repositories pour persister ou récupérer des données depuis la base MySQL.

Les services peuvent interroger des intégrations externes comme l'API NVD ou utiliser Nmap et SMTP pour déclencher des actions spécifiques.

Les résultats sont envoyés au client sous forme de pages web dynamiques, de rapports PDF/Excel ou de notifications.

3.4 Réalisation du projet

Le projet **Security Component Checker** a été développé dans le cadre d'un stage d'été 2024/2025. Il s'agit d'un système de vérification des composants du système d'information en matière de patchs de sécurité. Cette section présente les aspects techniques et fonctionnels de la réalisation de l'application.

3.4.1 Page de connexion

L'interface d'authentification (Figure 3.2) a été conçue pour offrir une expérience utilisateur simple et sécurisée. Elle comprend :

- Un champ pour l'adresse email professionnelle
- Un champ pour le mot de passe
- Une option "Maintenir la connexion" (remember me)
- Un lien de récupération de mot de passe oublié
- Une option pour les nouveaux employés nécessitant un accès

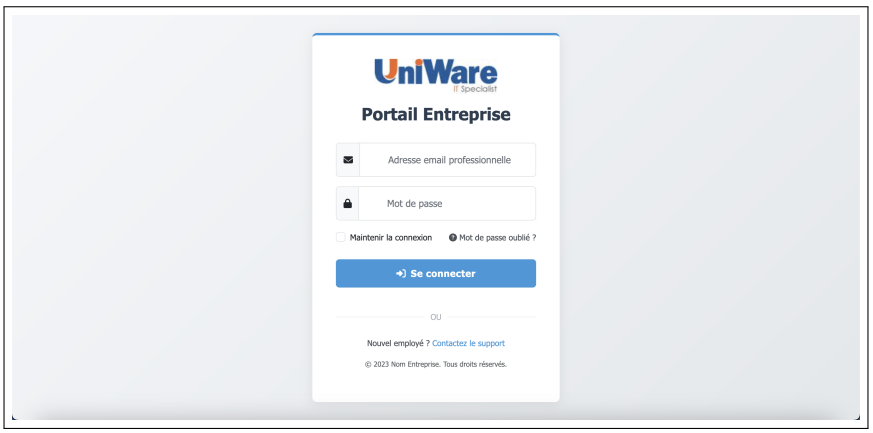


FIGURE 3.2 – Interface de connexion à l'application Security Component Checker

3.4.2 Menu de navigation

Le menu latéral (Figure 3.3) offre une navigation intuitive entre les différentes fonctionnalités de l'application, organisées en deux sections principales :

— **ADMINISTRATION** : regroupant la gestion des utilisateurs et les statistiques
— **PERSONNEL** : incluant le profil utilisateur, la recherche CVE, la gestion des machines et la cartographie

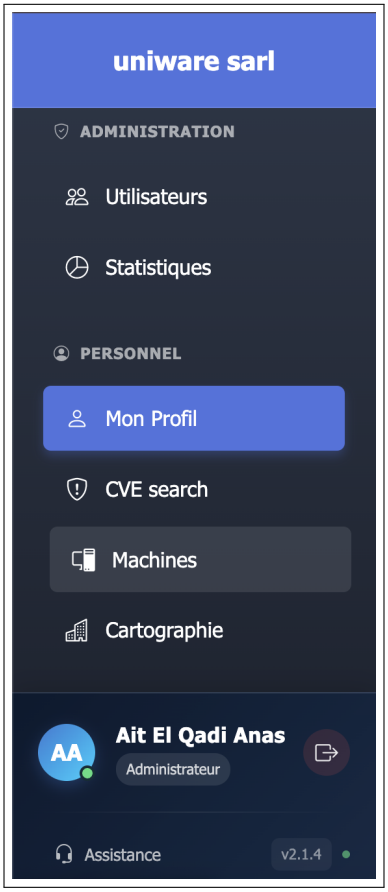


FIGURE 3.3 – Menu de navigation de l'application Security Component Checker

3.4.3 Tableau de bord statistique

L'interface du tableau de bord (Figure 3.4) offre une vue d'ensemble complète du parc informatique et des vulnérabilités détectées :

— Indicateurs clés de performance (KPI) sur l'état du parc
— Graphiques d'évolution des vulnérabilités par mois
— Répartition des vulnérabilités par sévérité et par type
— Filtres avancés pour l'analyse personnalisée des données
— Détail des machines avec leur état de sécurité

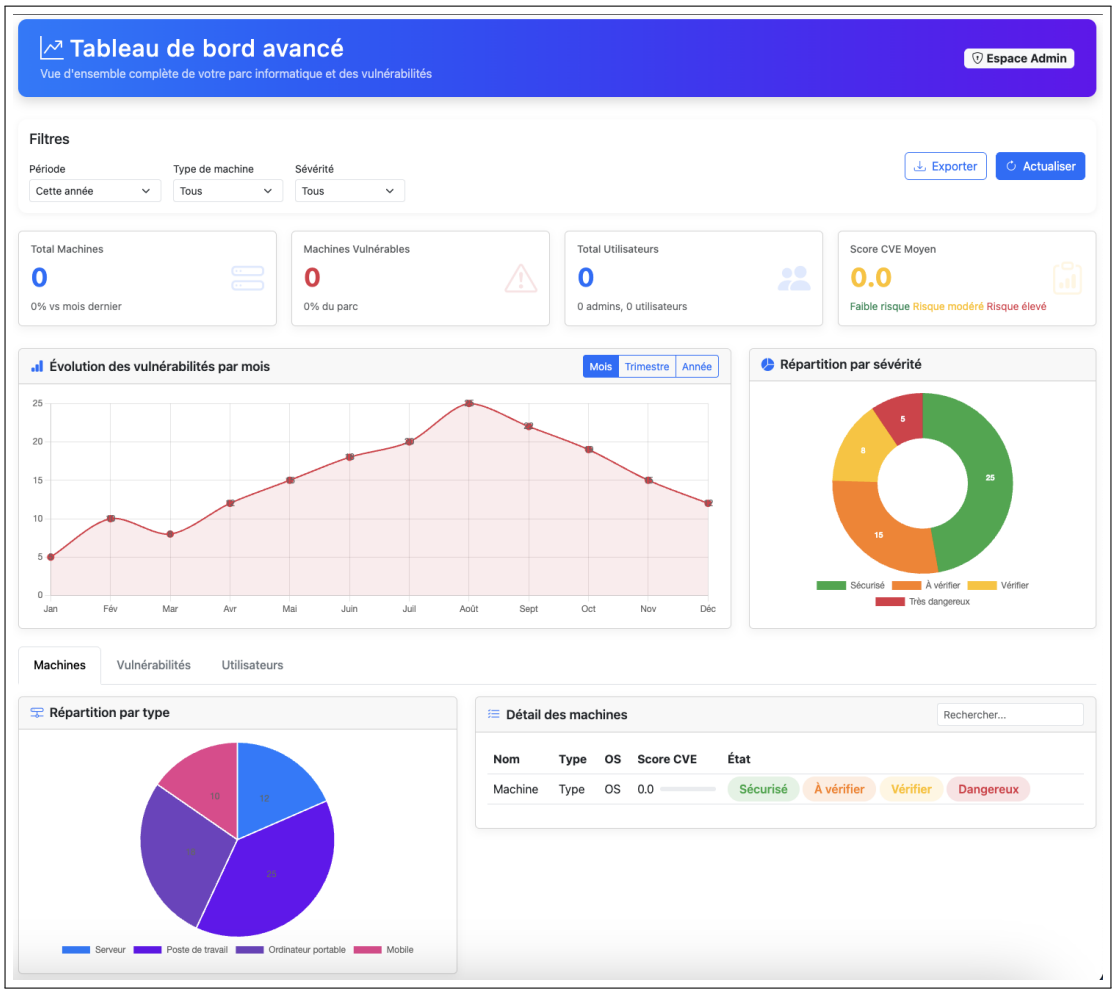


FIGURE 3.4 – Tableau de bord statistique de l'application

3.4.4 Recherche de vulnérabilités

Interface de recherche

L'interface de recherche de vulnérabilités CVE (Figure 3.5) permet aux utilisateurs de rechercher des failles de sécurité spécifiques dans la base de données :

- Champ de recherche par mot-clé
- Accès rapide à la base de données des CVE
- Interface simplifiée pour les recherches ponctuelles

Résultats de recherche

Les résultats de recherche (Figures 3.6 et 3.7) présentent les vulnérabilités correspondantes aux critères de recherche :

- Liste détaillée des CVE trouvées
- Informations techniques complètes (ID, description, dates)
- Scores CVSS et niveau de gravité
- Vecteurs d'attaque et impacts

