RÉPUBLIQUE DU SÉNÉGAL
Un Peuple – Un But – Une Foi



MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR, DE LA RECHERCHE ET DE L'INNOVATION



UNIVERSITE CHEIKH ANTA DIOP DE DAKAR

****

FACULTÉ DES SCIENCES ET TECHNIQUES

****

DEPARTEMENT DE MATHEMATIQUES ET INFORMATIQUE

SECTION INFORMATIQUE

MÉMOIRE DE MASTER INFORMATIQUE
Spécialité : Réseaux et Télécommunications
Option : Réseaux, Systèmes et Services

Sujet : Étude des vulnérabilités et Proposition des mécanismes de protection pour l'application web Easy Compta de Volkeno

Présenté et soutenu publiquement par :

Mme Fatou Bintou DIALLO

Encadré par Professeur Chérif DIALLO

Le : 13 Mai 2025

Jury

Pr. Ibrahima NIANG	Professeur Titulaire, UCAD	Président
Dr. Mohamed Ould DEYE	Maître Assistant Titulaire, UCAD	Membre
Dr. Ibrahima DIANE	Maître Assistant Titulaire, UCAD	Membre
Dr. Ousmane SADIO	Maître Assistant Titulaire, UCAD	Membre

Année universitaire 2023 – 2024

DÉDICACES

Je dédie ce mémoire :

A mes défunts parents, pour leur amour inconditionnel, pour tous leurs efforts et sacrifices. J'aurais aimé que ce moment leur trouve encore parmi nous mais Allah en a décidé ainsi, que leur ame repose en paix et que firdawsi soit leur demeure éternelle amiine;

A mes oncles et mes tantes, vous avez toujours été présents pour moi, en m'encourageant dans mes études. A mes frères, sœurs, cousins, pour votre amour, vos encouragements et votre implication. Puisse Allah Le Tout Puissant continuer à bénir la famille, puisse-t-il vous garder à ses côtés et vous garder dans toutes vos entreprises. Je vous aime tous;

A ma promotion de l'élémentaire jusqu'au Master (**Mansata Camara, Nafy Mané, Khoury Gueye Ba, Babacar Gueye, Oumar Gueye Kane, Aliou Sylla, Dieynaba Sarr, Ndeye Anta Sarr, Amissatou Thiam, Mamadou Sène et Mbaye Thiaw**), je vous souhaite le bonheur; la santé; la prospérité et qu'Allah vous assiste dans toutes vos épreuves;

À mes professeurs du lycée jusqu'à l'université, pour leur dévouement et leur transmission du savoir;

À tous les enseignants qui ont contribué à mon éducation depuis mon plus jeune âge, je vous en suis infiniment reconnaissante.

REMERCIEMENTS

Je remercie le Tout Puissant, le Tout Miséricordieux, qui m'a donné grâce et bénédiction pour mener à bout ce projet.

Je tiens bien sûr à remercier Mr Moussa Alkaly BADJI pour m'avoir encadré durant ces trois mois. Je le remercie aussi pour sa confiance, son soutien et ses précieux conseils qui m'ont permis de m'initier au monde fabuleux de la cybersécurité.
Je profite de cette occasion pour adresser mes sincères remerciements à toute l'équipe de la RED (Recherche, Education et Développement) de Volkeno pour leur accueil chaleureux et leur soutien constant.Leur esprit d'équipe, ainsi que leur générosité dans le partage de leurs connaissances, ont été pour moi des sources d'apprentissage particulièrement enrichissantes.

Je tiens à exprimer ma profonde gratitude à mon encadrant, Professeur Chérif DIALLO, pour son accompagnement et ses conseils précieux. Un remerciement particulier au Professeur Karim KONATÉ, dont les conseils éclairés, l'expertise et la disponibilité ont grandement contribué à la réussite de mon travail. Sa précieuse guidance a été un facteur déterminant dans l'aboutissement de ce projet.
Je remercie également toute l'équipe pédagogique de la Faculté des Sciences et Techniques pour la qualité de l'enseignement et l'accompagnement offert tout au long de mon parcours académique.

Je voudrais exprimer ma profonde gratitude à ma famille, pour leur soutien indéfectible, tant sur le plan moral que matériel, ainsi que pour leurs encouragements et leurs précieux conseils tout au long de cette aventure. Mes remerciements les plus sincères vont également à Monsieur Sidy Mouhammed AIDARA, pour sa disponibilité, ses conseils éclairés et son accompagnement constant durant la réalisation de ce travail.
Je tiens également à exprimer ma reconnaissance aux membres du jury, pour le temps qu'ils ont consacré à l'évaluation de mon travail, ainsi que pour leurs remarques et conseils avisés.

Enfin, je remercie mes amis de promotion, les étudiants du Master 2 BI, SIR et RETEL pour leur soutien moral et intellectuel tout au long de cette période. Un grand merci à toutes les personnes qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Résumé

Avec la digitalisation croissante des services, la sécurité des applications web est devenue une priorité pour les entreprises. Ce mémoire, réalisé dans le cadre d'un stage à Volkeno, étudie les vulnérabilités de l'application **Easy Compta** et propose des mécanismes de protection adaptés.Exposée à diverses menaces répertoriées dans l' **OWASP Top 10** (injection SQL, XSS, failles d'authentification), l'application a été analysée à travers des **tests d'intrusion** et l'utilisation d' **outils spécialisés** . Les résultats ont permis d'implémenter des solutions de sécurisation telles que le **renforcement de l'authentification, la validation des entrées utilisateur et l'intégration de mécanismes de surveillance** .

Ce travail souligne l'importance d'une approche **proactive en cybersécurité** et de l'adoption de bonnes pratiques pour renforcer la sécurité des applications web.

Mots-clés : Sécurité applicative, tests d'intrusion, OWASP Top 10, cybersécurité, vulnérabilités.

Abstract

With the increasing digitalization of services, web application security has become a priority for businesses. This dissertation, produced as part of an internship at Volcanic, studies the vulnerabilities of the application **Easy Account** and offers appropriate protection mechanisms. Exposed to various threats listed in the **OWASP Top 10** (SQL injection, XSS, authentication flaws), the application was analyzed using **tests d'intrusion** and the use of **specialized tools** . The results made it possible to implement security solutions such as **strengthening authentication, validation of user inputs and integration of monitoring mechanisms** .

This work highlights the importance of an approach **proactive in cybersecurity** and the adoption of best practices to strengthen the security of web applications.

Keywords : Application security, penetration testing, OWASP Top 10, cybersecurity, vulnerabilities.

Table des matières

DÉDICACES 1
REMERCIEMENTS 2
Résumé 3
Abstract 4
Liste des figures 11
Liste des tableaux 11
Introduction Générale 12
CHAPITRE 1 : CADRAGE DE L'ÉTUDE 14
1.1 Introduction 14
1.2 Contexte 14
1.3 Problématique 15
1.4 Objectifs du mémoire 16
1.5 Importance et intérêts du sujet 16
1.6 Méthodologie 18
1.7 Conclusion 19
CHAPITRE 2: ETAT DES LIEUX 21
2.1 Introduction 21
2.2 Présentation de l'entreprise 21
2.2.1 Objectifs et Mission de l'Entreprise 21
2.2.2 Enjeux stratégiques 22
2.3 Caractéristiques de l'application Easy Compta 23
2.4 Architecture existante 25
2.4.1 Hébergement VPS et son fonctionnement 26
2.4.1.1 Bénéfices de l'hébergement VPS 26
2.4.1.2 Informations système du VPS de Volkeno 26
2.4.1.3 Architecture du VPS 27
2.5 Conclusion 30
CHAPITRE 3: CONCEPTION ET ÉTUDE COMPARATIVE DES OUTILS D'ANALYSE DE SÉCURITÉ 31
3.1 Introduction 31
3.2 Etudes comparatives 31
3.2.1 Outils de scan réseau 32
3.2.1.1 Nmap 32
3.2.1.2 Masscan 34
3.2.1.3 Unicornscan 35
3.2.1.4 Synthèse de l'étude comparative des outils de scan réseau 36
3.2.2 Outils de scan de vulnérabilités 38
3.2.2.1 OWASP ZAP 38
3.2.2.2 Burp Suite 40

3.2.2.3 Nessus....42
3.2.2.4 Synthèse de l'étude comparative des outils de scan des vulnérabilités....43
3.2.3 Solutions d'hébergement....44
3.2.3.1 Hébergement VPS....45
3.2.3.2 Serveur dédié....46
3.2.3.3 Hébergement partagé....47
3.2.3.4 Synthèse de l'étude comparative des solutions d'hébergement....48
3.2.4 Outils de surveillance....49
3.2.4.1 Nagios....49
3.2.4.2 Prometheus....50
3.2.4.3 UptimeRobot....51
3.2.4.4 Synthèse de l'étude comparative des outils de surveillance....53
3.3 Maquette de test....54
3.3.1 Présentation des Outils Utilisés....54
3.3.2 Fonctionnement de la maquette de Test....54
3.3.3 Corrélation des Résultats et Analyse des Vulnérabilités....56
3.4 Conclusion....56
CHAPITRE 4 : IDENTIFICATION DES VULNÉRABILITÉS EXISTANTES ET ANALYSES DES RISQUES....57
4.1 Introduction....57
4.2 Définition du périmètre d'analyse....57
4.2.1 Objectifs de l'analyse....57
4.2.2 Systèmes cibles....58
4.2.3 Limites de l'analyse....58
4.3 Collecte d'informations et reconnaissance....59
4.3.1 Recueillir des informations sur le système....59
4.3.2 Cartographie du réseau et des services....60
4.4 Utilisation des outils d'analyse des vulnérabilités....62
4.4.1 Scan de vulnérabilités avec OWASP ZAP....62
4.4.2 Tests manuels pour la validation des vulnérabilités....66
4.5 Identification et classification des vulnérabilités....68
4.5.1 Identification des vulnérabilités....68
4.5.1.1 Métadonnée du Cloud potentiellement vulnérable....69
4.5.1.2 En-têtes HTTP de Sécurité non définies....69
4.5.1.3 Problèmes de stockage et de divulgation d'informations....71
4.5.1.4 Problèmes liés au cache et à la mise en cache....72
4.5.1.5 Gestion des sessions et de l'authentification....73
4.5.1.6 Serveur & Technologies....74
4.5.2 Classification des vulnérabilités....75
4.6 Évaluation des risques....76
4.7 Conclusion....77

CHAPITRE 5 : PROPOSITION DE MÉCANISMES DE PROTECTION..... 79
5.1 Introduction..... 79
5.2 Approche générale de la protection..... 79
5.2.1 Stratégie de défense en profondeur..... 79
5.2.2 Modèle de gestion des risques..... 80
5.3 Mécanismes de protection proposés..... 80
5.3.1 Sécurisation des métadonnées du Cloud..... 80
5.3.2 Définition des en-têtes HTTP de sécurité..... 81
5.3.3 Problèmes de stockage et de divulgation d'informations..... 83
5.3.4 Problèmes liés au cache et à la mise en cache..... 84
5.3.5 Gestion des sessions et de l'authentification..... 85
5.3.6 Sécurisation du serveur et des technologies utilisées..... 86
5.3.7 Surveillance et détection des incidents..... 86
5.4 Conclusion..... 87
CHAPITRE 6 : RENFORCEMENT DES CAPACITÉS EN SÉCURITÉ APPLICATIVE..... 89
6.1 Introduction..... 89
6.2 Sensibilisation des Parties Prenantes..... 89
6.2.1 Sensibilisation des Développeurs..... 89
6.2.1.1 Connaissance des Vulnérabilités et des Menaces..... 90
6.2.1.2 Adoption des Bonnes Pratiques de Développement Sécurisé..... 90
6.2.1.3 Sécurisation des Données et des Communications..... 91
6.2.2 Formation des Administrateurs Systèmes et Réseaux..... 91
6.2.2.1 Compréhension des Menaces et des Risques..... 91
6.2.2.2 Sécurisation des Systèmes et des Accès..... 92
6.2.2.3 Sécurisation du Réseau et des Communications..... 93
6.2.2.4 Surveillance, Détection et Réponse aux Incidents..... 93
6.2.2.5 Formation Continue et Certification..... 94
6.2.3 Sensibilisation des Utilisateurs Finaux..... 95
6.2.3.1 Compréhension des Enjeux de Sécurité..... 95
6.2.3.2 Adoption des Bonnes Pratiques de Sécurité..... 95
6.2.3.3 Mise en Place de Programmes de Sensibilisation..... 96
6.3 Documentation des Procédures et Bonnes Pratiques..... 97
6.3.1 Élaboration d'un Guide de Sécurité pour les Développeurs..... 97
6.3.2 Documentation des Procédures de Sécurisation des Infrastructures..... 98
6.3.3 Mise en Place d'une Politique de Gestion des Vulnérabilités..... 98
6.4 Conclusion..... 99
Conclusion Générale..... 100
Webographie et Bibliographie..... 101

Liste des figures

Figure 1 : Architecture d’un vps..... 27
Figure 2 : Architecture de Easy Compta jusqu’au VPS..... 28
Figure 3 : Flux de données..... 29
Figure 4 : ZAP entre le navigateur et l’application..... 39
Figure 5 : ZAP associé à un network proxy..... 39
Figure 6 : Maquette de test..... 55
Figure 7 : Maquette de test (suite)..... 56
Figure 8 : Scan réseau avec Nmap..... 61
Figure 9 : Scan réseau avec Nmap (suite)..... 61
Figure 10 : Scan des vulnérabilités avec OWASP ZAP..... 63
Figure 11 : Scan des vulnérabilités avec OWASP ZAP (la sortie de l’historique)..... 64
Figure 12 : Scan des vulnérabilités avec OWASP ZAP (les alertes)..... 64
Figure 13 : Page d’authentification..... 67
Figure 14 : Page d’administration django..... 67
Figure 15 : Interface de l’admin django..... 68
Figure 16 : Surveillance avec UptimeRobot..... 87

Liste des tableaux

Tableau 1 : Comparaison entre les outils de scan réseau..... 37
Tableau 2 : Comparaison entre les outils de scan de vulnérabilités..... 43
Tableau 3 : Comparaison entre les solutions d'Hébergement..... 48
Tableau 4 : Comparaison entre les outils de surveillances..... 53
Tableau 5 : Nombre d’alertes pour niveau de risque et confiance..... 63
Tableau 6 : Classification des vulnérabilités..... 72
Tableau 7 : Matrice d’évaluation des risques..... 74

Glossaire :

- **Authentification** : Processus permettant de vérifier l'identité d'un utilisateur avant de lui accorder l'accès à un système ou une application.
- **Autorisation** : Mécanisme qui définit les actions qu'un utilisateur authentifié est autorisé à effectuer dans un système.
- **Brute Force Attack (Attaque par force brute)** : Technique d'attaque consistant à tester toutes les combinaisons possibles de mots de passe pour un système.
- **Cross-Site Scripting (XSS)** : Vulnérabilité qui permet à un attaquant d'injecter du code malveillant (souvent JavaScript) dans une page web consultée par d'autres utilisateurs.
- **Déni de service (DoS/DDoS)** : Attaque visant à rendre un service indisponible en le surchargeant de requêtes malveillantes.
- **DevSecOps** : Approche intégrant la sécurité dès les premières phases du développement logiciel, dans une logique DevOps.
- **Injection SQL** : Attaque qui exploite des failles de sécurité dans les requêtes SQL afin d'accéder illégalement à une base de données.
- **OWASP (Open Web Application Security Project)** : Organisation qui publie des guides et recommandations pour améliorer la sécurité des applications web, notamment l'OWASP Top 10.
- **Pentesting (Test d'intrusion)** : Méthodologie consistant à simuler des attaques pour identifier et corriger les failles de sécurité d'un système.
- **Sécurité applicative** : Ensemble des pratiques et technologies visant à protéger les applications contre les cyberattaques et les vulnérabilités.
- **Session Hijacking (Détournement de session)** : Attaque où un pirate s'empare de la session d'un utilisateur légitime pour accéder à une application sans autorisation.
- **Token d'authentification** : Jeton numérique utilisé pour valider l'identité d'un utilisateur et maintenir une session sécurisée.
- **Vulnérabilité** : Faiblesse dans un système ou une application pouvant être exploitée par un attaquant pour nuire à sa sécurité.
- **Faille :** Une **faille de sécurité** est une **vulnérabilité** ou une **erreur** dans un système, une application ou un processus qui peut être exploitée par un attaquant.