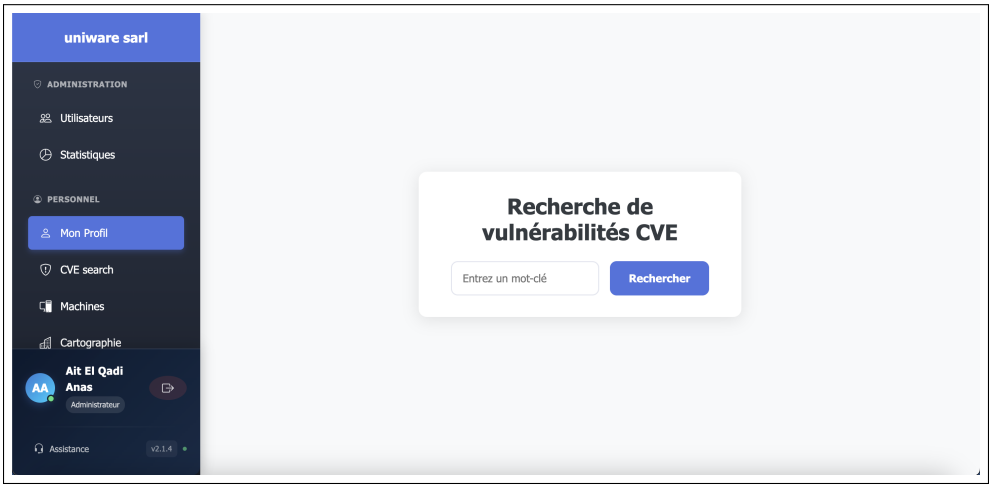


FIGURE 3.5 – Interface de recherche de vulnérabilités CVE

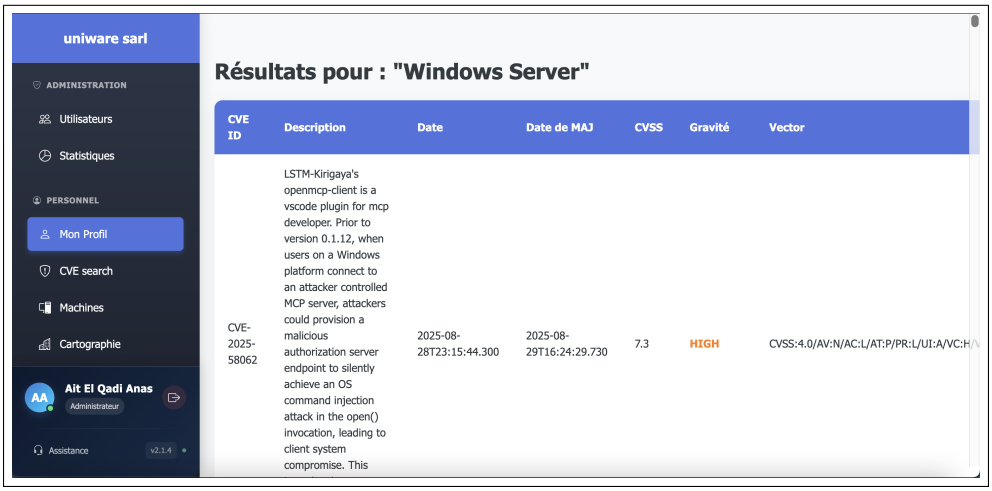


FIGURE 3.6 – Résultats de recherche de vulnérabilités (partie 1)

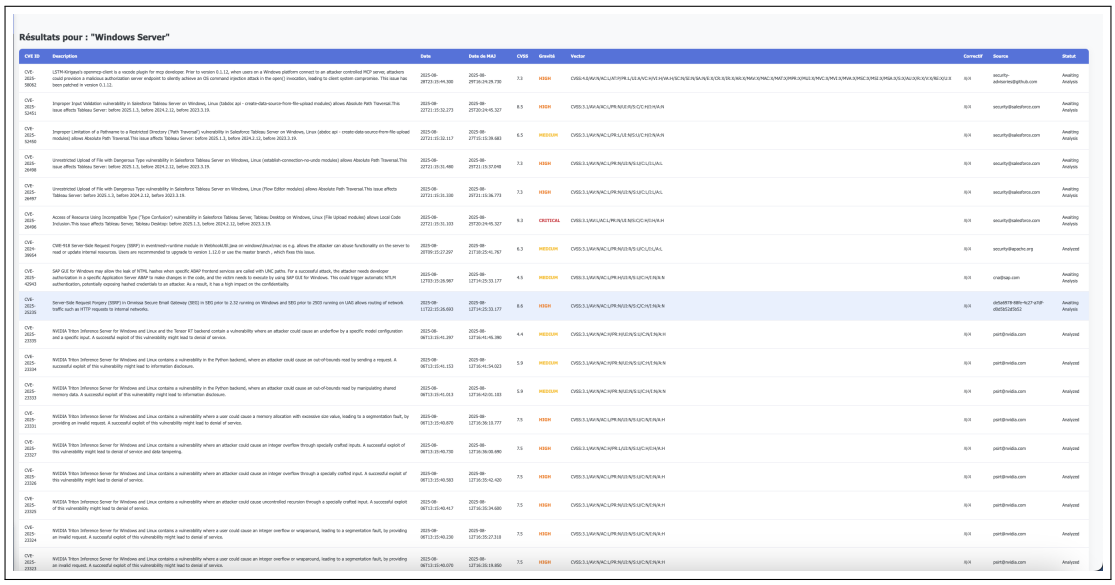


FIGURE 3.7 – Résultats de recherche de vulnérabilités (partie 2)

3.4.5 Profil utilisateur

Consultation du profil

L'interface de consultation du profil (Figure 3.8) permet à chaque utilisateur de visualiser ses informations personnelles :

— Nom complet et coordonnées
— Adresse email professionnelle
— Informations de contact (téléphone, adresse)
— Localisation (ville, pays)

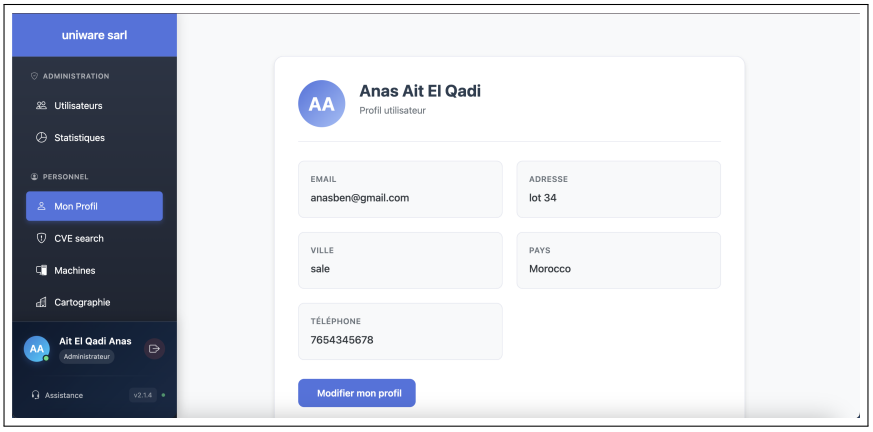


FIGURE 3.8 – Interface de consultation du profil utilisateur

Modification du profil

L'interface de modification du profil (Figure 3.9) permet aux utilisateurs de mettre à jour leurs informations personnelles :

— Modification des données personnelles (nom, prénom)
— Mise à jour des coordonnées (email, téléphone)
— Changement des informations de localisation (adresse, ville, pays)

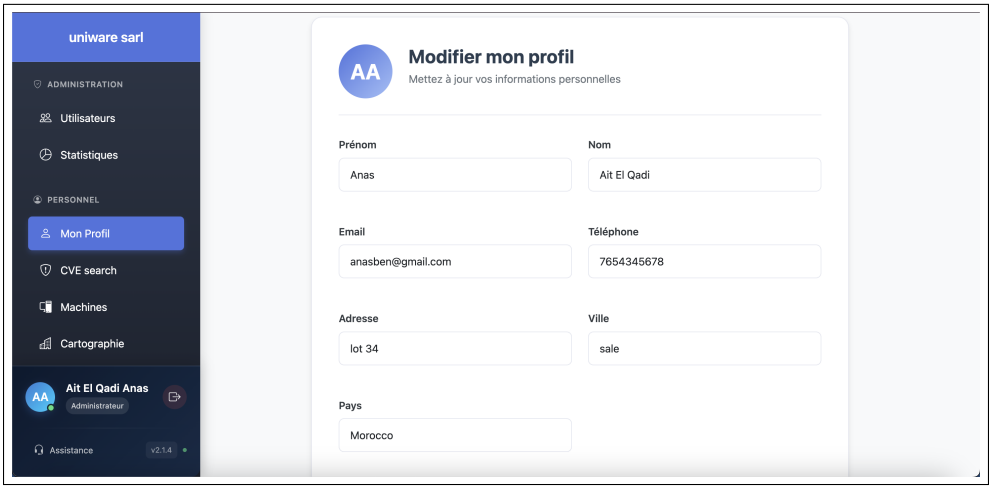


FIGURE 3.9 – Interface de modification du profil utilisateur

3.4.6 Gestion des utilisateurs

Liste des utilisateurs

L'interface de gestion des utilisateurs (Figure 3.10) permet aux administrateurs de visualiser l'ensemble des comptes utilisateurs avec leurs caractéristiques principales :

- Liste tabulaire de tous les utilisateurs
- Informations essentielles (nom, email, rôle)
- Statut de chaque compte (actif/inactif)
- Actions disponibles pour chaque utilisateur

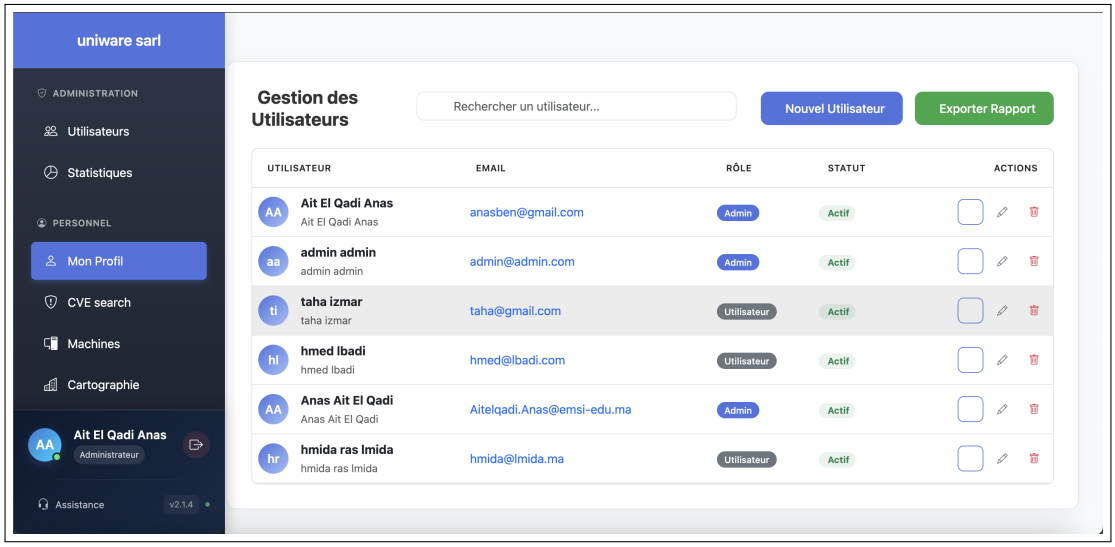


FIGURE 3.10 – Liste des utilisateurs du système

Détails d'un utilisateur

L'interface de détails d'un utilisateur (Figure 3.11) présente une vision complète des informations d'un compte spécifique :

- Identité complète de l'utilisateur
- Informations de contact (email)
- Matricule unique d'identification
- Date d'embauche et rôle dans l'organisation

Ajout d'un utilisateur

L'interface d'ajout d'un nouvel utilisateur (Figure 3.12) permet aux administrateurs de créer de nouveaux comptes avec toutes les informations nécessaires :

- Formulaire complet de saisie des informations personnelles
- Champs pour les données d'identification (nom, prénom)
- Informations de contact (email, téléphone)
- Données de localisation (adresse, ville, pays)
- Dates importantes (naissance, embauche)

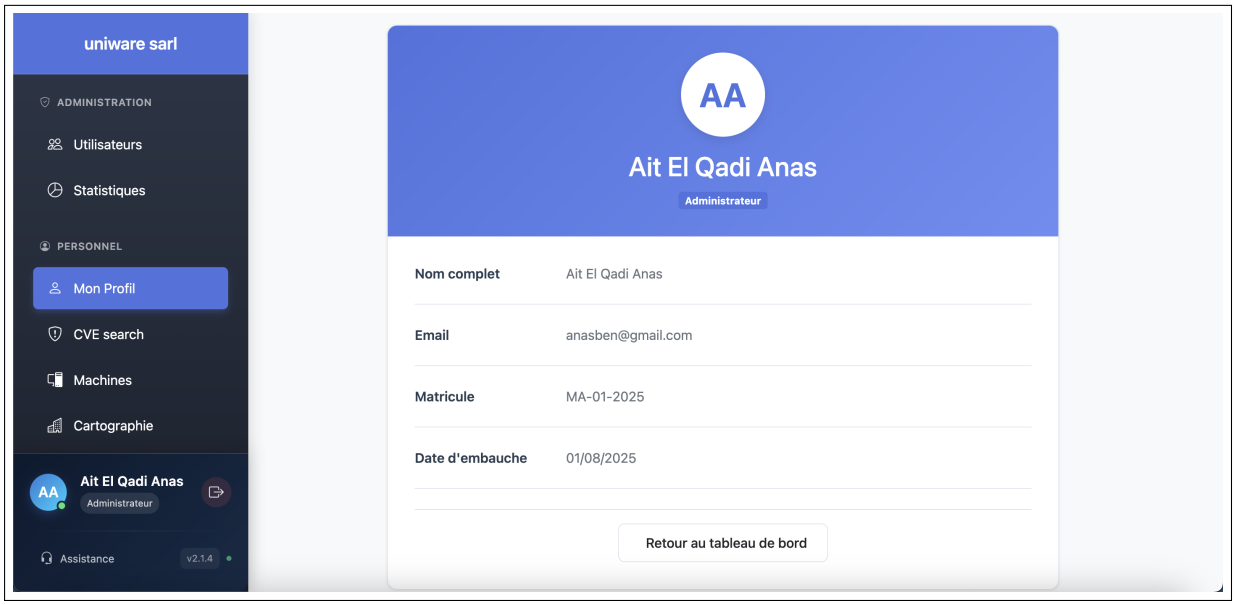


FIGURE 3.11 – Détails d'un utilisateur spécifique

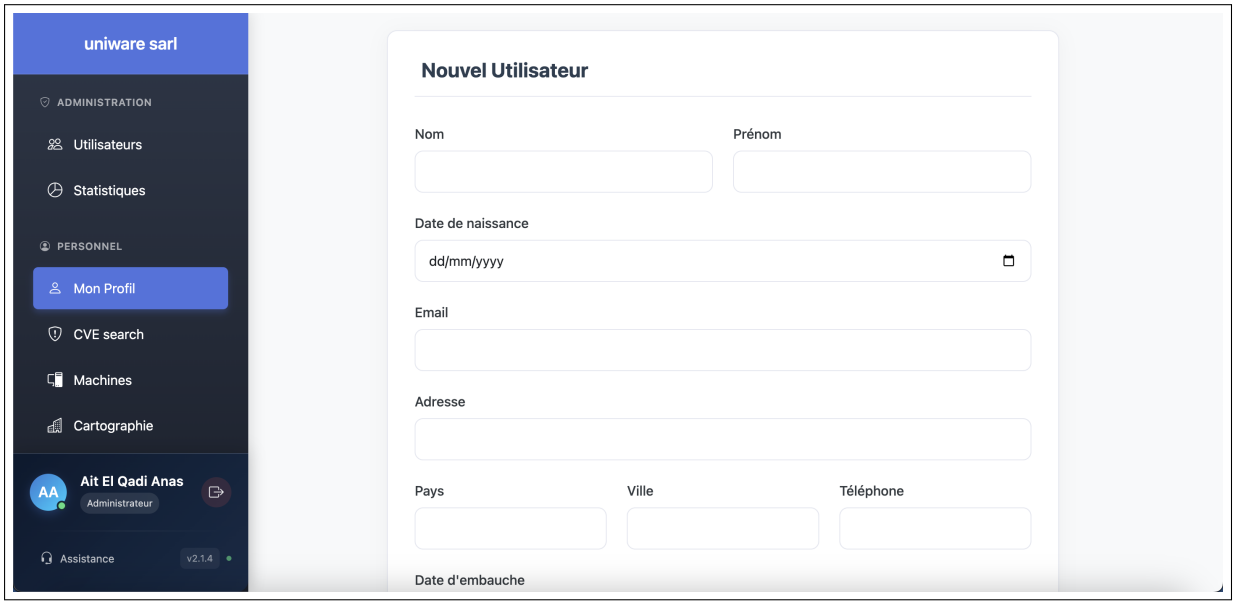


FIGURE 3.12 – Formulaire d'ajout d'un nouvel utilisateur

Modification d'un utilisateur

L'interface de modification d'un utilisateur (Figure 3.13) permet aux administrateurs de mettre à jour les informations d'un compte existant :

— Édition des informations personnelles
— Modification des coordonnées et de l'email
— Changement du rôle et des permissions
— Option de réinitialisation du mot de passe

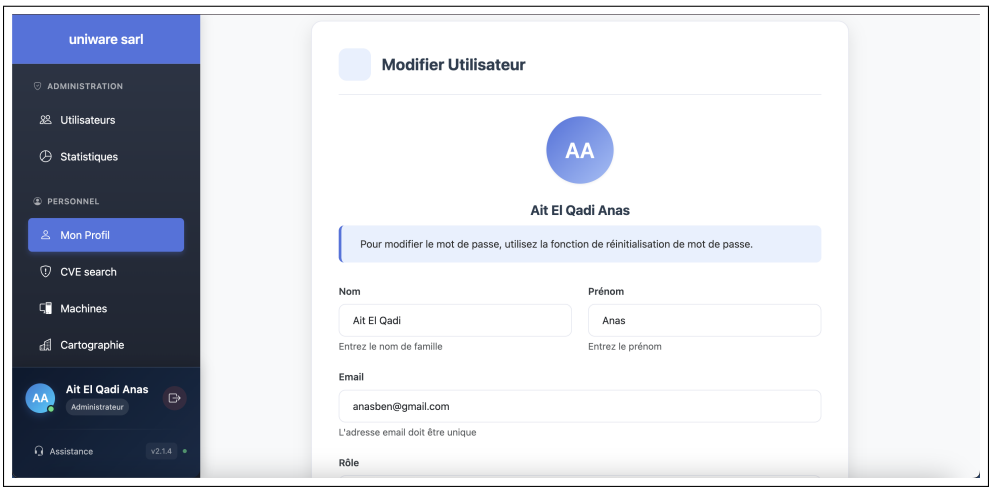


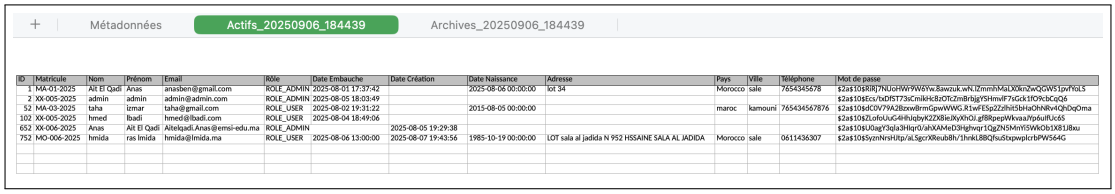
FIGURE 3.13 – Interface de modification d'un utilisateur

3.4.7 Rapports des utilisateurs

Rapport des utilisateurs actifs

Le système génère des rapports détaillés (Figure 3.14) présentant l'ensemble des utilisateurs actifs avec toutes leurs informations :

— Données complètes de chaque utilisateur
— Informations sensibles sécurisées (mots de passe hashés)
— Dates importantes (création, embauche, naissance)
— Données de localisation complètes