- Risque : Un **risque de sécurité** représente **la probabilité qu'une faille soit exploitée** et les **conséquences** qui en découlent.

Acronymes

- **ACL** : Access Control List (Liste de contrôle d’accès)
- **CORS :** Cross-Origin Resource Sharing (Partage des ressources entre origines multiples)
- **CSRF :** Cross-Site Request Forgery (Falsification de requête intersite)
- **CVE** : Common Vulnerabilities and Exposures (Référentiel des vulnérabilités connues)
- **DevSecOps :** Développement, Sécurité, Opérations (Développement, Sécurité, Opérations)
- **DDoS :** Déni de service distribué
- **DNS :** Domain Name System (Système de noms de domaine)
- **DoS :** Déni de service
- **HTTPS :** HyperText Transfer Protocol Secure (Protocole de transfert hypertexte sécurisé)
- **IA** : Intelligence Artificielle (Artificial Intelligence)
- **IDS :** Intrusion Detection System (Système de détection d'intrusion)
- **IPS :** Système de prévention d'intrusion
- **IXP** : Internet Exchange Point (Point d’échange Internet)
- **JSON :** JavaScript Object Notation (Format d'échange de données léger)
- **JWT :** JSON Web Token (Jeton d'authentification web)
- **MFA** : Multi-Factor Authentication (Authentification multifacteur)
- **MITM :** Man-In-The-Middle (Attaque de l'homme du milieu)
- **NIST :** Institut national des normes et de la technologie
- **OWASP** : Open Web Application Security Project (Projet de sécurité des applications web)
- **OWASP ZAP** : OWASP Zed Attack Proxy (Outil de test de sécurité des applications web)
- **RBAC :** Contrôle d'accès basé sur les rôles
- **RDP** : Remote Desktop Protocol (Protocole de bureau à distance)
- **SaaS** : Software as a Service (service basé sur le cloud)
- **SAST :** Test de sécurité statique des applications
- **SQLi :** Injection SQL (SQL Injection)
- **TLS :** Transport Layer Security (Sécurité de la couche de transport)

- **VPS** : Virtual Private Server (Serveur Privé Virtuel)
- **WAF :** Web Application Firewall (Pare-feu d'application web)
- **XSS :** Cross-Site Scripting (Injection de script intersite)

Introduction Générale

Dans un contexte où la digitalisation des services devient incontournable, les applications web jouent un rôle central dans la gestion des activités des entreprises. Cependant, cette évolution s'accompagne de nombreux défis en matière de sécurité, notamment face à la recrudescence des cyberattaques ciblant les applications en ligne. Les failles de sécurité peuvent exposer les données sensibles des utilisateurs, compromettre l'intégrité des systèmes et nuire à la réputation des entreprises.

L'application **Easy Compta**, développée pour la gestion comptable, s'inscrit dans cette dynamique en offrant aux utilisateurs une plateforme simplifiant la gestion financière. Toutefois, comme toute application web, elle peut être exposée à diverses vulnérabilités, notamment celles répertoriées dans l'OWASP Top 10, telles que l'injection SQL, le Cross-Site Scripting (XSS) ou encore les failles liées à une gestion insuffisante des accès et des sessions.

Ce mémoire vise à **identifier et analyser les vulnérabilités existantes dans l'application Easy Compta, puis à proposer des mécanismes de protection adaptés afin de renforcer sa sécurité**. Pour cela, nous adopterons une méthodologie basée sur des tests d'intrusion (pentesting), des analyses du code, ainsi que l'utilisation d'outils de détection des vulnérabilités.

L'objectif principal est d'apporter des solutions concrètes et éprouvées pour améliorer la robustesse de l'application face aux menaces potentielles. L'implémentation de bonnes pratiques de développement sécurisé, le durcissement des configurations serveur et l'intégration de solutions de surveillance feront partie des mesures proposées.

Ce document est structuré comme suit :

- **Le premier chapitre** présente le **contexte**, la **problématique**, les **objectifs de l'étude**, l'**importance et l'intérêt du sujet**, ainsi que la **méthodologie de recherche** adoptée.
- **Le deuxième chapitre** est consacré à la **présentation de l'application Easy Compta**, son **cadre de développement** ainsi que son **architecture**.

- **Le troisième chapitre** propose une **étude comparative des outils** qui seront utilisés pour l'analyse de sécurité.
- **Le quatrième chapitre** se focalise sur **l'identification des vulnérabilités existantes**, l'**analyse des risques** et leur **impact**.
- **Le cinquième chapitre** expose les **mécanismes de protection** mis en place pour sécuriser l'application.
- **Enfin, le dernier chapitre** met l'accent sur la **sensibilisation des parties prenantes** aux **enjeux de la sécurité applicative**.

À travers cette étude, nous souhaitons non seulement améliorer la sécurité de l'application Easy Compta, mais également contribuer à la sensibilisation aux bonnes pratiques de cybersécurité dans le développement des applications web.

CHAPITRE 1 : CADRAGE DE L'ÉTUDE

1.1 Introduction

Avec l'adoption massive de l'internet, la montée en puissance des technologies mobiles et l'explosion des réseaux sociaux, les développeurs d'applications Web se concentrent de plus en plus sur l'amélioration du contenu graphique et fonctionnel de leurs plateformes. Cependant, cette quête de performance et d'innovation se fait souvent au détriment des contrôles de sécurité. Cette négligence crée un terrain propice aux cyberattaques, qui exploitent les failles de sécurité pour atteindre leurs objectifs malveillants.

Les conséquences de ces attaques sont souvent graves, touchant aussi bien les individus que les organisations. Comprendre le fonctionnement de ces menaces est donc essentiel pour concevoir et mettre en œuvre des solutions efficaces capables d'atténuer leurs impacts négatifs.

Dans ce chapitre, nous détaillons notre méthodologie de travail. Nous commencerons par une contextualisation du sujet, suivie d'une présentation de la problématique sous forme de questions pratiques. Nous définirons ensuite les objectifs du mémoire, mettrons en avant l'importance et les intérêts du sujet, et décrirons finalement la méthodologie adoptée pour mener à bien cette recherche.

1.2 Contexte

Dans un monde de plus en plus interconnecté, les entreprises s'appuient fortement sur les applications Web pour gérer leurs activités quotidiennes et améliorer leur productivité. Ces applications, telles que **Easy Compta**, offrent des solutions pratiques pour la gestion comptable, facilitant l'accès aux données en temps réel, l'automatisation des processus et la collaboration entre les parties prenantes. Cependant, cette dépendance accrue aux outils numériques s'accompagne de nouveaux défis, notamment en matière de sécurité.

Les applications Web sont constamment exposées à des menaces diverses telles que les injections SQL, les attaques par déni de service distribué (DDoS) ou encore les violations d'authentification. Ces vulnérabilités peuvent découler de facteurs tels que :

- **L’évolution rapide des technologies** : Les cycles de développement accélérés favorisent parfois des failles non détectées;
- **La complexité croissante des applications** : Des fonctionnalités plus riches augmentent la surface d’attaque;
- **Le manque de sensibilisation des développeurs** : Les pratiques de sécurité sont parfois négligées face aux exigences de performance et d’innovation.

Easy Compta, en tant qu’application Web utilisée pour la gestion comptable, contient des données sensibles qui, si elles sont compromises, pourraient entraîner des pertes financières, des atteintes à la confidentialité des informations, ou des interruptions opérationnelles. La sécurité de cette application revêt donc une importance capitale pour garantir la confiance des utilisateurs et la pérennité des opérations.

Face à ces enjeux, une analyse approfondie des vulnérabilités d’Easy Compta, suivie d’une proposition de mécanismes de protection adaptés, est indispensable. Ce travail s’inscrit dans cette perspective en combinant une démarche technique et stratégique pour répondre aux menaces actuelles et futures.

1.3 Problématique

Les applications Web, en particulier celles manipulant des données sensibles, sont aujourd’hui la cible principale des cyberattaques. **Easy Compta**, en tant qu’application de gestion comptable, centralise des informations critiques, notamment des données financières et personnelles. Cependant, comme de nombreuses applications Web, elle peut être exposée à des vulnérabilités exploitables par des attaquants.

La problématique centrale de ce mémoire est la suivante :

Quelles sont les principales vulnérabilités de sécurité présentes dans l'application Easy Compta et comment les détecter ?

Quels sont les risques liés à la protection des données sensibles des utilisateurs et comment les réduire ?

Quels mécanismes de sécurité peuvent être mis en place pour renforcer la protection de l'application contre les menaces futures ?

La principale question de recherche liée à cette étude est :

Quelles stratégies d'automatisation peuvent être mises en œuvre pour vérifier la présence et la bonne configuration des en-têtes de sécurité dans un environnement de production ?

1.4 Objectifs du mémoire

L'objectif principal de ce mémoire est **d'évaluer et d'améliorer la sécurité de l'application Easy Compta en identifiant ses vulnérabilités, en analysant leurs impacts et en mettant en place des solutions adaptées pour les corriger et renforcer la résilience du système face aux menaces.**

Pour atteindre cet objectif général, plusieurs objectifs spécifiques ont été définis :

1. **Identification des vulnérabilités existantes et Analyse des risques :**
    - Réaliser une analyse approfondie de l'application Easy Compta pour détecter les failles de sécurité potentielles;
    - Utiliser des outils de test de sécurité tels que les scanners d'application Web pour identifier les points d'injection et autres vulnérabilités;
    - Évaluer les risques associés aux failles identifiées en tenant compte de la confidentialité, de l'intégrité et de la disponibilité des données;
    - Mesurer l'impact potentiel de ces vulnérabilités sur les utilisateurs et l'organisation.
2. **Proposition de mécanismes de protection :**
    - Proposer des stratégies de prévention, notamment l'adoption de bonnes pratiques en matière de développement sécurisé et de configuration des systèmes.
3. **Renforcement des capacités en sécurité applicative :**
    - Sensibiliser les parties prenantes (développeurs, administrateurs système) aux enjeux de sécurité;
    - Documenter les procédures et bonnes pratiques pour une gestion proactive de la sécurité de l'application.

1.5 Importance et intérêts du sujet

La sécurité des applications Web est devenue un enjeu stratégique pour les entreprises et les organisations dans un monde de plus en plus numérique. L'application **Easy Compta**, en tant que solution de gestion comptable, manipule des données critiques telles que des informations financières et personnelles. Sa sécurisation est donc essentielle, non seulement pour protéger ces données, mais aussi pour garantir la continuité des activités et la confiance des utilisateurs.

Importance du sujet

1. **Protection des données sensibles :**
   Les applications comme Easy Compta traitent des données confidentielles qui, en cas de compromission, pourraient entraîner des pertes financières, des litiges juridiques ou une atteinte à la réputation de l'entreprise.
2. **Réduction des risques opérationnels :**
   Une attaque réussie sur Easy Compta pourrait perturber les activités des utilisateurs, notamment par la perte ou la modification de données, compromettant ainsi les opérations quotidiennes des entreprises qui s'appuient sur cette application.
3. **Contribution à la sensibilisation à la cybersécurité :**
   En étudiant et en appliquant des mécanismes de protection à Easy Compta, ce mémoire contribue à renforcer la sensibilisation aux enjeux de la sécurité applicative, un domaine souvent négligé dans les processus de développement logiciel.

Intérêts du sujet

1. **Pour les utilisateurs :**
   Garantir la sécurité d'Easy Compta permet aux utilisateurs de bénéficier d'une application fiable, réduisant les risques de compromission des données et augmentant la confiance dans le système.
2. **Pour l'entreprise :**
   La sécurisation d'Easy Compta renforce l'image de l'entreprise en tant que fournisseur de solutions numériques fiables et conformes aux normes de sécurité. Cela pourrait également réduire les coûts liés à la gestion des incidents de sécurité.
3. **Pour la recherche académique et professionnelle :**
   Ce travail constitue une contribution précieuse dans le domaine de la sécurité

applicative. Il illustre des cas pratiques d'identification et de correction de vulnérabilités, ce qui peut servir de référence pour d'autres projets similaires.

4. **Pour le développement personnel et professionnel :**
   La réalisation de cette étude permet de développer des compétences techniques avancées en matière de cybersécurité, notamment l'utilisation d'outils d'analyse et la mise en œuvre de mécanismes de protection.

1.6 Méthodologie

La méthodologie adoptée pour sécuriser l'application **Easy Compta** se compose de plusieurs étapes essentielles, visant à identifier, corriger et prévenir les vulnérabilités de sécurité tout en maintenant une approche proactive et continue de la sécurité.

1. Audit de sécurité

Le premier volet de la méthodologie consiste à effectuer un audit complet de l'application pour détecter les failles existantes. Cela inclut :

- **Analyse statique** : Étudier le code source de l'application pour identifier les vulnérabilités logicielles (vérification des injections SQL, XSS, fuites de données, etc.);
- **Analyse dynamique** : Tester l'application en fonctionnement pour découvrir les vulnérabilités exploitables en temps réel;
- **Utilisation d'outils** : Recourir à des outils spécialisés comme OWASP ZAP, SQLmap, Burp Suite et autres scanners de sécurité pour une détection précise des failles.

2. Plan de remédiation

Sur la base des résultats de l'audit, un plan de remédiation est élaboré. Ce plan vise à prioriser les corrections en fonction de la criticité des vulnérabilités identifiées. Les étapes incluent :

- **Évaluation des vulnérabilités** : Utilisation de modèles comme CVSS (Common Vulnerability Scoring System) pour évaluer la gravité des failles;

- **Priorisation** : Corriger en priorité les failles critiques ayant un impact significatif sur la confidentialité, l'intégrité ou la disponibilité des données;
- **Correction progressive** : Implémenter les corrections par étapes pour minimiser les interruptions et tester chaque étape avant d'aller plus loin.

3. Mise en œuvre des protections

Une fois les vulnérabilités corrigées, des mesures de protection sont appliquées de manière progressive. Cela inclut :

- **Gestion des sessions sécurisées** : Renforcement des mécanismes de gestion des sessions avec des tokens d'authentification sécurisés;
- **Filtrage des entrées** : Mise en œuvre de validations strictes des entrées utilisateur pour éviter les injections malveillantes.

4. Suivi continu

Pour maintenir une sécurité optimale, un processus de surveillance continue est instauré, comprenant :

- **Surveillance régulière** : Surveillance en temps réel des systèmes pour détecter d'éventuelles activités malveillantes ou anomalies de sécurité;
- **Audits périodiques** : Réalisation d'audits de sécurité réguliers pour identifier les nouvelles menaces et les vulnérabilités émergentes;
- **Améliorations continues** : Intégration des retours d'expérience et des nouvelles technologies de sécurité dans les mises à jour régulières de l'application.

5. Formation des développeurs aux meilleures pratiques de sécurité et aux nouvelles menaces :

Une des étapes clés consiste à former les développeurs et les parties prenantes sur la sécurité applicative. Cela inclut :

- **Intégration de DevSecOps** : Encourager l'intégration de la sécurité dès la phase de conception des applications pour minimiser les risques dès le départ;
- **Formation continue** : Sensibilisation régulière aux nouvelles menaces, à l'évolution des normes de sécurité et à l'utilisation des outils de sécurité modernes;

- **Documentation des pratiques** : Mettre en place des guides et des procédures pour assurer un développement sécurisé tout au long du cycle de vie des applications.

1.7 Conclusion

Dans ce chapitre, nous avons présenté l'approche méthodologique adoptée pour sécuriser l'application **Easy Compta** face aux menaces croissantes pesant sur les applications Web. La combinaison d'un audit de sécurité rigoureux, d'un plan de remédiation structuré, de l'application de mécanismes de protection, et d'un suivi continu assure une démarche systématique et efficace pour réduire les risques de cyberattaques.

De plus, l'intégration de la sécurité dès les premières étapes du cycle de développement (DevSecOps) et la formation des équipes aux meilleures pratiques représentent des piliers essentiels pour maintenir une application fiable et robuste à long terme.

Cette méthodologie constitue une base solide pour atteindre les objectifs fixés dans ce mémoire, tout en soulignant l'importance d'une gestion proactive et continue de la sécurité. Les prochaines sections détailleront les résultats obtenus à travers cette approche et les solutions proposées pour répondre aux défis identifiés.

CHAPITRE 2: ETAT DES LIEUX

2.1 Introduction

Pour assurer une sécurisation optimale de l'application **Easy Compta** , il est essentiel de dresser un état des lieux complet et détaillé de son environnement actuel. Ce chapitre a pour objectif de fournir une analyse approfondie des différents éléments susceptibles d'impacter la sécurité de l'application.

Dans un premier temps, nous présentons l'entreprise qui développe **Easy Compta** , en abordant ses objectifs, son contexte organisationnel et les enjeux stratégiques associés à la gestion de ses systèmes d'information. Cela permettra de mieux comprendre l'importance de la sécurité dans le cadre de l'utilisation de cette application au sein de l'entreprise.

Puis, une description détaillée de **l'application Easy Compta** sera effectuée. Cette analyse couvre les principales fonctionnalités de l'application, ainsi que les technologies et outils utilisés pour son développement. L'objectif est d'identifier les aspects techniques et structurels qui peuvent présenter des vulnérabilités ou des risques pour la sécurité.

Cet état des lieux constitue une étape fondamentale, car il offre une vision claire des défis spécifiques rencontrés par **Easy Compta** , ainsi qu'une base solide pour la proposition de solutions de sécurisation adaptées et efficaces.

2.2 Présentation de l'entreprise

2.2.1 Objectifs et Mission de l'Entreprise

Volkeno est une entreprise sénégalaise fondée en 2015 à Dakar, qui se positionne comme un acteur clé dans le domaine de la transformation numérique en Afrique. Son objectif principal est de développer les compétences numériques des jeunes africains tout en les formant aux nouveaux métiers du digital. L'entreprise s'articule autour de deux grands axes :
la formation professionnelle à travers son programme phare **Bakeli**, et le développement de solutions numériques en accompagnant des projets technologiques via son startup.

Volkeno offre une expertise dans plusieurs domaines liés au digital, notamment le développement web et mobile, l'UX/UI design, et le marketing digital. Volkeno développe des projets pour ses clients en proposant divers services liés aux nouvelles technologies. L'entreprise accompagne les entreprises et startups dans la création et l'optimisation de leurs solutions numériques, notamment dans les domaines suivants :

- **Développement logiciel** : Création d'applications web et mobiles sur mesure;
- **Design UX/UI** : Conception d'interfaces utilisateur intuitives et ergonomiques;
- **Marketing digital** : Stratégies de communication et de visibilité en ligne;
- **Infographie et Montage vidéo** : Création de contenus visuels et audiovisuels.

L'entreprise fonctionne également comme un startup studio, développant ses propres solutions et incubant des projets innovants en interne.

Avec une approche orientée vers l'innovation et le développement durable, Volkeno s'efforce de répondre aux besoins croissants des entreprises en matière de transformation numérique, tout en créant un impact positif dans la société africaine.
Son ambition à long terme est de catalyser la création de 1000 startups en Afrique au cours des 10 prochaines années. Elle envisage un avenir numérique où ses solutions tiennent compte des réalités sociétales du continent africain.

2.2.2 Enjeux stratégiques

Volkeno développe **Easy Compta** en **collaboration étroite avec le cabinet CECA** (Centre d'Études, de Conseil et d'Audit), une entreprise spécialisée en **audit, conseil et expertise comptable**. L'objectif est de concevoir une **solution comptable sur mesure**, parfaitement alignée avec les exigences comptables et fiscales des entreprises et des professionnels. Easy Compta traite des données sensibles qui sont cruciales tant pour les clients que pour la réputation de l'entreprise elle-même. Ces données comprennent :

- **Données personnelles** : Cela inclut les informations personnelles des utilisateurs, telles que le nom, l'adresse, le numéro de téléphone et les identifiants de connexion (nom d'utilisateur, mot de passe). Ces informations doivent être protégées pour éviter des violations de données qui pourraient entraîner des pertes de confiance, des sanctions financières ou juridiques, et des atteintes à la réputation de l'entreprise;

- **Données financières** : **Easy Compta** manipule des informations critiques liées aux finances des entreprises et des particuliers, telles que des transactions bancaires, des bilans financiers, des rapports de dépenses et des déclarations fiscales. Ces données sont particulièrement sensibles car elles peuvent être exploitées à des fins frauduleuses en cas de compromission. Par exemple, les informations sur les comptes bancaires ou les numéros de cartes de crédit doivent être protégées contre le vol ou l'accès non autorisé;
- **Données fiscales et comptables** : Les informations relatives aux déclarations fiscales et aux états financiers sont également des données sensibles, souvent soumises à des réglementations strictes. Une fuite ou une manipulation incorrecte de ces données pourrait non seulement nuire à la sécurité financière des clients, mais aussi entraîner des conséquences juridiques pour **Volkeno**.

2.3 Caractéristiques de l'application Easy Compta

Easy Compta est une application web de gestion comptable destinée principalement aux petites et moyennes entreprises (PME). Son objectif est de simplifier et d'automatiser les processus comptables quotidiens, permettant ainsi aux entreprises de se concentrer sur leur cœur de métier sans avoir à se soucier des tâches administratives complexes liées à la comptabilité. Voici un aperçu détaillé des fonctionnalités principales d' Easy Compta :

1. **Gestion des Factures**

L'application permet aux utilisateurs de créer, envoyer et suivre leurs factures en ligne, facilitant ainsi la gestion des transactions commerciales. Les fonctionnalités clés de la gestion des factures incluent :

- **Création de factures :** Possibilité de générer des factures personnalisées avec des informations détaillées sur les produits ou services fournis, les taxes applicables, et les conditions de paiement;
- **Envoi de factures :** Les factures peuvent être envoyées directement aux clients par email depuis l'application, notamment les délais de transmission et d'archivage;
- **Suivi des factures :** Les utilisateurs peuvent suivre l'état de leurs factures (payée, en attente de paiement, en retard), ce qui permet de maintenir une bonne gestion des créances.

2. **Suivi des paiements**

Easy Compta permet aux entreprises de suivre les paiements reçus et en attente , ce qui facilite la gestion de la trésorerie et permet de s'assurer que les paiements sont effectués en temps et en heure. Cette fonctionnalité inclut :

- **Enregistrement des paiements :** Lorsqu'un paiement est effectué, il peut être enregistré et lié à la facture correspondante, permettant une gestion efficace des flux financiers;
- **Alertes de paiements en retard :** L'application peut envoyer des notifications aux utilisateurs lorsque des paiements sont en retard, leur permettant de prendre des mesures appropriées pour recouvrer les créances.

3. **Gestion des Clients et Fournisseurs**

Une gestion efficace des relations commerciales est essentielle pour toute entreprise. Easy Compta permet la création et la mise à jour des profils clients et fournisseurs , en centralisant toutes les informations importantes dans une base de données. Cela inclut :

- **Informations détaillées :** Chaque profil peut contenir des informations clés telles que le nom, l'adresse, les coordonnées, les conditions de paiement et l'historique des transactions;
- **Gestion des contacts :** Cette fonctionnalité permet également de suivre l'historique des communications avec chaque client ou fournisseur, ce qui aide à maintenir une bonne relation professionnelle.

4. **Génération de Rapports Financiers**

L'application offre des rapports financiers automatiques , permettant aux utilisateurs de suivre la santé financière de leur entreprise en temps réel. Les rapports générés comprennent :

- **Revenus et dépenses :** Easy Compta génère des rapports détaillant les revenus et les dépenses sur une période donnée, facilitant ainsi l'analyse financière;
- **Soldes financiers :** Les utilisateurs peuvent générer des rapports de bilan pour évaluer la rentabilité de leur entreprise, la liquidité, et les soldes de leurs comptes.

5. **Gestion des États Financiers**

La fonctionnalité de gestion des états financiers permet aux utilisateurs de générer des bilans financiers détaillés , des comptes de résultat et d'autres rapports nécessaires pour comprendre la situation financière de l'entreprise. Ces rapports sont essentiels lors de la préparation des déclarations fiscales, ainsi que pour les évaluations internes de la performance de l'entreprise.

6. **Notifications et Rappels**

Afin d'améliorer la gestion des paiements et des échéances, l'application offre un système de notifications et rappels automatiques . Cela inclut :

- **Alertes pour paiements en retard :** Les utilisateurs sont informés des paiements impayés et des dates limites pour effectuer des paiements, ce qui réduit les risques d'impayés;
- **Rappels pour échéances fiscales ou autres obligations :** Easy Compta rappelle aux utilisateurs les échéances fiscales ou autres obligations comptables importantes, contribuant à assurer la conformité avec les réglementations.

Technologies utilisées

Easy Compta s’appuie sur des technologies modernes pour assurer sa performance et son évolutivité :

- **Frontend** : React.js pour une interface utilisateur interactive et dynamique;
- **Backend** : Django Rest Framework pour la gestion des API et des fonctionnalités côté serveur;
- **Base de données** : PostGresqL pour le stockage des données transactionnelles et administratives;
- **API REST** : Communication entre le frontend et le backend via des endpoints sécurisés.

Public cible

Easy Compta cible principalement :

- **Petites et moyennes entreprises (PME)** : Qui cherchent une solution économique et simple pour gérer leurs finances;
- **Professionnels indépendants** : tels que les consultants et les freelances;

- **Entrepreneurs débutants** : Avec peu de connaissances comptables.

2.4 Architecture existante

L'application suit une architecture client-serveur basée sur les principes RESTful :

- **Frontend** : Application React.js déployée sur un serveur web;
- **Backend** : Serveur Django REST déployé sur un VPS;
- **Base de données** : Stockée sur le même serveur que le backend.

Hébergement :

- **Type d'hébergement** : Actuellement déployé sur un **VPS** (Virtual Private Server);
- **Système d'exploitation** : Linux (Debian), utilisé pour sa stabilité et sa compatibilité avec les technologies web;
- **Protocole réseau** : Utilisation de HTTP pour les échanges.

2.4.1 Hébergement VPS et son fonctionnement

Volkeno utilise un **VPS (Virtual Private Server)** pour héberger ses applications, notamment **Easy Compta**. Ce serveur virtuel est fourni par **DigitalOcean**, une plateforme cloud réputée pour ses solutions flexibles et performantes.

2.4.1.1 Bénéfices de l'hébergement VPS

Un serveur privé virtuel (VPS) est une machine qui héberge tous les logiciels et les données nécessaires au fonctionnement d'une application ou d'un site web.

Le **VPS** est une architecture hybride permettant de bénéficier de l'isolement et des ressources dédiées d'un serveur dédié, tout en étant plus flexible et moins coûteux qu'un serveur entièrement dédié.

L'hébergement sur VPS permet à Volkeno de bénéficier :

- **d'une infrastructure dédiée** : contrairement à l'hébergement mutualisé, un VPS offre des ressources dédiées (CPU, RAM, stockage).

- **d’une meilleure sécurité** : les données sensibles des clients d’Easy Compta sont stockées de manière sécurisée avec des accès restreints.
- **d’une scalabilité** : Volkeno peut facilement augmenter la puissance du serveur si les besoins évoluent.
- **d’un contrôle total** : le VPS permet d’installer et de configurer les logiciels et services selon les besoins.

2.4.1.2 Informations système du VPS de Volkeno

Voici les principales caractéristiques du serveur :

- **Fournisseur** : DigitalOcean (Droplet)
- **Système d’exploitation** : Debian GNU/Linux 11 (Bullseye)
- **Processeur** : 4 vCPU
- **Mémoire RAM** : 6 Go
- **Nom d’hôte** : debian-s-1vcpu-1gb-fra1-01-crafit
- **Sécurité** : Clés SSH activées

2.4.1.3 Architecture du VPS

Le **VPS** fonctionne grâce à une **architecture virtuelle** , où un serveur physique, appelé **matériel nu** , est subdivisé en plusieurs **machines virtuelles (VM)** grâce à une couche de virtualisation, également appelée **hyperviseur** . L'hyperviseur permet de diviser le serveur physique en **segments virtuels indépendants** , chacun possédant ses propres ressources attribuées, telles que le processeur, la mémoire RAM, le système d'exploitation et les applications. Cette technologie est souvent appelée **virtualisation KVM** (Kernel-based Virtual Machine).