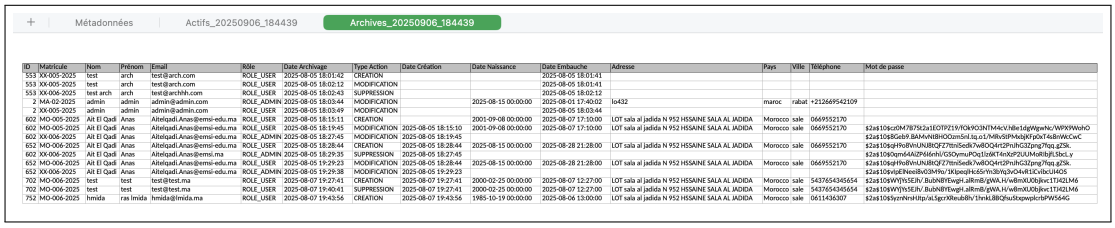
ID	Matricule	Nom	Prénom	Email	Rôle	Date Embauche	Date Création	Date Naissance	Adresse	Pays	Ville	Téléphone	Mot de passe
1	MA-01-2025	Ait El Qadi	Anas	anasben@gmail.com	ROLE_ADMIN	2025-08-01 17:37:42		2025-08-06 00:00:00	lot 34	Morocco	sale	7654345678	$2a$10$RIRj7NUoHWr9W6Yw.8awzuk.wN.IZmmhMaLX0knZwQCWS1pvfYoLS
2	XX-005-2025	admin	admin	admin@admin.com	ROLE_ADMIN	2025-08-05 18:03:49							$2a$10$Ecs/hxDEtT73xCmikHc8zOTcZm8rBjgYSHnvFF7sGckIfORcbCqQ6
52	MA-03-2025	taha	izmar	taha@gmail.com	ROLE_USER	2025-08-02 19:31:22		2015-08-05 00:00:00		maroc	kamouni	765434567876	$2a$10$dC0V79A2Bzxw8mCgwvWWG.R1wFESgZZzfnlSbHaOhNBv4QhDqOma
102	XX-005-2025	hmed	lbadi	hmed@lbadi.com	ROLE_USER	2025-08-04 18:49:06							$2a$10$ZLofoUuG4HhUqbyK2ZX8ieJKyXhOJ.gf8RpepMkvaaJYp6ulfUc65
652	XX-006-2025	Anas	Ait El Qadi	Aitelqadi.Anas@emsi-edu.ma	ROLE_ADMIN		2025-08-05 19:29:38						$2a$10$UOagY3qla3HiqrO/ahXAMeD3HghvqrI0gZN5MnYi5WkOb1X81J8xu
752	MO-006-2025	hmida	ras imida	hmida@imida.ma	ROLE_USER	2025-08-06 13:00:00	2025-08-07 19:43:56	1985-10-19 00:00:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0611436307	$2a$10$SyznNnsHUtp/aLSgcrXRezuBRh/1hrkLBBQfuSbqwpkJcrbPW564G

FIGURE 3.14 – Rapport détaillé des utilisateurs actifs

Rapport des utilisateurs archives

Le système maintient un historique complet des modifications (Figure 3.15) avec archivage des changements apportés aux comptes utilisateurs :

— Historique de toutes les actions (création, modification, suppression)
— Traçabilité complète des évolutions des comptes
— Conservation des anciennes versions des données



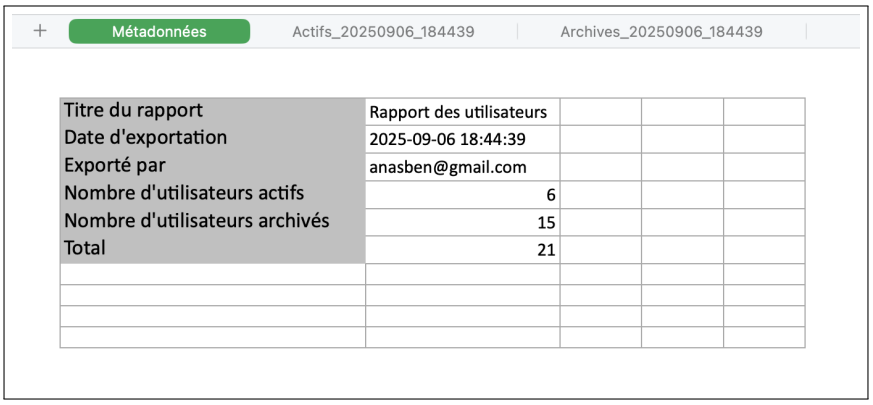
+ | Métadonnées | Actifs_20250906_184439 | Archives_20250906_184439

ID	Matricule	Nom	Prénom	Email	Rôle	Date Archivage	Type Action	Date Création	Date Naissance	Date Embauche	Adresse	Pays	Ville	Téléphone	Mot de passe
553	XX-005-2025	test	arch	test@arch.com	ROLE_USER	2025-08-05 18:01:42	CREATION			2025-08-05 18:01:41					
553	XX-005-2025	test	arch	test@arch.com	ROLE_USER	2025-08-05 18:02:12	MODIFICATION			2025-08-05 18:01:41					
553	XX-006-2025	test arch	arch	test@archhh.com	ROLE_USER	2025-08-05 18:02:43	SUPPRESSION			2025-08-05 18:02:12					
2	MA-02-2025	admin	admin	admin@admin.com	ROLE_ADMIN	2025-08-05 18:03:44	MODIFICATION		2025-08-15 00:00:00	2025-08-01 17:40:02	lo432	maroc	rabat	+212669542109	
2	XX-005-2025	admin	admin	admin@admin.com	ROLE_USER	2025-08-05 18:03:49	MODIFICATION			2025-08-05 18:03:44					
602	MD-005-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_USER	2025-08-05 18:15:11	CREATION		2001-09-08 00:00:00	2025-08-07 17:10:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0669552170	
602	MD-005-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_USER	2025-08-05 18:19:45	MODIFICATION	2025-08-05 18:15:10	2001-09-08 00:00:00	2025-08-07 17:10:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0669552170	$2a$10$cz0M7B75z2a1EOTP219/fOk9O3NTM4cV.Nlle1dgWgwNc/WPKPWohO
602	XX-006-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_ADMIN	2025-08-05 18:27:45	MODIFICATION	2025-08-05 18:19:45							$2a$10$8Geb9.BAMvNtBHOOzmSnLtq.oL/MRv5tPMkbjKFpOxT4sBnWcCwC
652	MD-006-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_USER	2025-08-05 18:28:44	CREATION	2025-08-05 18:28:44	2025-08-15 00:00:00	2025-08-28 21:28:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0669552170	$2a$10$qHRvRHvUNJBIQfZ7ltnSedk7w8OQ4rt2PnJhG3Zpng7Rqq.gZ5k.
602	XX-006-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi.ma	ROLE_ADMIN	2025-08-05 18:29:35	SUPPRESSION	2025-08-05 18:27:45							$2a$10$Oqmd4AIiZP6l6nhl/GSOymuPOq1Izd0KT4nXzP2UUMoRlbjfLSbcL.y
652	MD-006-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_USER	2025-08-05 19:29:23	MODIFICATION	2025-08-05 18:28:44	2025-08-15 00:00:00	2025-08-28 21:28:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0669552170	$2a$10$qHRvRHvUNJBIQfZ7ltnSedk7w8OQ4rt2PnJhG3Zpng7Rqq.gZ5k.
652	XX-006-2025	Ait El Qadi	Anas	Aitelqadi.Anas@emsi-edu.ma	ROLE_ADMIN	2025-08-05 19:29:38	MODIFICATION	2025-08-05 19:29:23							$2a$10$VjdENeeiBvO3HNu/.WJepqHc65rVn3B/6jSvO4eR1ICvIbcUHO5
702	MD-006-2025	test	test	test@test.ma	ROLE_USER	2025-08-07 19:27:41	CREATION	2025-08-07 19:27:41	2000-02-25 00:00:00	2025-08-07 12:27:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	5437654345654	$2a$10$WYjYk5EJh/.BubNBYEwgH.aIRm8/gWA.H/w8mXU0bjkvc1TJ42LM6
702	MD-006-2025	test	test	test@test.ma	ROLE_USER	2025-08-07 19:40:41	SUPPRESSION	2025-08-07 19:27:41	2000-02-25 00:00:00	2025-08-07 12:27:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	5437654345654	$2a$10$WYjYk5EJh/.BubNBYEwgH.aIRm8/gWA.H/w8mXU0bjkvc1TJ42LM6
752	MD-006-2025	hmida	ras lmida	hmida@imida.ma	ROLE_USER	2025-08-07 19:43:56	CREATION	2025-08-07 19:43:56	1985-10-19 00:00:00	2025-08-06 13:00:00	LOT sala al jadida N 952 HSSAINE SALA AL JADIDA	Morocco	sale	0611436307	$2a$10$SynHrnHitp/aLSgrrXReubBh/1hnkLBlQfsuSxpwpkcbPVS64G

FIGURE 3.15 – Archives et historique des modifications des utilisateurs

Métadonnées des rapports utilisateurs

Chaque rapport généré inclut des métadonnées complètes (Figure 3.16) permettant d'identifier son contexte de création :
— Date et heure de génération du rapport
— Identité de l'utilisateur ayant généré le rapport
— Statistiques sur le nombre d'utilisateurs (actifs, archivés, total)



+ | Métadonnées | Actifs_20250906_184439 | Archives_20250906_184439

Titre du rapport	Rapport des utilisateurs
Date d'exportation	2025-09-06 18:44:39
Exporté par	anasben@gmail.com
Nombre d'utilisateurs actifs	6
Nombre d'utilisateurs archivés	15
Total	21

FIGURE 3.16 – Métadonnées des rapports utilisateurs

3.4.8 Formulaire d'ajout de machine

L'interface d'ajout de nouvelles machines (Figure 3.17) permet aux administrateurs d'enrichir l'inventaire du système d'information avec tous les détails techniques nécessaires :
— Informations d'identification (nom, type)
— Caractéristiques techniques (fabricant, modèle, numéro de série)
— Paramètres réseau (adresse IP, adresse MAC)
— Spécifications du système d'exploitation

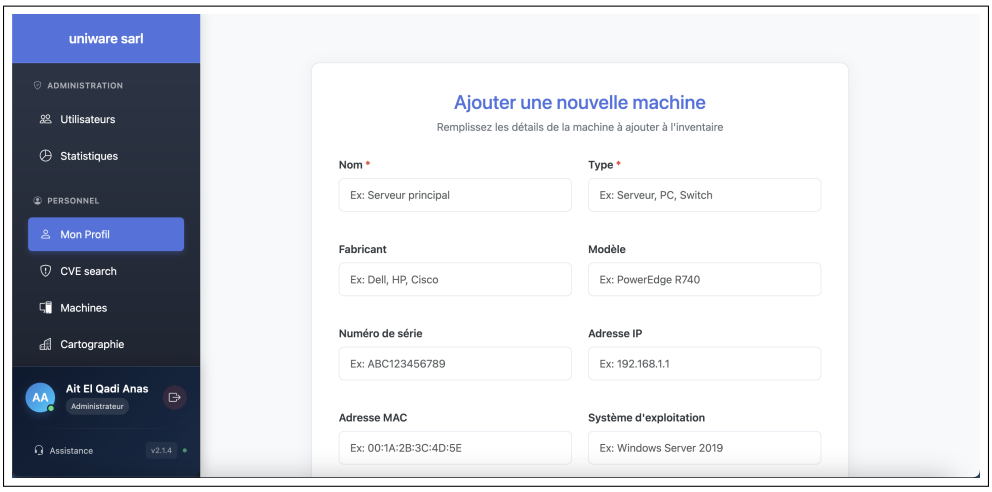


FIGURE 3.17 – Formulaire d'ajout d'une nouvelle machine à l'inventaire

3.4.9 Liste des machines

L'interface de listing des machines (Figure 3.18) présente l'ensemble des équipements du système d'information avec leurs caractéristiques principales et permet diverses actions :

— Visualisation tabulaire des machines avec leurs propriétés essentielles
— Options d'export de rapports
— Filtrage des machines vulnérables
— Lancement de scans du réseau

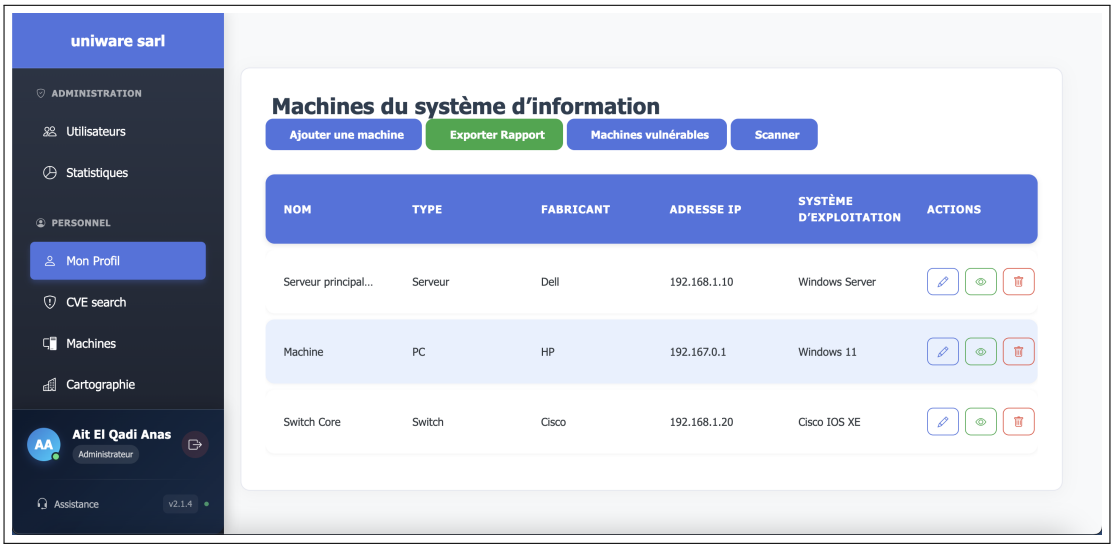


FIGURE 3.18 – Liste des machines du système d'information

3.4.10 Détails d'une machine

L'application offre une vue détaillée pour chaque machine (Figures 3.19 et 3.20), organisée en plusieurs sections :

— Identité complète de la machine (type, fabricant, modèle, numéro de série)

— Configuration réseau (adresse IP, adresse MAC)
— Caractéristiques matérielles (processeur, mémoire RAM, stockage)
— Informations de localisation physique
— Dates importantes (achat, fin de garantie)
— Description et notes supplémentaires

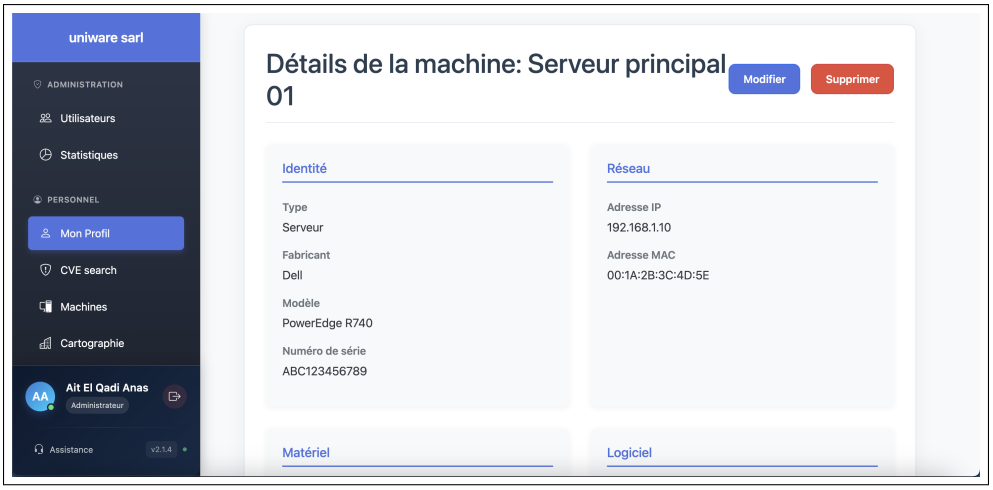


FIGURE 3.19 – Section supérieure des détails d'une machine

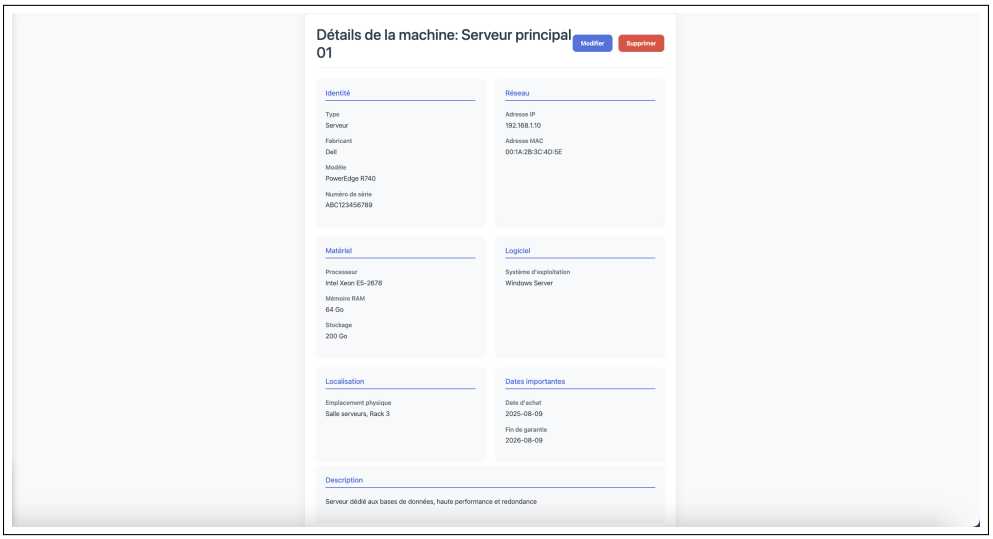


FIGURE 3.20 – Section inférieure des détails d'une machine

3.4.11 Édition d'une machine

L'interface d'édition (Figure 3.21) permet la modification des propriétés d'une machine existante, avec conservation de l'historique des changements pour assurer la traçabilité.

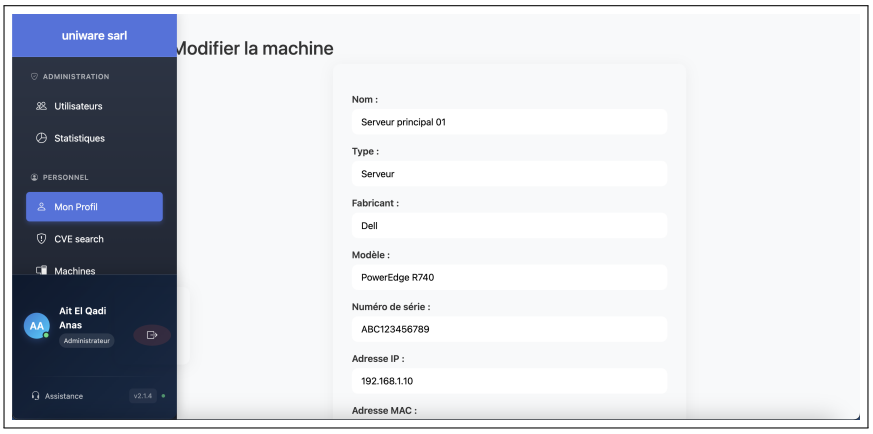


FIGURE 3.21 – Interface de modification d'une machine

3.4.12 Détection des vulnérabilités

Le module de détection des vulnérabilités (Figures 3.22 et 3.23) identifie les failles de sécurité présentes sur les machines analysées :

- Liste des machines vulnérables avec leurs caractéristiques
- Indication de la cause de la vulnérabilité (OS, logiciel, etc.)
- Score CVSS pour prioriser les corrections
- Description détaillée des CVE identifiées