Chaque **machine virtuelle** fonctionne comme un serveur indépendant, avec un système d'exploitation et des applications dédiées, ce qui permet une gestion isolée de chaque instance. Ainsi, chaque utilisateur ou application ayant un VPS bénéficie d'une configuration personnalisée et sécurisée, tout en partageant les ressources du serveur physique de manière optimisée.

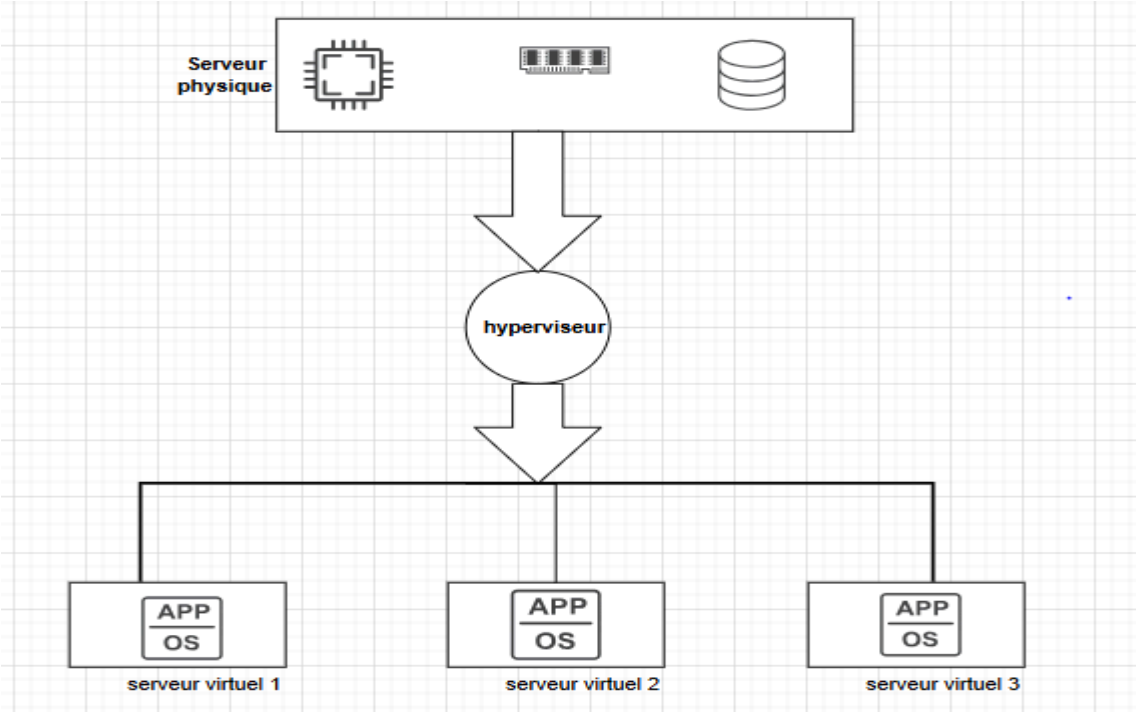


Figure 1 : Architecture d'un vps

Voici l'architecture existante de **Easy Compta** :

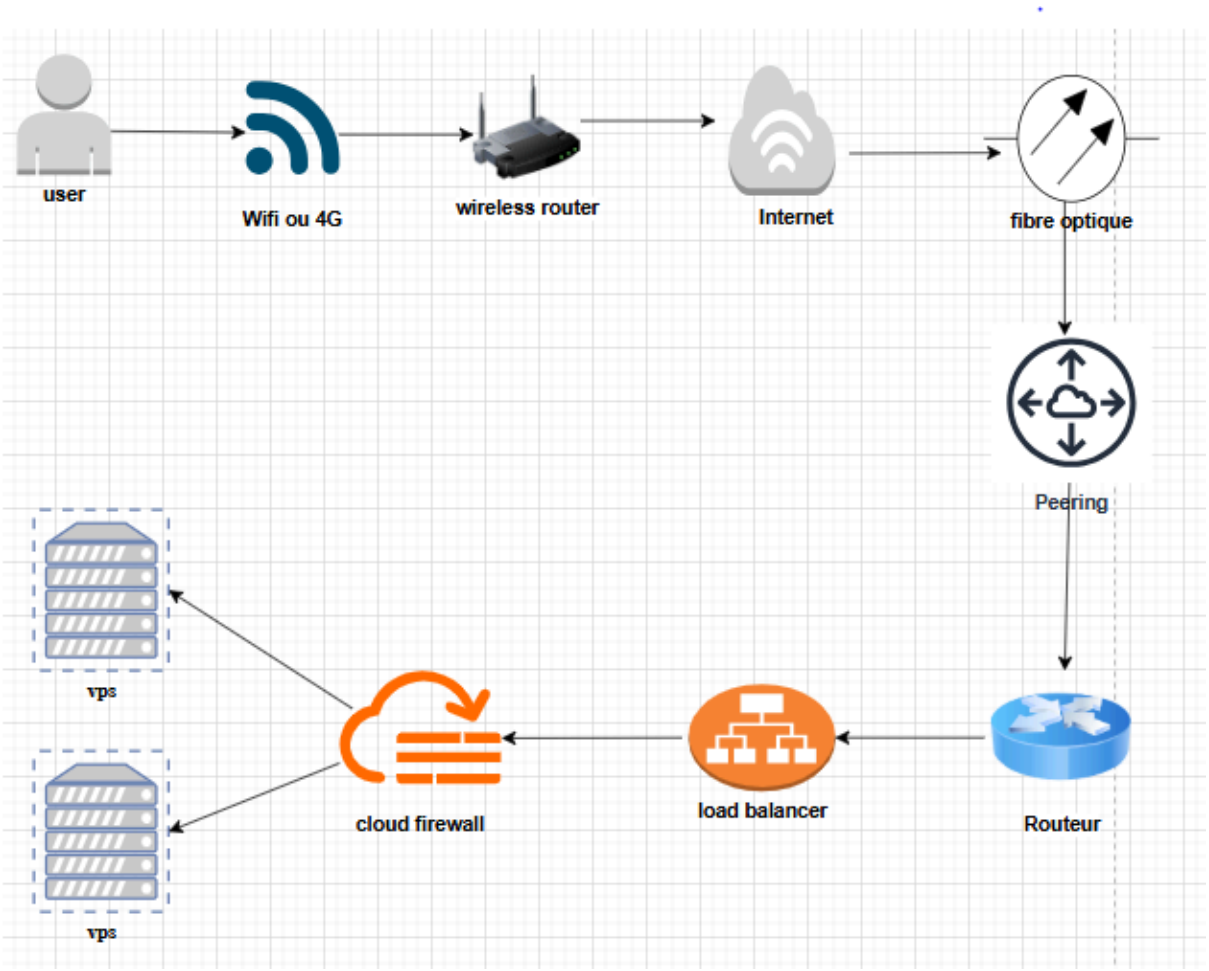


Figure 2 : Architecture de Easy Compta jusqu'au VPS

La liaison Internet entre le **client** et le **serveur** se fait généralement à travers plusieurs étapes de réseau, de l'Internet public vers le VPS, qui est hébergé dans un des centres de données de DigitalOcean. Voici comment la liaison Internet fonctionne :

- **Client (Utilisateur)** : Le client peut être un utilisateur situé à n'importe quel endroit dans le monde, utilisant un appareil connecté à Internet (ordinateur, smartphone, etc.);
- L'utilisateur se connecte à son fournisseur d'accès à Internet (FAI), qui a une connexion à Internet (Wifi ou 4G). Le FAI est responsable de diriger le trafic vers l'Internet global;
- Le trafic passe par le routeur sans fil;
- Le trafic est ensuite transporté via des câbles à fibre optique pour une transmission rapide et fiable;
- Passage par les Points d'Échange Internet et Peering (Internet vers DigitalOcean) : Les points d'échange Internet sont des infrastructures qui permettent aux différents

fournisseurs d'accès à Internet (FAI) et réseaux de se connecter et d'échanger du trafic. Le trafic du client passe par ces points pour atteindre DigitalOcean. DigitalOcean est directement connecté aux fournisseurs d'accès à Internet par des liaisons de peering direct. Cela permet à DigitalOcean de recevoir des données de manière rapide et efficace en passant par des intermédiaires;

- Connexion au Data Center de DigitalOcean : Lorsque les paquets arrivent chez DigitalOcean, ils entrent dans les **routeurs** du centre de données. Ces routeurs acheminent le trafic vers le centre de données qui héberge le VPS spécifique. une fois dans le centre de données, DigitalOcean utilise son propre réseau interne pour diriger le trafic vers le VPS (serveur virtuel privé) spécifique en fonction de l'adresse IP publique de ce serveur;
- **Load Balancer** : Assure la répartition les requêtes entre plusieurs VPS pour éviter la surcharge;
- DigitalOcean propose des **Cloud Firewalls** pour filtrer les accès avant le VPS.

En résumé, la liaison Internet du VPS chez DigitalOcean passe par une infrastructure publique (l'Internet), avec des points de transit (peering, IXP), puis entre dans le réseau privé de DigitalOcean pour atteindre le serveur. Le trafic est dirigé à travers des routeurs et, une fois arrivé sur le VPS, il est traité en fonction des services configurés.

Voici les **flux de données** pour décrire le mouvement des informations à travers le réseau :

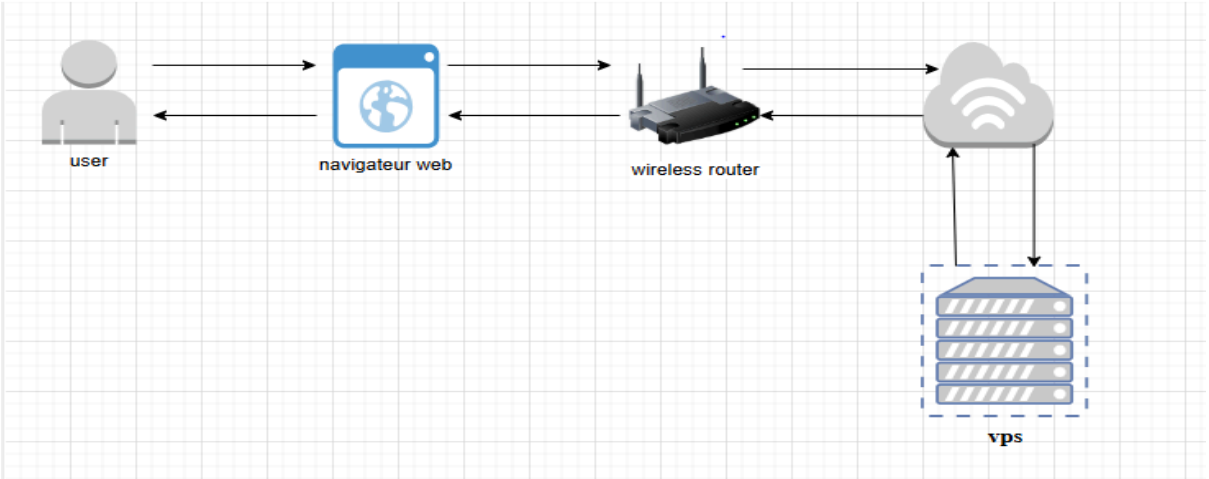


Figure 3 : Flux de données

L'utilisateur accède aux services via un navigateur web.

- **Navigateur Web** : c'est l'application utilisée par l'utilisateur pour interagir avec **Easy Compta**.Il envoie des requêtes HTTP/HTTPS vers le serveur distant (vps) et affiche les réponses reçues.
- **Wireless Router (Routeur sans fil)** : il permet à l'utilisateur de se connecter à Internet via un réseau sans fil.Il assure la communication entre le navigateur web et le serveur distant en acheminant les requêtes et les réponses.
- **Internet** : c'est l'infrastructure réseau mondiale permettant la communication entre différents appareils connectés.Il assure la transmission des données entre le routeur WiFi et le serveur VPS.
- **VPS (Serveur Privé Virtuel)** : c'est un serveur distant hébergé dans un centre de données.Il héberge l'application web **Easy Compta** accessible via Internet.Il reçoit les requêtes du navigateur web et y répond en voyant des pages web,

2.5 Conclusion

L'état des lieux réalisé dans ce chapitre a permis de mettre en lumière les vulnérabilités critiques qui peuvent compromettre la sécurité de l'application web **Easy Compta**. Les analyses menées ont révélé des insuffisances notables dans les mécanismes de protection existants. Cette étude a mis en évidence les vulnérabilités existantes, qu'elles soient liées à

des lacunes dans la configuration de l'application, des failles dans le code, ou des insuffisances dans les mécanismes de protection en place.

Le chapitre suivant, dédié à la **conception et aux études comparatives**, s'attachera à proposer des mécanismes de protection conformes aux meilleures pratiques en matière de sécurité des applications web. Une attention particulière sera accordée au choix des outils, technologies et méthodologies en fonction de leur efficacité et de leur compatibilité avec les besoins d'**Easy Compta**. Cette démarche comparative permettra de sélectionner les solutions les plus pertinentes pour garantir la confidentialité, l'intégrité et la disponibilité des données.

CHAPITRE 3: CONCEPTION ET ÉTUDE COMPARATIVE DES OUTILS D'ANALYSE DE SÉCURITÉ

3.1 Introduction

Après avoir compris les enjeux de sécurité dans l'état des lieux, ce chapitre se concentre sur la conception des solutions adaptées pour protéger l'application **Easy Compta** . Nous explorons ici les différentes approches possibles et procédons à des études comparatives pour choisir les mécanismes de sécurité les plus efficaces et adaptés à notre contexte.

Dans un premier temps, nous comparons les outils et techniques disponibles pour la mise en œuvre des solutions proposées, en prenant en compte des critères tels que l'efficacité, la faisabilité et la compatibilité avec les technologies utilisées dans **Easy Compta**. Ensuite, nous décrirons les principes de conception adoptés pour renforcer la sécurité de l'application.

Ce chapitre vise à poser les bases d'une architecture sécurisée et à justifier les choix technologiques et stratégiques adoptés dans la mise en œuvre des solutions.

3.2 Etudes comparatives

Cette section vise à comparer plusieurs outils ou technologies similaires en fonction de critères spécifiques. L'objectif est de mettre en évidence les avantages et inconvénients de chaque solution, en tenant compte des besoins de l'organisation ou du projet. L'étude comparative permet de sélectionner la solution la plus adaptée à votre environnement technique et opérationnel.
L'objectif principal de cette étude est de **justifier le choix de l'outil** en comparant plusieurs options disponibles. Cette analyse se base sur des critères techniques, des performances, des coûts, de la facilité de mise en œuvre, et de la compatibilité avec l'infrastructure existante.

3.2.1 Outils de scan réseau

3.2.1.1 Nmap

Dans l'univers de la cybersécurité, la connaissance approfondie de la topologie du réseau est cruciale pour défendre efficacement les entreprises contre les menaces numériques en constante évolution. C'est là que NMAP (Network Mapper) entre en jeu en tant qu'outil incontournable pour la cartographie des réseaux informatiques. NMAP est bien plus qu'un simple scanner de ports ; il offre une panoplie de fonctionnalités sophistiquées qui permettent aux professionnels de la cybersécurité et aux administrateurs réseaux de comprendre la structure, les vulnérabilités potentielles et les services exposées de leur réseau.