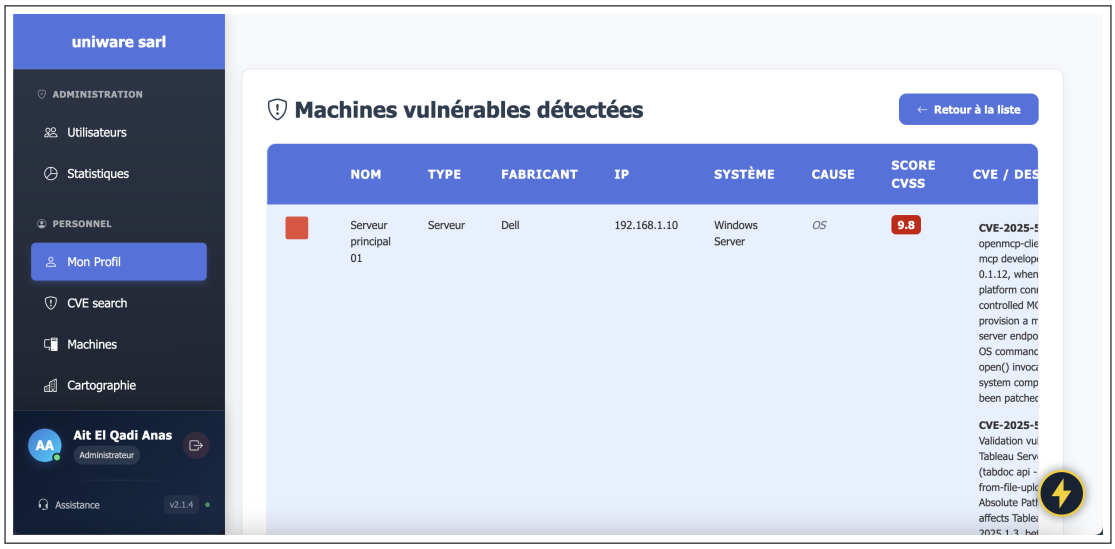
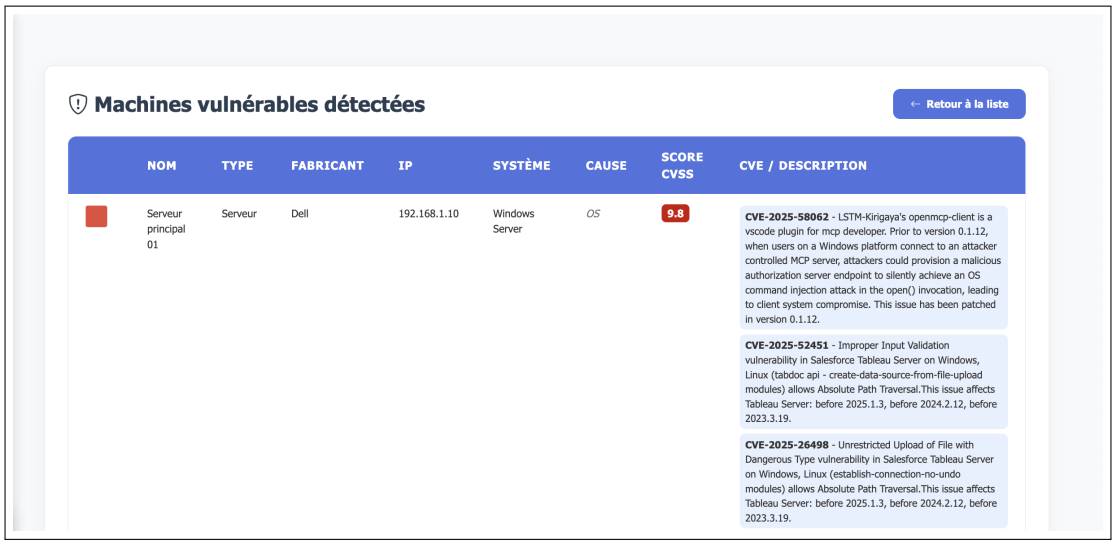


FIGURE 3.22 – Tableau des machines vulnérables détectées (partie 1)



Machines vulnérables détectées

← Retour à la liste

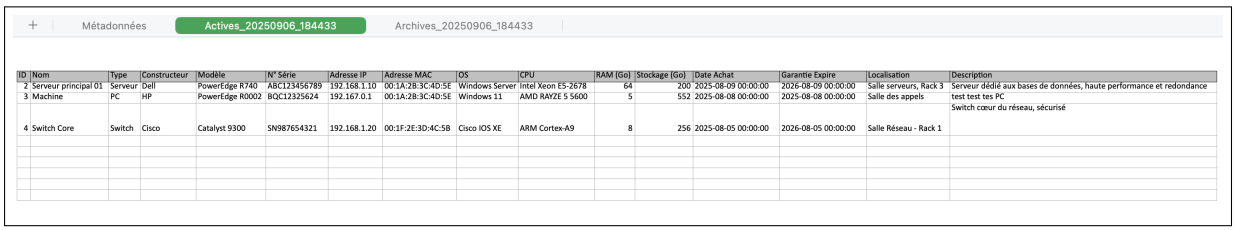
NOM	TYPE	FABRICANT	IP	SYSTÈME	CAUSE	SCORE CVSS	CVE / DESCRIPTION
Serveur principal 01	Serveur	Dell	192.168.1.10	Windows Server	*OS*	9.8	**CVE-2025-58062** - LSTM-Kirigaya's openmcp-client is a vscode plugin for mcp developer. Prior to version 0.1.12, when users on a Windows platform connect to an attacker controlled MCP server, attackers could provision a malicious authorization server endpoint to silently achieve an OS command injection attack in the open() invocation, leading to client system compromise. This issue has been patched in version 0.1.12. **CVE-2025-52451** - Improper Input Validation vulnerability in Salesforce Tableau Server on Windows, Linux (tabdoc api - create-data-source-from-file-upload modules) allows Absolute Path Traversal.This issue affects Tableau Server: before 2025.1.3, before 2024.2.12, before 2023.3.19. **CVE-2025-26498** - Unrestricted Upload of File with Dangerous Type vulnerability in Salesforce Tableau Server on Windows, Linux (establish-connection-no-undo modules) allows Absolute Path Traversal.This issue affects Tableau Server: before 2025.1.3, before 2024.2.12, before 2023.3.19.

FIGURE 3.23 – Tableau des machines vulnérables détectées (partie 2)

3.4.13 Génération de rapports machines

Rapport des machines actives

L'application génère des rapports détaillés (Figure 3.24) présentant l'ensemble des machines actives dans le système d'information avec toutes leurs caractéristiques techniques.



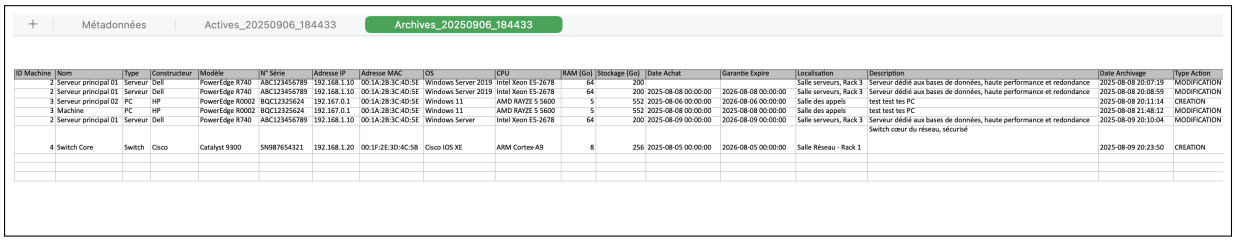
+ | Métadonnées | Actives_20250906_184433 | Archives_20250906_184433

ID	Nom	Type	Constructeur	Modèle	N° Série	Adresse IP	Adresse MAC	OS	CPU	RAM (Go)	Stockage (Go)	Date Achat	Garantie Expire	Localisation	Description
2	Serveur principal 01	Serveur	Dell	PowerEdge R740	ABC123456789	192.168.1.10	00:1A:2B:3C:4D:5E	Windows Server	Intel Xeon E5-2678	64	200	2025-08-09 00:00:00	2026-08-09 00:00:00	Salle serveurs, Rack 3	Serveur dédié aux bases de données, haute performance et redondance
3	Machine	PC	HP	PowerEdge R0002	BQC12325624	192.167.0.1	00:1A:2B:3C:4D:5E	Windows 11	AMD RAYZE 5 5600	5	552	2025-08-08 00:00:00	2025-08-08 00:00:00	Salle des appels	test test tes PC
4	Switch Core	Switch	Cisco	Catalyst 9300	SN987654321	192.168.1.20	00:1F:2E:3D:4C:5B	Cisco IOS XE	ARM Cortex-A9	8	256	2025-08-05 00:00:00	2026-08-05 00:00:00	Salle Réseau - Rack 1	Switch cœur du réseau, sécurisé

FIGURE 3.24 – Rapport détaillé des machines actives

Rapport des machines archives

Le système maintient un historique complet des modifications (Figure 3.25) avec archivage des changements apportés aux machines, permettant une traçabilité complète des évolutions du parc.



+ | Métadonnées | Actives_20250906_184433 | Archives_20250906_184433

ID Machine	Nom	Type	Constructeur	Modèle	N° Série	Adresse IP	Adresse MAC	OS	CPU	RAM (Go)	Stockage (Go)	Date Achat	Garantie Expire	Localisation	Description	Date Archivage	Type Action
2	Serveur principal 01	Serveur	Dell	PowerEdge R740	ABC123456789	192.168.1.10	00:1A:2B:3C:4D:5E	Windows Server 2019	Intel Xeon E5-2678	64	200			Salle serveurs, Rack 3	Serveur dédié aux bases de données, haute performance et redondance	2025-08-08 20:07:19	MODIFICATION
2	Serveur principal 01	Serveur	Dell	PowerEdge R740	ABC123456789	192.168.1.10	00:1A:2B:3C:4D:5E	Windows Server 2019	Intel Xeon E5-2678	64	200	2025-08-08 00:00:00	2026-08-08 00:00:00	Salle serveurs, Rack 3	Serveur dédié aux bases de données, haute performance et redondance	2025-08-08 20:08:59	MODIFICATION
3	Serveur principal 02	PC	HP	PowerEdge R0002	BQC12325624	192.167.0.1	00:1A:2B:3C:4D:5E	Windows 11	AMD RAYZE 5 5600	5	552	2025-08-06 00:00:00	2026-08-06 00:00:00	Salle des appels	test test tes PC	2025-08-08 20:11:14	CREATION
3	Machine	PC	HP	PowerEdge R0002	BQC12325624	192.167.0.1	00:1A:2B:3C:4D:5E	Windows 11	AMD RAYZE 5 5600	5	552	2025-08-08 00:00:00	2025-08-08 00:00:00	Salle des appels	test test tes PC	2025-08-08 21:48:12	MODIFICATION
2	Serveur principal 01	Serveur	Dell	PowerEdge R740	ABC123456789	192.168.1.10	00:1A:2B:3C:4D:5E	Windows Server	Intel Xeon E5-2678	64	200	2025-08-09 00:00:00	2026-08-09 00:00:00	Salle serveurs, Rack 3	Serveur dédié aux bases de données, haute performance et redondance	2025-08-09 20:10:04	MODIFICATION
4	Switch Core	Switch	Cisco	Catalyst 9300	SN987654321	192.168.1.20	00:1F:2E:3D:4C:5B	Cisco IOS XE	ARM Cortex-A9	8	256	2025-08-05 00:00:00	2026-08-05 00:00:00	Salle Réseau - Rack 1	Switch cœur du réseau, sécurisé	2025-08-09 20:23:50	CREATION

FIGURE 3.25 – Archives et historique des modifications des machines

Métadonnées des rapports machines

Chaque rapport généré inclut des métadonnées complètes (Figure 3.26) permettant d'identifier son contexte de création et son contenu.

+ Métadonnées | Actives_20250906_184433 | Archives_20250906_184433

Titre du rapport	Rapport des machines
Date d'exportation	2025-09-06 18:44:33
Exporté par	anasben@gmail.com
Nombre de machines actives	3
Nombre de machines archivées	6
Total	9

FIGURE 3.26 – Métadonnées des rapports machines

3.4.14 Scan du réseau (Version bêta)

Interface de scan réseau

Machines détectées sur le réseau

ID	IP	MAC	OS	Manufacturer
1	127.0.0.1	UNKNOWN	UNKNOWN	UNKNOWN
2	192.168.1.1	28:77:77:9C:68:34	UNKNOWN	UNKNOWN
3	192.168.1.7	UNKNOWN	UNKNOWN	UNKNOWN
4	192.168.1.107	BC:32:5F:8E:57:8E	UNKNOWN	UNKNOWN
5	192.168.1.3	AA:13:75:F8:4C:48	UNKNOWN	UNKNOWN
6	192.168.1.4	60:70:72:5B:1B:77	UNKNOWN	UNKNOWN
7	192.168.1.8	UNKNOWN	UNKNOWN	UNKNOWN
8	192.168.1.11	UNKNOWN	UNKNOWN	UNKNOWN
9	192.168.1.17	56:F4:DA:82:5B:3D	UNKNOWN	UNKNOWN
10	192.168.1.9	UNKNOWN	UNKNOWN	UNKNOWN
11	192.168.1.10	UNKNOWN	UNKNOWN	UNKNOWN
12	192.168.1.2	7E:1E:B1:A4:8A:AF	UNKNOWN	UNKNOWN
13	192.168.1.5	UNKNOWN	UNKNOWN	UNKNOWN
14	192.168.1.6	AE:4E:F0:36:1B:DB	UNKNOWN	UNKNOWN
15	192.168.1.18	64:48:42:8B:EF:1C	UNKNOWN	UNKNOWN
16	192.168.1.20	6E:66:85:AD:80:D9	UNKNOWN	UNKNOWN
17	192.168.1.30	5A:B2:A1:7A:B7:C0	UNKNOWN	UNKNOWN
18	192.168.1.31	12:F7:F1:BC:36:C0	UNKNOWN	UNKNOWN
19	192.168.1.69	60:70:72:5B:1B:77	UNKNOWN	UNKNOWN

FIGURE 3.27 – Interface de scan réseau - Liste des machines détectées

L'interface de scan réseau (Figure 3.27) permet aux administrateurs de détecter automatiquement les machines connectées au réseau local. Cette fonctionnalité, actuellement en version bêta, offre :

— Détection automatique des adresses IP actives sur le réseau
— Identification des adresses MAC des équipements
— Liste complète des machines détectées avec leurs caractéristiques
— Intégration automatique des nouvelles machines dans l'inventaire

Fonctionnement backend du scan

Le module de scan réseau utilise des requêtes Hibernate pour interagir avec la base de données (Figures 3.28 et 3.29) :
— Vérification de l'existence des machines dans la base de données
— Insertion automatique des nouvelles machines détectées
— Mise à jour des informations techniques (adresses MAC, fabricants)
— Gestion des doublons et des conflits d'adresses IP

```
Scan réseau : 192.168.1.0/24
Hibernate:
    select
        ms1_0.id,
        ms1_0.ip_address,
        ms1_0.mac_address,
        ms1_0.manufacturer,
        ms1_0.operating_system
    from
        machine_scan ms1_0
    where
        ms1_0.ip_address=?
```

FIGURE 3.28 – Backend du scan réseau - Requêtes de vérification

```
Machine détectée : 192.168.1.1 / 28:77:77:9C:68:34
Hibernate:
    select
        ms1_0.id,
        ms1_0.ip_address,
        ms1_0.mac_address,
        ms1_0.manufacturer,
        ms1_0.operating_system
    from
        machine_scan ms1_0
    where
        ms1_0.ip_address=?
Hibernate:
    insert
    into
        machine_scan
        (ip_address, mac_address, manufacturer, operating_system)
    values
        (?, ?, ?, ?)
```

FIGURE 3.29 – Backend du scan réseau - Insertion des nouvelles machines

3.4.15 Cartographie du réseau (Version bêta)

L'interface de cartographie (Figure 3.30) offre une visualisation graphique du réseau d'entreprise, actuellement en version bêta :

— Représentation visuelle de l'infrastructure réseau
— Organisation des équipements par étages ou zones géographiques
— Visualisation des relations entre les différents composants
— Accès rapide aux détails des machines via leur représentation graphique

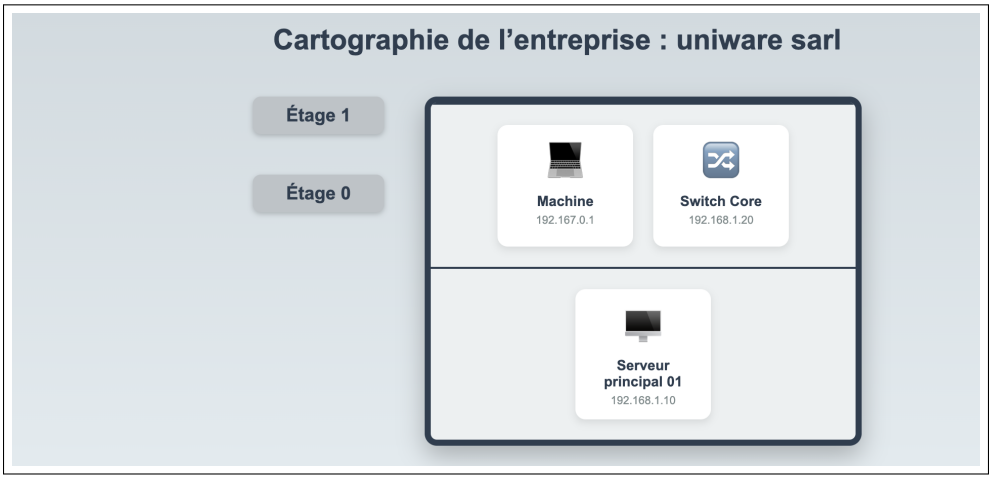


FIGURE 3.30 – Interface de cartographie du réseau d'entreprise

Note : Les fonctionnalités de scan réseau et de cartographie sont actuellement en version bêta et font l'objet de tests intensifs. Leur intégration complète est prévue pour la prochaine version majeure de l'application.

3.4.16 Fonctionnalités implémentées

Gestion des utilisateurs et authentification

Le système d'authentification a été développé avec Spring Security, offrant :

— Une authentification sécurisée par JWT (JSON Web Token)
— Une gestion des rôles (Administrateur/Utilisateur) avec permissions différenciées
— Un mécanisme de hachage des mots de passe (BCrypt)
— Une expiration automatique des sessions pour renforcer la sécurité

Gestion des composants

Le module de gestion des composants permet :

— L'ajout manuel ou automatique (via import) des composants matériels et logiciels
— La consultation détaillée de chaque composant avec ses caractéristiques techniques
— L'archivage et la traçabilité des modifications apportées aux composants

Détection des vulnérabilités

Le cœur de l'application repose sur l'intégration avec l'API NVD (National Vulnerability Database) :

— Connexion régulière à l'API pour récupérer les dernières CVE
— Analyse des composants enregistrés contre la base de vulnérabilités
— Attribution d'un statut de sécurité pour chaque composant (vulnérable, à mettre à jour, à jour)
— Calcul automatique des scores CVSS pour prioriser les actions

Génération de rapports

Le module de reporting propose :
— L'export des données au format PDF (via ReportLab) et Excel (via Apache POI)
— La génération automatique de rapports d'audit périodiques
— Un historique des rapports générés avec archivage sécurisé
— Des notifications par email pour les vulnérabilités de haute criticité

3.4.17 Tableau de bord et visualisation

L'application inclut un tableau de bord offrant une vue d'ensemble de la sécurité du système d'information :
— Des indicateurs visuels (KPI) sur l'état global de sécurité
— Des graphiques statistiques sur l'évolution des vulnérabilités
— Une cartographie des composants et leurs relations
— Un suivi des périodes d'attaque et des mesures correctives appliquées

3.4.18 Déploiement et utilisation

L'application a été déployée selon une approche DevOps avec les caractéristiques suivantes :
— Configuration via le fichier `application.properties` de Spring Boot
— Scripts Maven pour l'automatisation des builds
— Base de données MySQL avec sauvegardes automatiques
— Journalisation (logging) complète pour le débogage et l'audit

L'accès à l'application se fait via l'URL `http://localhost:8080/login` après démarrage du serveur Spring Boot.