NMAP (Network Mapper) est un **outil de sécurité informatique open-source** qui permet de découvrir et d'analyser les hôtes et les services présents sur un réseau [W6]. Développé par **Gordon Lyon** (alias Fyodor) dans les années 1990, NMAP est devenu un pilier de l'arsenal des **professionnels de la cybersécurité** et des **administrateurs réseaux**.

Son principal objectif est de **fournir des informations détaillées** sur la topologie du réseau, les services en cours d'exécution sur les hôtes, les ports ouverts, les systèmes d'exploitation utilisés et les vulnérabilités potentielles. NMAP utilise des **techniques d'analyse avancées**, telles que le scanning de ports, l'OS fingerprinting (identification du système d'exploitation) et le script scanning, pour effectuer des sondages approfondis sur le réseau.

Le fonctionnement de Nmap :

Le fonctionnement de NMAP est basé sur l'envoi de paquets réseaux personnalisés à des hôtes cibles et l'analyse des réponses reçues [W6]**. L'outil peut être utilisé en ligne de commande sur différentes plateformes (Windows, Linux, macOS) et offre une multitude d'options de configuration pour répondre aux besoins spécifiques des utilisateurs.**

Parmi les principales fonctionnalités de NMAP, on trouve :

- **Scanning de ports :** Le scanning de ports est l'une des fonctionnalités les plus connues de NMAP. Il permet de découvrir les ports ouverts sur les hôtes cibles, indiquant ainsi les services qui sont en cours d'exécution et accessibles depuis l'extérieur. Cette fonctionnalité est essentielle pour identifier les points d'entrée

potentiels pour des attaques, ainsi que pour vérifier la configuration des pare-feux et des routeurs;

- **OS Fingerprinting :** NMAP peut également effectuer un OS Fingerprinting (identification du système d'exploitation) pour déterminer le système d'exploitation utilisé par les hôtes cibles. En analysant les réponses aux paquets envoyés, NMAP peut fournir des informations sur le système d'exploitation, la version du noyau et d'autres détails pertinents. Cette fonctionnalité est utile pour comprendre la composition du réseau et identifier les vulnérabilités spécifiques aux différents systèmes d'exploitation;
- **Script scanning** : L'une des fonctionnalités les plus puissantes de NMAP est le script scanning. NMAP propose une bibliothèque de scripts pré-écrits et personnalisables qui permettent d'effectuer des **tâches d'analyse avancées**. Ces scripts peuvent détecter les vulnérabilités spécifiques, recueillir des informations sur les services, réaliser des analyses de sécurité et bien plus encore. Les utilisateurs peuvent également créer leurs **propres scripts** pour adapter NMAP aux besoins spécifiques de leur environnement;
- **Détection de pare-feu** : NMAP est capable de **détecter les pare-feux** en analysant les réponses des hôtes cibles aux paquets envoyés. Cette fonctionnalité permet d'identifier les **règles de filtrage** en place et de déterminer si des ports sont ouverts ou bloqués par les pare-feux. Cela peut aider à mieux comprendre la configuration du pare-feu et à identifier d'éventuelles failles de sécurité;
- **Cartographie du réseau** : En combinant les informations obtenues à partir du scanning de ports et de l'OS fingerprinting, NMAP peut créer une c**artographie complète** du réseau. Cette cartographie montre les hôtes connectés, les relations entre les différents éléments du réseau, les services exposés et les connexions entre les hôtes. Cette vue d'ensemble permet de mieux comprendre la topologie du réseau et d'identifier les **points d'accès éventuels** pour des attaques;
- **Analyse de paquets** : NMAP peut également effectuer une analyse approfondie des paquets en examinant leur contenu. Cette fonctionnalité permet de détecter des **anomalies** ou des **signes d'activités malveillantes** dans le trafic réseau, ce qui peut être utile pour identifier des attaques en cours;
- **Scan de vulnérabilités** : Bien que NMAP ne soit pas un scanner de vulnérabilités complet, ses fonctionnalités de **script scanning** permettent d'identifier certaines

vulnérabilités connues. En utilisant des scripts spécialisés, NMAP peut détecter des **vulnérabilités courantes** telles que Heartbleed, Shellshock et bien d'autres.

Les avantages de Nmap :

Tout d'abord, NMAP permet d'obtenir une vision claire de la topologie du réseau, en identifiant les hôtes connectés, les ports ouverts et les services en cours d'exécution. Cette visibilité accrue aide à **détecter les vulnérabilités potentielles** et à renforcer la sécurité du réseau.

En outre, NMAP peut être utilisé pour effectuer des analyses ponctuelles ou régulières afin de suivre les **changements dans le réseau** au fil du temps. Cela permet de détecter rapidement tout nouvel équipement ou service potentiellement non autorisé.

L'utilisation de scripts personnalisés dans NMAP permet également d'adapter l'outil aux besoins spécifiques de l'entreprise, en analysant des aspects particuliers du réseau et en collectant des **informations pertinentes** pour la prise de décision.

Les limites de Nmap :

Malgré ses avantages, NMAP présente également **certaines limites**. Tout d'abord, certaines organisations peuvent considérer les scans effectués par NMAP comme intrusifs ou inappropriés, car ils pourraient **perturber le fonctionnement normal** du réseau. Par conséquent, il est essentiel de bien comprendre les politiques de sécurité en place et d'**obtenir les autorisations nécessaires** avant de lancer un scan avec NMAP.

De plus, NMAP ne garantit pas une **détection exhaustive** de toutes les vulnérabilités présentes sur le réseau. Certaines vulnérabilités pourraient rester cachées ou ne pas être détectées par des **scans standard**. C'est pourquoi il est important d'utiliser NMAP en combinaison avec d'autres outils de sécurité pour une approche plus complète de la cybersécurité.

Enfin, l'interprétation des résultats des scans de NMAP nécessite des **compétences en analyse** et en **interprétation des données**. Les utilisateurs doivent être en mesure de comprendre les informations fournies par NMAP pour prendre des décisions éclairées en matière de sécurité.

3.2.1.2 Masscan

Masscan est le scanner de port réseau le plus rapide [W8]. Il peut scanner l'ensemble d'Internet en moins de 6 minutes avec une vitesse de transmission de données de 25 millions par seconde. Est-il plus rapide que le flash ?
Ce scanner de port le plus rapide donne une sortie similaire à nmap mais Massacan fonctionne comme unicornscann, Zenmap en interne (analyse asynchrone). Il est plus rapide en raison de sa flexibilité qui permet des plages et des plages de ports arbitraires. Masscan utilise sa propre pile TCP/IP personnalisée. Toute opération autre qu'une simple analyse de port peut entraîner un conflit avec la pile TCP/IP locale.
Les principales caractéristiques de Masscan sont :

- **Évolutif** : La caractéristique la plus importante de Masscan est probablement sa capacité à transmettre jusqu'à 10 millions de paquets par seconde grâce à son architecture asynchrone;
- **Portabilité** : Le logiciel peut être compilé et exécuté sur les trois principaux systèmes d'exploitation : Windows, MacOS et Linux;
- **Vérification des bannières** : en plus d'effectuer simplement des analyses de ports, l'outil peut également établir des connexions TCP pour récupérer des informations de base sur les bannières;
- **Compatibilité Nmap** : Masscan a été développé dans le but de rendre l'utilisation et le rendu de l'outil aussi similaires que possible à ceux de Nmap. Cela permet aux utilisateurs de traduire rapidement leurs connaissances Nmap.

Bien qu'il existe une grande compatibilité avec Nmap, il existe quelques différences qui méritent d'être mentionnées [W9] et qui séparent Masscan de Nmap :

- Nmap utilise le mode d'analyse synchrone qui est très lent mais précis, mais Masscan utilise le mode d'analyse asynchrone qui est très rapide mais pas si précis;
- Masscan ne vérifie pas si l'hôte est actif ou non, il traite toujours l'hôte comme en ligne mais Nmap vérifie toujours si l'hôte est actif ou non;
- Masscan ne convertit jamais les noms de domaine en adresses IP. Nous devons trouver manuellement l'adresse IP et la fournir à Masscan. Mais nmap fonctionne bien avec les noms IP et de domaine;

- Nous devons toujours spécifier les ports sur Masscan.

3.2.1.3 Unicornscan

Unicornscan est un outil d'analyse de ports très apprécié qui propose des centaines de fonctionnalités individuelles [W10]. Les scanners de ports sont des outils indispensables lorsque vous démarrez une enquête de sécurité sur un réseau local ou distant. Le scanner de ports unicornscan, une tentative de pile TCP/IP distribuée en mode utilisateur, est conçu pour fournir aux utilisateurs une meilleure interface pour initier des stimuli et mesurer ensuite les réponses des réseaux ou périphériques compatibles TCP/IP.

Unicornscan est le plus souvent utilisé par les administrateurs réseau et système et les programmeurs. Cet outil est essentiel pour analyser les serveurs et les hôtes afin de voir quels ports disponibles sont utilisés pour les communications réseaux.

Fonctionnalités de Unicornscan

Bien qu'il dispose de centaines de fonctionnalités [W11], certaines de ses fonctionnalités clés incluent :

- Analyse TCP asynchrone sans état avec chacun des indicateurs TCP ou des combinaisons d'indicateurs;
- Capture de bannière asynchrone pour l'empreinte digitale des applications et du système d'exploitation;
- Analyse UDP spécifique au protocole asynchrone;
- Détection active et passive du système d'exploitation et des applications à distance;
- Enregistrement et filtrage des fichiers PCAP;
- Sortie de base de données relationnelle pour stocker les résultats de vos analyses;
- Prise en charge de modules personnalisés afin que les pentesters puissent l'adapter à leurs besoins spécifiques;
- Vues d'ensembles de données personnalisés.

L'une des principales caractéristiques d'unicornscan qui le distingue de nmap et des autres scanners de ports est qu'il possède sa propre pile TCP/IP. Les autres scanners de ports utilisent tous la pile TCP/IP du système d'exploitation hôte sous-jacent. Cela permet à unicornscan d'analyser beaucoup plus rapidement que les autres car il peut, par exemple, envoyer des paquets SYN avec un thread et recevoir les réponses avec un autre thread. Cela peut faire une énorme différence lors de l'analyse de très grands réseaux en tant que chercheur en sécurité/pentesteur où nous pouvons analyser des milliers d'adresses IP et être

encore plus important pour un attaquant qui peut analyser des millions d'adresses à la recherche d'un port ouvert ou d'une vulnérabilité particulière. De plus, comme il possède sa propre pile TCP/IP, il est capable d'envoyer des paquets avec des empreintes digitales de système d'exploitation différentes de celles du système d'exploitation de votre hôte. Cela peut être très utile pour masquer votre identité, en particulier combiné à l'usurpation d'adresse IP.

3.2.1.4 Synthèse de l'étude comparative des outils de scan réseau

Le tableau suivant donne une vue d'ensemble sur les principales caractéristiques des outils de scan de sécurité choisis.

Tableau 1 : Comparaison entre les outils de scan réseau

Critères	**Nmap**	**Masscan**	**Unicornscan**
Type	Scanner de ports et réseau	Scanner de ports ultra-rapide	Scanner de ports haute performance
Licence	Open Source (GPL)	Open Source	Open Source
Vitesse	Moyenne à rapide (1 000 ports/seconde à 100 000 ports/seconde)	Extrêmement rapide (millions de ports/seconde)	Très rapide (200 000 à 500 000 paquets/seconde)
Technique de scan	TCP Connect, SYN scan, UDP scan, OS detection, NSE scripting	SYN Scan uniquement	TCP SYN Scan, UDP Scan
Fiabilité des résultats	Très élevée (analyse détaillée)	Moyenne (ne vérifie pas les services, juste les ports ouverts)	Moyenne (avec des faux positifs)
Usage principal	Scan détaillé, test de sécurité, pentest	Scan massif de très grandes plages IP	Scan rapide à grande échelle, analyse réseau
Facilité d'utilisation	Moyenne (options nombreuses mais bien documentées)	Facile (mais nécessite des ajustements pour être précis)	Moyenne (moins connu que Nmap)

Critères	Nmap	Masscan	Unicornscan
Plateformes supportées	Windows, Linux, macOS	Linux	Linux, BSD

Après avoir étudié ces différents outils de scanner de réseau (Tableau 1), notre choix s'est porté sur **Nmap.**

Nmap offre des résultats plus précis et détaillés grâce à ses nombreuses techniques d'analyse. Il permet d'éviter les faux positifs, contrairement à **Masscan** et **Unicornscan**, qui privilégient la vitesse au détriment de la précision. **Nmap** ne se limite pas à un simple scan de ports : il peut détecter **les services actifs, leurs versions et même le système d'exploitation** des machines scannées. **Masscan**, par exemple, ne fait que lister les ports ouverts sans identifier les services derrière.

Il supporte plusieurs types de scans : TCP, UDP, ICMP, SYN, connect(), et bien d'autres. Possibilité d'utiliser des **scripts avancés (NSE – Nmap Scripting Engine)** pour détecter des vulnérabilités, extraire des informations ou automatiser des tests de sécurité.

Unicornscan est moins flexible en termes de fonctionnalités avancées.

Nmap permet d'ajuster la vitesse du scan en fonction du contexte :

- Mode **agressif** pour des scans rapides;
- Mode **discret** pour échapper aux systèmes de détection (IDS/IPS).

Masscan est certes plus rapide, mais il manque de finesse et peut facilement déclencher des alertes de sécurité.

Nmap est l'un des outils de sécurité les plus documentés et utilisés par les **pentesters, administrateurs réseaux et chercheurs en cybersécurité**. Fonctionne sous **Windows, Linux et macOS**, tandis que **Masscan** est limité à Linux et **Unicornscan** est principalement conçu pour Linux et BSD.

3.2.2 Outils de scan de vulnérabilités

Un **scanner de vulnérabilités** est un outil permettant d'**analyser un système informatique, un réseau ou une application** afin d'identifier des failles de sécurité exploitables par des

attaquants. Il automatise la détection des **ports ouverts, services, configurations faibles et vulnérabilités connues (CVE)**.

3.2.2.1 OWASP ZAP

Zed Attack Proxy (ZAP) de Checkmarx est un outil de test de pénétration gratuit et open source. ZAP est conçu spécifiquement pour tester les applications Web et est à la fois flexible et extensible.

À la base, ZAP est ce que l'on appelle un « proxy manipulateur intermédiaire »[W13]. Il se situe entre le navigateur du testeur et l'application Web (Figure 4) afin de pouvoir intercepter et inspecter les messages envoyés entre le navigateur et l'application Web, modifier le contenu si nécessaire, puis transmettre ces paquets à la destination. Il peut être utilisé comme une application autonome et comme un processus démon.