3.5 Conclusion

Pour conclure cette section consacrée à l'étude pratique de notre projet, nous avons minutieusement analysé les outils de développement utilisés et la séquence des interfaces. Nos éléments sont essentiels afin de garantir la solidité technique et l'ergonomie de notre application . Le choix méticuleux des outils et la planification minutieuse des interfaces représentent deux étapes cruciales pour réaliser notre vision.

Conclusion générale

En conclusion, le projet Security Component Checker représente une contribution significative à l'amélioration de la cybersécurité dans le cadre de la gestion des systèmes d'information. Il a permis de concevoir et de mettre en œuvre une application web moderne, capable de vérifier l'état des composants logiciels et matériels, d'identifier les vulnérabilités connues et de centraliser l'ensemble des informations critiques liées à la sécurité.

Cette solution répond à un besoin croissant des entreprises : disposer d'un outil automatisé permettant de réduire la dépendance aux vérifications manuelles souvent longues, coûteuses et sujettes aux erreurs humaines. Grâce à l'utilisation des technologies Java Spring Boot, Spring MVC et Thymeleaf, l'application repose sur une architecture modulaire et robuste, garantissant maintenabilité et évolutivité. L'intégration de l'API NVD pour l'extraction des vulnérabilités (CVE), couplée à l'utilisation d'outils externes tels que Nmap, a permis de renforcer la pertinence et la fiabilité des analyses. Les fonctionnalités développées – notamment la gestion des utilisateurs avec rôles et traçabilité, la détection des vulnérabilités, la génération de rapports automatisés (PDF/Excel) et la mise en place d'un tableau de bord interactif – contribuent à faire de cette application un véritable outil d'aide à la décision pour les administrateurs et responsables sécurité.

Ce projet a également été l'occasion de relever des défis techniques et organisationnels, tels que la sécurisation des échanges via JWT et hashage de mots de passe, la conformité aux normes de protection des données (RGPD, CVSS), ainsi que l'adoption d'une méthode agile basée sur des sprints successifs permettant un suivi rigoureux de l'avancement. Au-delà des aspects purement techniques, ce travail a consolidé nos compétences en développement web full stack, en intégration d'APIs de cybersécurité, en architecture logicielle orientée sécurité et en gestion de projet agile. Ainsi, l'application Security Component Checker constitue une base solide qui apporte une réelle valeur ajoutée aux organisations, en renforçant leur capacité de veille proactive et en optimisant la gestion de leurs risques liés aux vulnérabilités.

Perspectives

Bien que le projet ait atteint ses objectifs principaux, plusieurs perspectives d'évolution méritent d'être envisagées pour renforcer l'efficacité et l'impact de la solution.

Un premier axe majeur consiste à développer une cartographie dynamique et interactive du système d'information, permettant de représenter visuellement les machines, serveurs, réseaux et applications, ainsi que leurs interconnexions. Une telle fonctionnalité faciliterait la détection des points sensibles, l'analyse des dépendances entre services et l'identification des zones à haut risque.

En complément, l'implémentation d'un modèle d'intelligence artificielle constitue une évolution stratégique : celui-ci permettrait d'améliorer la détection automatique des machines et composants grâce à des algorithmes de classification et de reconnaissance basés sur les données historiques collectées. Cette approche ouvrirait la voie à une analyse prédictive des vulnérabilités, capable d'anticiper l'émergence de nouvelles menaces et de recommander des correctifs prioritaires selon leur criticité et leur impact métier.

Un autre axe de développement concerne l'extension fonctionnelle et technique de l'application : la mise en place d'une version mobile garantirait un accès simplifié et sécurisé aux fonctionnalités essentielles, favorisant une supervision en temps réel quel que soit l'endroit où se trouvent les administrateurs. De plus, l'intégration avec des solutions de SIEM (Security Information and Event Management) ou de SOC (Security Operations Center) permettrait de connecter l'application à des infrastructures de surveillance plus globales, offrant ainsi une couverture de sécurité renforcée et une meilleure gestion des incidents.

Enfin, l'amélioration continue en matière de conformité réglementaire représente une perspective incontournable : la génération de rapports normalisés alignés sur des standards tels que ISO 27001, NIST ou CIS Controls renforcerait la crédibilité de l'application et faciliterait son adoption dans des contextes professionnels exigeants.

À long terme, ces évolutions permettront de transformer le Security Component Checker en une plateforme intelligente et intégrée de cybersécurité, alliant diagnostic, prédiction et visualisation, et répondant ainsi de manière proactive aux défis actuels et futurs de la protection des systèmes d'information.

Webographie

1. Pour STARUML : https://staruml.io/user-guide
2. Pour springboot : https://docs.spring.io/spring-boot/index.html
3. Pour bootstrap : https://getbootstrap.com/docs/5.3/getting-started/introduction/
4. Pour l'éditeur intellij idea ultimate : https://www.jetbrains.com/idea/download
5. Pour w3schools (guide por HTML/CSS/JS) : https://www.w3schools.com/
6. Pour CVE : https://www.cve.org/
7. Pour API NVD : https://nvd.nist.gov/developers

Bibliographie

Pour la réalisation de ce projet, un ensemble complet de ressources documentaires a été mobilisé afin de garantir une maîtrise approfondie des technologies utilisées, notamment Java et le framework Spring Boot. Concernant Java, plusieurs ouvrages de référence ont été essentiels. Effective Java de Joshua Bloch (Addison-Wesley, 2018) a permis de comprendre et d'appliquer les meilleures pratiques de programmation orientée objet, d'optimiser la conception des API, la gestion de la mémoire et le traitement des exceptions. Head First Java de Kathy Sierra et Bert Bates (O'Reilly, 2008), avec son approche pédagogique visuelle et interactive, a facilité la consolidation des bases du langage et l'assimilation des concepts fondamentaux de la programmation orientée objet. Pour approfondir la compréhension du langage et de ses mécanismes avancés, Thinking in Java de Bruce Eckel (Prentice Hall, 2006) a fourni une vision complète de la philosophie Java, couvrant des aspects tels que les interfaces, les collections, la gestion des exceptions et les modèles de conception. La documentation officielle Java SE d'Oracle (https ://docs.oracle.com/javase/) a complété ces ouvrages en apportant des informations précises sur les classes, interfaces, packages et fonctionnalités du langage, tandis que les conventions de codage Java d'Oracle (https ://www.oracle.com/java/technologies/javase/codeconventions-contents.html) ont été suivies afin d'assurer la cohérence, la lisibilité et le respect des standards industriels.

En ce qui concerne Spring Boot, Spring in Action de Craig Walls (Manning, 2018) a servi de guide fondamental pour comprendre l'architecture modulaire de Spring, la configuration automatique, l'injection de dépendance, ainsi que l'utilisation et l'intégration des modules tels que Spring Data JPA, Spring Security et Spring MVC. L'ouvrage Clean Code de Robert C. Martin (Prentice Hall, 2008) a renforcé l'application de pratiques de code maintenable et structuré. La documentation officielle de Spring Boot (https ://docs.spring.io/spring-boot/docs/current/reference/html/) a été consultée pour approfondir la compréhension des options de configuration, des annotations et des bonnes pratiques de développement.

Pour renforcer les connaissances pratiques, plusieurs sites web spécialisés ont été utilisés. Baeldung (https ://www.baeldung.com/) a fourni des articles détaillés et des cas d'usage concrets autour de Spring, Spring Boot, JPA et de la sécurité web. GeeksforGeeks (https ://www.geeksforgeeks.org/java/) a permis de compléter la formation grâce à des explications claires et des exemples pratiques. OpenClassrooms a fourni des tutoriels détaillés pour la création d'applications web avec Spring Boot, de la configuration initiale à l'implémentation des fonctionnalités avancées.

Enfin, pour la sécurité des composants logiciels et le suivi des vulnérabilités, des ressources documentaires détaillées sur les API NVD (National Vulnerability Database, https ://nvd.nist.gov/developers/vulnerabilities) et CVE (Common Vulnerabilities and Exposures, https ://cveawg.mitre.org/api-docs/) ont été mobilisées. La NVD, gérée par le National Institute of Standards and Technology (NIST), fournit une base de données

centralisée et structurée sur toutes les vulnérabilités publiées, incluant des informations sur le type de vulnérabilité, sa gravité via le score CVSS (Common Vulnerability Scoring System), les produits affectés et les références associées. L'API NVD permet d'accéder automatiquement à ces données au format JSON, facilitant leur intégration dans des systèmes de vérification et de suivi automatisés. Le système CVE, maintenu par MITRE, attribue un identifiant unique à chaque vulnérabilité connue, ce qui permet un suivi normalisé et reconnu mondialement. L'utilisation combinée des API NVD et CVE dans le cadre du projet a permis de vérifier la sécurité des dépendances logicielles et des composants tiers, d'identifier les vulnérabilités critiques, et d'automatiser la collecte et l'analyse des données de sécurité. Des outils tiers tels que NVDLib et OpenCVE ont été exploités pour faciliter l'extraction, l'analyse et la visualisation des vulnérabilités, permettant de mettre en place un suivi régulier et proactif des composants logiciels utilisés, contribuant ainsi à la robustesse et à la sécurité globale de l'application développée.