Figure 4 : ZAP entre le navigateur et l'application

S'il existe un autre proxy réseau déjà utilisé, comme dans de nombreux environnements d'entreprise, ZAP peut être configuré pour se connecter à ce proxy (Figure 5).



Figure 5 : ZAP associé à un network proxy

Les principales fonctionnalités [W12] de OWASP ZAP sont :

- **Analyse passive** : Surveille le trafic pour détecter les failles sans attaquer activement l'application;

- **Analyse active :** Réalise des scans d’intrusion pour identifier les vulnérabilités potentielles;
- **Automatisation des scans :** ZAP peut être intégré dans les pipelines CI/CD pour des analyses automatisées;
- **Interface graphique et ligne de commande :** Accessibilité avec une interface intuitive et des options en ligne de commande pour les utilisateurs avancés;
- **Exploration (Spidering) :** L'exploration, autrement dénommée Spidering, est un avantage majeur d'OWASP ZAP. Ce processus balaye le site web dans le but d'y déceler tous les éléments de contenu et de fonctionnalités qu'il contient. Cette exploration est indispensable pour assurer une évaluation sécuritaire intégrale, en ce sens qu'elle permet à ZAP de couvrir tous les pans du site;
- **Tests d'Intrusion (Fuzzing) :** Les tests d'intrusion, connus sous le terme de Fuzzing, désignent une méthode sécuritaire qui consiste à envoyer des données erronées ou sans forme établie vers le site web pour observer ses réactions. Cette approche peut permettre de déceler des failles qui passeront à travers les mailles d'autres techniques de vérification;
- **Authentification :** OWASP ZAP bénéficie de multiples méthodes d'authentification, parmi lesquelles l'authentification à l'aide de formulaires ou encore l'authentification HTTP. Cela permet à ZAP d'accéder à tous les espaces du site, y compris ceux d'accès restreints;
- **Extensions :** OWASP ZAP a la particularité d'être évolutif, autrement dit, vous pouvez lui rajouter des fonctionnalités en plus grâce à l'installation des add-ons. Un vaste choix de ces modules supplémentaires vous est proposé, notamment pour l'évaluation sécuritaire avancée, l'automatisation des essais ou encore la synergie avec les autres utilitaires sécuritaires.

Les avantages de l’Utilisation de OWASP ZAP sont :

- Détection précoce des vulnérabilités dans les applications web;
- Solution open-source avec un large support communautaire;
- Automatisation et intégra²tion CI/CD pour des scans de sécurité en continu;
- Interface intuitive pour les débutants et puissante pour les experts;

- Rapports de sécurité détaillés avec des recommandations d'atténuation.

Les inconvénients de OWASP ZAP sont :

- Peut ralentir le serveur cible si des tests intensifs sont effectués en mode actif;
- **Les scans peuvent être lents**, surtout pour des applications web complexes avec de nombreuses pages dynamiques;
- Interface Utilisateur Peu Intuitive;
- Moins efficace pour des **tests avancés d'intrusion** et les attaques complexes comme le fuzzing avancé.

3.2.2.2 Burp Suite

Burp Suite est un logiciel spécialisé pour les audits de sécurité web (pentests web) [W14]. Il est développé par l'entreprise PortSwigger qui est une référence dans le monde de la sécurité web.Burp Suite, appelé le plus souvent seulement Burp, est optimisé et pensé pour répondre aux besoins des pentesters professionnels et est d'ailleurs l'outil le plus utilisé dans son domaine. C'est un outil modulaire permettant de réaliser à la fois des tests manuels et automatisés qui vont aider le pentester à identifier efficacement des vulnérabilités sur les applications web.

Les principales fonctionnalités [W14] de Burp Suite sont :

- **Le Proxy :** La fonctionnalité principale de Burp Suite est le Proxy. Le Proxy va permettre à Burp de jouer le rôle d'intermédiaire entre le client (navigateur web) et le serveur hébergeant l'application web;
- **Le Répéteur :** Le Repeater (Répéteur) est le module qui va permettre de rejouer des requêtes à volonté. Comme son nom l'indique, le pentester va pouvoir répéter des requêtes et les modifier à sa guise avant de les envoyer au serveur;
- **Le scanner de vulnérabilités :** Le Scanner est une des fonctionnalités les plus utiles de l'édition « Professional » de Burp Suite;
- **Collaborateur :** Le Collaborateur de Burp va permettre au pentester d'avoir une plateforme tierce sur Internet, accessible depuis n'importe où;
- **Intrus** : automatisez les attaques personnalisées pour tester les défenses des applications contre les techniques d'attaque courantes;

- **Séquenceur** : évaluez le caractère aléatoire et la prévisibilité des jetons de session et des valeurs aléatoires;
- **Extensibilité** : personnalisez et améliorez les fonctionnalités via des API pour des tests de sécurité sur mesure.

Les avantages de Burp Suite :
Lors d'un pentest web, un pentester va généralement utiliser Burp Suite 90% du temps, ce qui en fait un outil incontournable. Il existe certaines alternatives comme le logiciel ZAP, développé par l'OWASP mais Burp Suite reste la référence dans le monde du pentest web.Le fonctionnement global de Burp est pensé de façon modulaire. Certains de ses modules sont installés par défaut sur le logiciel. Ce sont les modules essentiels à la conduite d'un audit. D'autres modules complémentaires nommés extensions sont disponibles au téléchargement via l'extender (le « catalogue » de Burp).En effet, la force de Burp Suite, outre sa modularité et son ergonomie, réside dans sa communauté très active qui développe de nouvelles extensions et crée de la documentation détaillée sur les modules.

Burp est un outil complet qui saura répondre à la majorité des besoins d'un pentester. Même si dans le cas où les fonctionnalités de base ne répondent pas à un besoin spécifique, il existe dans la plupart des cas, une extension développée par la communauté et prévue à cet effet.

Les inconvénients de Burp Suite sont :

- Coût Élevé (Version Pro);
- Consommation Élevée de Ressources;
- Interface Complexe pour les Débutants;
- Scans Lents
- Moins Adapté aux Applications Modernes (SPA)

3.2.2.3 Nessus

Nessus est un scanner de vulnérabilités édité par l'entreprise Tenable Network Security depuis 1998. Il permet de **détecter les failles de sécurité** présentes dans la plupart des assets IT. Cela comprend le hardware, les applications web, les infrastructures cloud et tout asset exposé à Internet. Nessus est compatible avec de nombreux systèmes d'exploitation.

Les particularités de Nessus :

Nessus intègre différentes fonctionnalités très utiles [W15]. Le regroupement de vulnérabilités au sein d'une **vue groupée** en est un exemple. Il est possible de présenter des problèmes similaires ou des catégories de vulnérabilités dans un même thread. Les vues groupées facilitent la lecture des résultats des scans, ainsi que la priorisation des remédiations.

Les **résultats en direct** constituent une autre fonction phare de Nessus. À chaque mise à jour d'un plugin utilisé dans une analyse précédente, le scanner effectue une vérification automatique. Il se base sur les données provenant de l'historique de scans pour réaliser une recherche de vulnérabilité hors ligne.

L'utilisateur sait ainsi quand il est utile de lancer un scan pour confirmer la présence ou non de la faille en question. Plus largement, Nessus est un scanner de vulnérabilités relativement facile à utiliser. Son interface intuitive associée à ses **plugins dynamiquement compilés** permet de gagner beaucoup de temps lors d'analyses d'assets IT.

Les avantages de Nessus :

Lorsque vous naviguez dans les eaux périlleuses de l'analyse des vulnérabilités du réseau, il est essentiel de comprendre pourquoi Nessus se démarque des autres [W16]. Explorons les principaux avantages qui distinguent Nessus.

- **Ne fait aucune supposition sur la configuration du serveur;**
- **Hautement extensible :** Nessus fournit une architecture de plug-ins et un langage de script appelé NASL;
- **Informations actualisées sur les vulnérabilités;**
- **Open source :** Nessus est un logiciel open source, son utilisation est donc gratuite et le code source est accessible au public;
- **Assistance à la mise à jour corrective :** lorsqu'une vulnérabilité est détectée, Nessus fournit des informations détaillées et recommande des solutions pour corriger le problème.

Les inconvénients de Nessus sont :

- Pas Idéal pour les Tests de Pénétration Avancés;
- Fort Consommateur de Ressources;
- Dépendance aux Mises à Jour;

- Détection Passive Limitée.

3.2.2.4 Synthèse de l'étude comparative des outils de scan des vulnérabilités

Le tableau suivant donne une vue d'ensemble sur les principales caractéristiques des outils de scan de vulnérabilités choisis.

Tableau 2 : Comparaison entre les outils de scan de vulnérabilités

Critères	**OWASP ZAP**	**Burp Suite**	**Nessus**
Type d'outil	Open source, gratuit	Payant (avec version limitée gratuite à 30 jours)	Payant (version d'essai gratuite)
Précision et faux positifs	Taux de faux positifs faible	Bonnes capacités d'analyse, faible taux de faux positifs	Bonnes capacités d'analyse
Facilité d'utilisation	Interface assez simple	Interface professionnelle, nécessite une courbe d'apprentissage	Facile à utiliser, orienté réseau
Coût	Gratuit	Version gratuite limitée, version pro payante	Version gratuite limitée, version pro payante
Rapports générés	Rapports détaillés mais peu personnalisables	Rapports très détaillés, personnalisables	Rapports détaillés et analytiques
Compatibilité	Prend en charge plusieurs plateformes et langages	Compatible avec diverses plateformes	Scanner les infrastructures réseau et les applications web
Limites	Moins adapté aux tests de performance	Nécessite une configuration manuelle avancée pour des tests poussés	Plus orienté réseau que web

À la suite de l'étude comparative des différents outils de scan des vulnérabilités (Tableau 2), nous avons choisi **OWASP ZAP** comme solution principale pour faire le scan de notre application. Contrairement aux outils comme **Burp Suite Pro** ou **Nessus** , qui auraient dû avoir une licence pour débloquer toutes leurs fonctionnalités payantes, **OWASP ZAP** est totalement gratuit et open source . Il est donc idéal pour les entreprises et les pentesters indépendants qui veulent un scanner de vulnérabilités sans coût supplémentaire.
Lorsque ZAP parcourt votre application Web, il construit une carte des pages de votre application Web et des ressources utilisées pour afficher ces pages. Il enregistre ensuite les requêtes et les réponses envoyées à chaque page et crée des alertes en cas de problème potentiel avec une requête ou une réponse.
Contrairement à certains scanners qui peinent avec les **Single Page Applications (SPA)** , OWASP ZAP supporte bien **JavaScript, AJAX et les API REST** , ce qui le rend adapté aux applications modernes développées avec **React, Angular, Vue.js** .

3.2.3 Solutions d'hébergement

Les solutions d'hébergement web permettent d'héberger des sites internet et des applications. Elles offrent généralement un espace serveur, des ressources (CPU, RAM) et des services associés.

3.2.3.1 Hébergement VPS

Un serveur privé virtuel, également appelé VPS, agit comme un environnement virtuel isolé sur un serveur physique détenu et géré par un fournisseur de services cloud ou d'hébergement Web [W3]. L'hébergement VPS utilise une technologie de virtualisation pour scinder une unique machine physique en plusieurs environnements de serveur privé partageant ses ressources.
Un VPS, ou serveur privé virtuel, est une machine virtuelle qui fournit des ressources de serveur virtualisées sur un serveur physique partagé avec d'autres utilisateurs. Avec l'hébergement VPS, vous bénéficiez d'un espace serveur dédié avec une quantité réservée de ressources, ce qui vous offre plus de possibilités de contrôle et de personnalisation qu'un hébergement partagé.

Le fonctionnement du vps :

Même si vous partagez un serveur physique avec d'autres utilisateurs, un VPS simule un environnement d'hébergement de serveur dédié. Votre fournisseur d'hébergement installe un hyperviseur (une couche virtuelle) par-dessus le système d'exploitation (OS) du serveur physique, ce qui lui permet de diviser celui-ci en compartiments virtuels [W2]. Cette couche permet à chacun de ces compartiments d'exécuter son propre système d'exploitation et ses propres logiciels. Ainsi, chaque environnement peut fonctionner indépendamment des autres.

Même si, techniquement, vous partagez des ressources avec d'autres utilisateurs, vos ressources propres sont garanties. Vous payez un montant défini qui est alloué à votre VPS et ne peut pas être utilisé par un autre compte.

Les avantages de l'hébergement VPS :
L'hébergement de serveur privé virtuel offre un certain nombre d'avantages en matière d'isolation et de contrôle des comptes, notamment :

- En dédiant un serveur virtuel à votre site Web, vous bénéficiez d'un contrôle total sur l'environnement du serveur, ce qui vous permet d'ajuster les paramètres et les logiciels du serveur pour répondre parfaitement aux besoins de votre site Web;
- L'hébergement VPS est basé sur la virtualisation des serveurs, qui permet de configurer plusieurs serveurs virtuels sur un seul serveur physique;
- L'un des principaux avantages de l'hébergement VPS est l'accès root, qui offre un contrôle complet sur l'environnement du serveur;
- L'hébergement VPS offre un environnement isolé, garantissant que votre site Web n'est pas affecté par les activités d'autres sites Web sur le même serveur.

Les inconvénients de l'hébergement VPS sont :

- La sécurité n'est pas prise en charge par le fournisseur (sauf pour les VPS managés);
- Scalabilité Limitée (Comparé au Cloud);
- Problèmes de Fiabilité en Cas de Panne du Serveur Hôte;
- Complexité de la Sauvegarde et de la Restauration.

3.2.3.2 Serveur dédié

Un serveur Bare Metal, également connu sous le nom de serveur dédié, est un ordinateur physique qui se trouve au sein d'un datacenter et qui est attribué à un client. Chaque client peut configurer et gérer son serveur en fonction de ses besoins spécifiques [W4].

Contrairement au serveur virtuel, qui partage son matériel avec d'autres serveurs virtuels, le serveur dédié fournit exclusivement toutes ses ressources de mémoire, de processeur et de stockage à un client.

Les serveurs dédiés n'exécutent pas d'hyperviseur. Le client profite donc d'un accès complet aux performances et à la capacité matérielle du serveur, qu'il peut utiliser à n'importe quelle fin. L'hébergement de ses propres serveurs ou machines virtuelles est inclus. Enfin et surtout, les performances du serveur ne peuvent pas être affectées par les autres utilisateurs du datacenter.

Les avantages des serveurs dédiés sont :

- Fiabilité : Les utilisateurs des serveurs dédiés ne partagent pas leurs ressources, ce qui signifie que le matériel sert uniquement pour les charges de travail allouées par un utilisateur spécifique;
- Contrôle : Un serveur Bare Metal dédié à un seul utilisateur peut être personnalisé : système d'exploitation, processeurs, nombre de cœurs, GPU, mémoire RAM, type de disque dur, etc;
- Sécurité : Les hyperviseurs ont beau être sécurisés, ils partagent tout de même le matériel et la couche hyperviseur avec d'autres serveurs virtuels. Toute configuration ou erreur de codage pourrait rendre le serveur virtuel vulnérable. Les serveurs dédiés préviennent ce type de menace potentielle, car ils ne sont pas partagés.

Les inconvénients des serveurs Bare Metal sont :

- Coût : Les serveurs Bare Metal dédient l'ensemble de leurs ressources à un seul utilisateur, qu'il les utilise dans leur totalité non;
- **Connaissances informatiques :** Les serveurs Bare Metal sont contrôlés par l'utilisateur. Pour utiliser un serveur dédié, l'utilisateur doit disposer de connaissances techniques suffisantes pour le configurer, pour installer les applications et les logiciels, mais surtout pour s'assurer que les spécifications correspondent aux exigences des charges de travail;
- **Usage des serveurs dédiés :** Les serveurs dédiés offrent aux clients un contrôle total sur les applications et les services qu'ils peuvent exécuter.

3.2.3.3 Hébergement partagé

L'hébergement mutualisé ou partagé est un moyen abordable et convivial d'héberger un site internet, c'est donc une excellente option pour les débutants [W5].

Les avantages de l'hébergement partagé sont :

- Le coût. Puisque le coût est réparti entre de nombreux utilisateurs, l'hébergement mutualisé est la solution d'hébergement la plus abordable.
- **Une gestion facile**. Les fournisseurs d'hébergement mutualisés offrent généralement un panneau de contrôle, vous aidant à configurer et à gérer facilement les sites Web.
- **Une maintenance simplifiée.** Avec l'hébergement mutualisé, l'hôte web s'occupera des aspects techniques, y compris des tâches de maintenance du serveur telles que l'installation et les mises à jour des logiciels.
- La flexibilité. La plupart des fournisseurs d'hébergement mutualisés proposent des plans évolutifs, ce qui vous permet de mettre à niveau les ressources et les fonctionnalités à mesure que le site se développe.

Les inconvénients de l'hébergement partagé sont :
Malgré son caractère abordable et son accessibilité, le service d'hébergement mutualisé présente également quelques inconvénients, tels que :

- Des ressources limitées : Le partage des ressources avec d'autres sites Web signifie un espace de stockage, une bande passante et une puissance CPU limités;
- Pas d'accès root : L'accès root vous donne le plus haut niveau d'accès au serveur;
- Une adresse IP partagée : Outre les ressources, vous partagerez également une adresse IP avec d'autres utilisateurs.

3.2.3.4 Synthèse de l'étude comparative des solutions d'hébergement

Le tableau suivant donne une vue d'ensemble sur les principales caractéristiques des solutions d'hébergement choisies.

Tableau 3 : Comparaison entre les solutions d'Hébergement

Critères	**VPS**	**Serveur dédié**	**Hébergement partagé**
Définition	Serveur virtuel hébergé sur un serveur physique partagé	Serveur physique entièrement dédié à un seul client	Serveur unique partagé entre plusieurs utilisateurs
Coût	Moins cher (idéal pour les	Plus coûteux en raison	Le coût est réduit car les

Critères	VPS	Serveur dédié	Hébergement partagé
	budgets limités)	de la dépréciation des ressources	frais d'exploitation du serveur sont répartis entre tous les utilisateurs
Performances	Performances partagées avec d'autres VPS sur le même hôte	Performances maximales car les ressources sont entièrement dédiées	Limitée, car les ressources sont partagées avec d'autres utilisateurs
Évolutivité	Facilement évolutif (ajout de RAM, CPU ou stockage)	Moins flexible, nécessite souvent des mises à niveau matérielles physiques	Faible : difficile à étendre car l'environnement est limité
Ressources	Partagées avec d'autres utilisateurs sur le même hôte	100% dédié à l'utilisateur	Ressources limitées et imposées par l'hébergeur
Complexité technique	Plus simple à configurer et à gérer pour les petites entreprises	Nécessité plus de compétences techniques pour la gestion	Faible : Accessible aux débutants
Gestion	Souvent géré par l'hébergeur (selon le plan choisi)	Peut être géré ou non, selon le service choisi	L'hébergeur gère entièrement le serveur

Après cette étude (Tableau 3) nous avons choisi le **VPS** car il donne un accès à des ressources garanties et à un contrôle total sans avoir à utiliser votre propre serveur.

Un VPS vous donne un **accès complet à l'administration** du serveur, vous permettant de configurer votre environnement exactement comme vous le souhaitez. Cela est particulièrement utile si vous avez des besoins spécifiques en matière de **logiciels, configurations réseau ou sécurité**.

Un VPS constitue un excellent **compromis entre coût, flexibilité, contrôle et performance**. Il est idéal pour des utilisateurs ayant besoin d'un environnement personnalisé et sécurisé sans avoir à gérer l'infrastructure d'un serveur dédié. Si vous avez des prévisions de

croissance et que vous souhaitez garder un contrôle sur les ressources et la configuration, le VPS est une **solution évolutive et rentable**.

3.2.4 Outils de surveillance

Les **outils de surveillance** sont essentiels pour assurer la performance, la sécurité et la disponibilité des systèmes, réseaux et applications.

3.2.4.1 Nagios

Nagios, ou Nagios Core est un logiciel ordonnanceur qui surveille les systèmes, les réseaux et l'infrastructure. Nagios offre des services de surveillance et d'alerte pour les serveurs, les commutateurs, les applications et les services. Il alerte les utilisateurs en cas d'incidents et les avertit une deuxième fois lorsque le problème a été résolu. Nagios a été conçu à l'origine pour fonctionner sous Linux, mais il fonctionne aussi bien sur d'autres variantes d'Unix [W20].

Nagios a été le premier outil de monitoring IT open source à se positionner en 1999. Il se nommait alors NetSaint. Derrière le logiciel, une véritable communauté s'est créée avec plus d'un million d'utilisateurs à travers le monde.

Les caractéristiques de nagios sont :

- **Surveillance:** Nagios est un logiciel d'analyse réseau léger qui offre la surveillance des applications Web, la surveillance SNMP, la surveillance des journaux et Windows surveillance;
- **Évolutivité et visibilité :** Cet outil d'analyse réseau gratuit offre à votre réseau un haut degré d'évolutivité et de visibilité, vous aidant à résoudre les problèmes liés à plusieurs réseaux;
- **Définition des fichiers individuels :** Nagios permet de lire sa configuration à partir d'un répertoire entier, ce qui peut aider à décider comment définir des fichiers individuels;
- **Intégrations:** Ce logiciel d'analyse réseau s'intègre facilement à OTRS, Request Tracker, xMatters et Jira;
- **Plates-formes supportées:** compatible avec Windows et Linux.

Les avantages de nagios :

Nagios permet :

- la surveillance des équipements et systèmes cibles, à travers notamment des protocoles tels SMTP, POP3, HTTP, NNTP, ICMP, SNMP, SNMP, FTP, SSH ((scripts shell, C++, Perl, Ruby, Python, PHP, C#, etc...);
- la surveillance des réseaux, systèmes d'exploitation et tous types de matériel (comme les sondes de température, les alarmes, etc.) via des scripts communément appelé plugins de supervision;
- L'alerte en cas de de dépassement de seuil ou panne via un système de notification qui prévient l'exploitant du SI ou l'administrateur.

Les inconvénients de nagios sont :

- Configuration Complexe et Peu Intuitive;
- Besoin de Nombreux Plugins pour une Surveillance Avancée;
- Absence de Fonctionnalités Modernes (Monitoring Cloud et Conteneurs);
- Manque de scalabilité pour les grandes infrastructures.

3.2.4.2 Prometheus

Prometheus est un outil open-source dédié au monitoring de performances d'applications et de systèmes informatiques [W19]. Découvrez tout ce que vous devez savoir sur ce logiciel, ses fonctionnalités, ses avantages et les formations permettant d'apprendre à le manier.

Dans tous les secteurs d'activité, de nombreuses entreprises rencontrent le succès grâce aux **métriques**. Ces dernières permettent de suivre et de comprendre les performances d'une application ou d'une infrastructure.

Pour cette raison, les données télémétriques et des bases de données de séries temporelles sont de plus en plus exploitées depuis plusieurs années. L'un des meilleurs outils de ce domaine est Prometheus Monitoring.

Le fonctionnement de prometheus :
Pour obtenir des métriques, Prometheus requiert un **endpoint HTTP exposé**. Le logiciel peut alors commencer le « scraping » de données numériques et la capture sous forme de **séries temporelles**.

Les données sont ensuite stockées dans une base de données locale, spécialement conçue pour les données de **séries temporelles**. Il est aussi possible de coupler des dépôts de stockage distants.

Les utilisateurs peuvent construire des séries temporelles temporaires depuis la source en utilisant des requêtes. Les labels et noms de métriques sont utilisés pour définir ces séries.

Les requêtes s'effectuent avec le langage de **programmation PromQL**, permettant aux utilisateurs de sélectionner et d'agréger les données de séries temporelles en temps réel.

De plus, PromQL peut aussi être utilisé pour créer des conditions d'alerte envoyant des messages aux systèmes externes comme les boîtes email ou les messageries telles que **Slack**.

L'interface utilisateur web de Prometheus permet de présenter les données collectées sous forme de tableau ou de graphique. Il est également possible d'utiliser des **APIs** pour connecter le logiciel à des outils de visualisation tiers comme Grafana.

Les avantages de prometheus sont :

- Système de stockage efficace (Time-Series Database);
- Architecture en mode "pull" (Scalabilité et Flexibilité);
- Langage de requête puissant (PromQL);
- Gestion avancée des alertes (Alertmanager);
- Écosystème riche et nombreuses intégrations.

Les inconvénients de prometheus sont :

- Pas conçu pour la surveillance des logs et des événements;
- Pas idéal pour les infrastructures traditionnelles (réseaux, Windows, etc.) ;
- Gestion limitée des utilisateurs et des accès.

3.2.4.3 UptimeRobot

UptimeRobot est un outil qui permet de surveiller et d'analyser jusqu'à 50 URL de sites gratuitement [W18]. La prise en main et la mise en place d'actions de monitoring sont faciles et rapides.

L'outil propose un monitoring qui vous alerte directement si votre site web tombe en panne et devient momentanément indisponible. Vous pouvez également surveiller vos certificats SSL et recevoir une notification à l'approche de la date d'expiration de ceux-ci (30, 14, et 7 jours

auparavant). La plateforme peut effectuer un monitoring des ports et des ping, une surveillance des travaux Cron ainsi que le tracking de différents mots-clés.

Pour que vous soyez informé facilement lors d'anomalies détectées, UptimeRobot propose en plus des alertes e-mail, SMS et appels, plusieurs intégrations : Twitter, Slack, Zapier, Telegram, Microsoft Teams…

UptimeRobot propose des pages de statut qui permettent de présenter l'état d'un site et de communiquer sur les incidents et maintenances en cours ou terminés, en choisissant les différents facteurs à présenter (état de disponibilité, ratio des jours précédents, temps de réponse…). L'accès aux pages de statut peut être protégé par un mot de passe qu'il suffit de partager aux collaborateurs qui ont besoin d'accéder aux informations.

Les fonctionnalités avancées [W17] de UptimeRobot sont :

- **Partager les mises à jour des incidents :** Envoyez des mises à jour de statut par e-mail aux abonnés de votre page de statut.
- **Pages d'état en marque blanche :** Personnalisez les couleurs, ajoutez votre logo ou même liez votre domaine personnalisé.
- **Surveillance du temps de réponse :** Consultez vos temps de réponse dans un graphique pour révéler les problèmes de performances.
- **Fenêtres de maintenance :** Configurez des fenêtres de maintenance pour éviter les alarmes lors de la maintenance planifiée.
- **Requêtes HTTP personnalisées :** Vous avez besoin de définir un en-tête HTTP avancé ? Nous avons ce qu'il vous faut !
- **Contrôles multi-sites :** Nous vérifions les incidents à partir de plusieurs emplacements pour éviter les faux positifs.

Les avantages de UptimeRobot sont :

- Facilité d'utilisation;
- Surveillance gratuite et abordable;
- Alertes en temps réel;
- Historique et rapports de disponibilité;
- Aucune installation requise.

Les inconvénients de UptimeRobot sont :

- Fonctionnalités limitées par rapport aux outils avancés de surveillance;
- Fréquence de surveillance limitée en version gratuite;

- Alertes SMS limitées dans la version gratuite;
- Pas adapté aux grandes infrastructures.

3.2.4.4 Synthèse de l'étude comparative des outils de surveillance

Le tableau suivant donne une vue d'ensemble sur les principales caractéristiques des outils de surveillance choisis.

Tableau 4 : Comparaison entre les outils de surveillances

Critères	**Prometheus**	**Nagios**	**UptimeRobot**
Type	Outil de surveillance	Outil de surveillance	Outil de surveillance en ligne (SaaS)
Détection des services	Problèmes en temps réel, alertes	Manuelle via des plugins, alertes	Surveillance de disponibilité, alertes
Intégration	Multicloud, Kubernetes	Réseaux, systèmes	Multicloud
Performance	Haute	Moyenne	Basique (centré sur la disponibilité)
Coût	Gratuit, payant	Gratuit, payant	Gratuit (limité), Payant (avancé)
Alerte	Gestionnaire d'alertes intégré (AlertManager)	Courriel, SMS, Slack, Webhooks	Email, SMS, Slack, Webhooks (limité en gratuit)

Après cette étude (Tableau 4) nous avons choisi **UptimeRobot** qui est un service de surveillance de disponibilité basé sur le cloud, conçu pour suivre en temps réel l'état des sites web, serveurs et services en ligne. Il permet aux entreprises et aux développeurs de s'assurer que leurs services restent accessibles en détectant les pannes et en envoyant des alertes immédiatement.

Contrairement à **Prometheus, Nagios et Zabbix**, qui nécessitent une installation et une configuration avancée, **UptimeRobot est une solution SaaS** (basée sur le cloud), ce qui signifie qu'il n'y a rien à installer ni à gérer sur un serveur.

Il est parfait pour **détecter rapidement les pannes** et envoyer des alertes. **Prometheus, Nagios et Zabbix** sont plus complets mais nécessitent des configurations complexes et sont plus adaptés aux infrastructures IT avancées.
Notre objectif principalement est de **vérifier que les services sont en ligne et fonctionnels**, c'est pourquoi notre choix s'est porté sur **UptimeRobot** car il est **léger, rapide à mettre en place et efficace**.

3.3 Maquette de test

Cette maquette de test de sécurité a été conçue pour analyser la robustesse de l'application Easy Compta, hébergée sur un VPS. Elle met en œuvre plusieurs outils d'évaluation de la sécurité et de la disponibilité afin d'identifier les vulnérabilités potentielles et d'assurer un suivi en temps réel.

3.3.1 Présentation des Outils Utilisés

Nmap : Scanner réseau permettant de détecter les ports ouverts et les services exposés sur le serveur hébergeant l'application. Il effectue également une résolution de nom de domaine pour obtenir l'adresse IP du VPS.
OWASP ZAP : Outil d'analyse dynamique de sécurité des applications web, utilisé pour intercepter et analyser les requêtes entre l'utilisateur et le serveur afin d'identifier des failles potentielles.
UptimeRobot : Service de surveillance qui envoie périodiquement des requêtes HTTP(S) à l'application pour vérifier sa disponibilité et détecter d'éventuelles interruptions de service.
VPS : Serveur virtuel privé sur lequel l'application Easy Compta est hébergée.

3.3.2 Fonctionnement de la maquette de Test

Surveillance de la disponibilité avec UptimeRobot
Un robot de monitoring envoie des requêtes HTTP(S) à l'application (Figure 6) via son nom de domaine (easycompta.fewnu.app) pour vérifier si le service est actif.
En cas d'échec, une alerte est générée pour signaler une panne potentielle.
Analyse des ports et services avec Nmap

Nmap résout d'abord le nom de domaine de l'application pour obtenir l'adresse IP du VPS (Figure 6) . Il effectue ensuite un scan des ports ouverts et des services actifs pour identifier d'éventuelles configurations vulnérables.

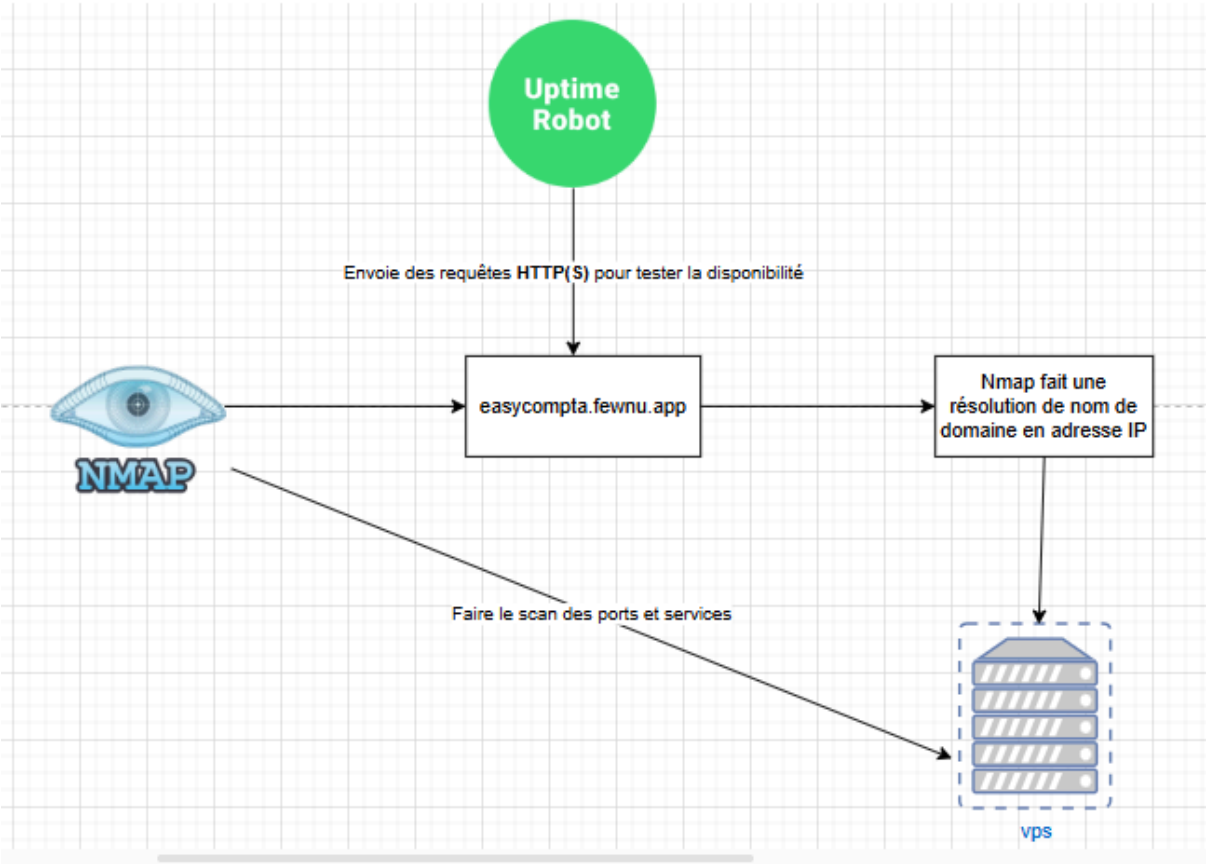


Figure 6 : Maquette de test

Scan avec OWASP ZAP

Un utilisateur interagit normalement avec l'application via une interface web (Figure 7). OWASP ZAP intercepte les requêtes HTTP envoyées par l'utilisateur et analyse leur contenu pour détecter des vulnérabilités applicatives.

Après que ZAP a détecté les vulnérabilités, le pentester intervient (Figure 7) pour vérifier et exploiter les failles manuellement. Le pentester va tester directement sur l'application en fonction des vulnérabilités détectées par ZAP.

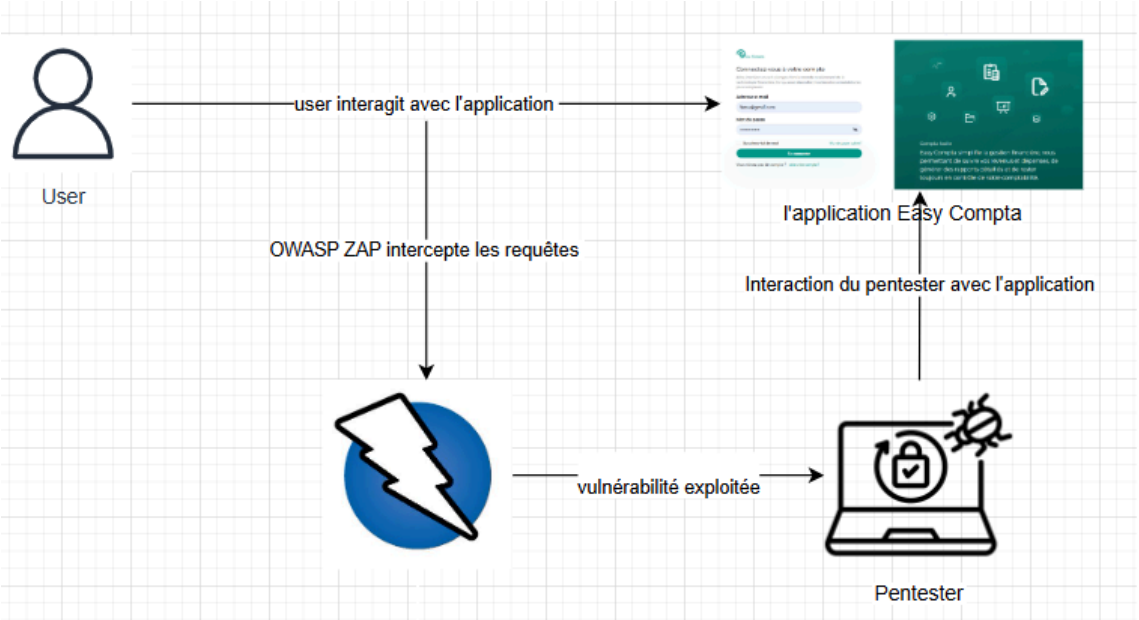


Figure 7 : Maquette de test (suite)

3.3.3 Corrélation des Résultats et Analyse des Vulnérabilités

Une fois les tests effectués :

Les résultats de Nmap permettent d'identifier des services non sécurisés ou mal configurés.

Les analyses d'OWASP ZAP révèlent d'éventuelles failles dans le code et la logique applicative.

Les alertes d'UptimeRobot permettent de surveiller la stabilité et la disponibilité de l'application dans le temps.

3.4 Conclusion

Dans ce chapitre, nous avons effectué une étude comparative des différentes solutions afin de choisir les outils les plus efficaces et nous avons aussi défini une maquette de test qui a permis de structurer le processus de test de sécurité.

Ainsi, cette démarche a permis de dégager une vision complète et intégrée de l'environnement sécurisé nécessaire pour protéger l'application Easy Compta contre les vulnérabilités et les menaces externes.

CHAPITRE 4 : IDENTIFICATION DES VULNÉRABILITÉS EXISTANTES ET ANALYSES DES RISQUES

4.1 Introduction

L’identification des vulnérabilités existantes est un élément clé de la gestion de la sécurité informatique. Les vulnérabilités, qu'elles soient présentes dans les logiciels, les réseaux ou les systèmes matériels, représentent des failles pouvant être exploitées par des attaquants pour compromettre la confidentialité, l'intégrité et la disponibilité des données et services. Avec l'évolution constante des technologies et l'augmentation des cybermenaces, il devient crucial de procéder à une identification régulière des vulnérabilités afin de garantir la résilience des infrastructures et des applications face à ces attaques.

Ce chapitre se concentre sur l’identification des vulnérabilités existantes dans le système étudié. L’objectif est d’identifier et d’évaluer les vulnérabilités qui peuvent affecter la sécurité du système, d’en comprendre les mécanismes d’exploitation potentiels et d’en analyser les impacts. À travers cette analyse, nous visons à fournir une vision claire des risques associés à chaque vulnérabilité, ainsi que des recommandations pour les atténuer ou les corriger.

L’approche adoptée dans ce chapitre repose sur l’utilisation de divers outils d’analyse de vulnérabilités et sur une méthodologie systématique, permettant une évaluation détaillée des différentes failles de sécurité présentes. En fonction de la gravité et de l’impact de ces vulnérabilités, des solutions seront proposées pour renforcer la sécurité globale du système.

4.2 Définition du périmètre d'analyse

La définition du périmètre d'analyse est une étape clé dans la gestion de la sécurité, car elle permet de focaliser les efforts sur les systèmes et composants les plus sensibles tout en précisant ce qui ne sera pas inclus dans l'analyse. Cette section a pour but d'identifier les objectifs spécifiques de l'analyse, de décrire les systèmes cibles, et de définir les zones exclues de l'analyse pour éviter des malentendus ou une analyse incomplète.

4.2.1 Objectifs de l'analyse

L'un des objectifs fondamentaux de la définition du périmètre est de clarifier ce que l'on cherche à atteindre au cours de l'analyse des vulnérabilités. Les objectifs peuvent inclure :

- **Identification des vulnérabilités critiques** : Détecter les failles qui pourraient avoir un impact majeur sur la sécurité des données, la disponibilité des services ou la réputation de l'organisation;
- **Recherche de failles spécifiques** : cibler certaines vulnérabilités connues liées à des menaces précises (par exemple, des attaques par injection SQL, des failles de sécurité dans des logiciels ou des protocoles spécifiques);
- **Analyse proactive des risques** : Identifier les vulnérabilités avant qu'elles ne soient exploitées par des attaquants, dans une optique de prévention.

En précisant ces objectifs, on veille à ce que l'analyse soit bien orientée, avec une vision claire des menaces à traiter et des risques à atténuer.

4.2.2 Systèmes cibles

Une fois les objectifs définis, il est crucial de préciser les systèmes et composants qui feront l'objet de l'analyse. Cette étape permet de concentrer les efforts sur les éléments les plus importants et les plus vulnérables du système. Parmi les systèmes cibles, on retrouve typiquement :

- **Serveurs** : Les serveurs web, bases de données, de messagerie ou de fichiers, qui peuvent contenir des données sensibles ou être des cibles privilégiées d'attaques;
- **Applications** : Les applications logicielles utilisées par l'organisation, qu'elles soient internes (ex. applications de gestion) ou externes (applications accessibles par les utilisateurs ou clients);
- **Bases de données** : Les systèmes de gestion de bases de données (SGBD) qui stockent des informations critiques, telles que les informations personnelles, financières ou professionnelles.

Chaque composant du système doit être examiné en fonction de sa criticité, de sa visibilité sur le réseau et de la nature des données qu'il manipule. Une telle identification permet de

s'assurer que l'analyse est exhaustive, tout en se concentrant sur les parties les plus sensibles du système.

4.2.3 Limites de l'analyse

Il est également essentiel de définir clairement les limites de l'analyse. Cela permet de préciser les zones qui ne seront pas couvertes et d'éviter toute ambiguïté concernant le périmètre de l'étude. Les limites peuvent inclure :

- **Tests non autorisés** : Dans certains cas, certaines vulnérabilités ou systèmes ne peuvent pas être testés en raison de contraintes légales ou d'un manque d'autorisation explicite de la part des responsables du système;
- **Portée temporelle de l'analyse** : L'analyse peut être limitée à une période spécifique, excluant les changements futurs ou les évolutions du système qui pourraient affecter les vulnérabilités détectées;
- Aucune analyse des appareils mobiles personnels des employés ne sera effectuée.

En fixant des limites claires, on s'assure que l'analyse est cadrée et respecte les contraintes de l'entreprise, tant en termes de ressources que de temps.

4.3 Collecte d'informations et reconnaissance

La collecte d'informations et la reconnaissance visent à obtenir une vision complète du système cible, en identifiant les composants, les services et les configurations qui pourraient présenter des vulnérabilités. Cette étape est généralement la première dans un processus d'analyse des vulnérabilités ou de test de pénétration (pentesting). Elle repose sur l'utilisation de différentes techniques et outils pour obtenir des données essentielles, sans interagir de manière intrusive avec les systèmes.

4.3.1 Recueillir des informations sur le système

L'objectif principal de cette phase est de rassembler toutes les informations possibles sur les technologies utilisées, les configurations des systèmes et des réseaux, ainsi que les politiques de sécurité appliquées. Ces informations sont cruciales pour déterminer où se trouvent les potentielles failles.

Ils ont comme système d'exploitation **Debian GNU/Linux 11 (Bullseye)** qui est une version stable avec des mises à jour régulière et comme serveur Web **Nginx** avec la version **1.14.2** .
Pour les politiques de sécurité, nous avons trouvé cette directive `SendEnv LANG LC_*` dans le fichier `/etc/ssh/ssh.conf` qui permet d'envoyer les variables d'environnement locales (langue, encodage, etc.) au serveur distant.
Problème : Cela peut **fuiter des informations système** vers un attaquant qui intercepterait une connexion SSH.
La connexion SSH est possible via un **mot de passe** , cela peut être une faille de sécurité. Il est préférable de forcer l'authentification par **clé SSH** .
Changement du port SSH (optionnel) : Par défaut, SSH écoute sur le **port 22** , ce qui est une cible fréquente d'attaques par force brute. Nous pouvons si possible changer le port d'écoute de SSH.

4.3.2 Cartographie du réseau et des services

Cette étape consiste à analyser le réseau et à identifier les services exposés. Elle permet de repérer les points d'accès aux systèmes et de détecter les services qui pourraient être vulnérables à des attaques.
Nous allons utiliser notre outil de scanner de réseau **Nmap** pour effectuer des analyses de port sur le serveur et autres appareils du réseau. Cette analyse permettra d'identifier les ports ouverts, les services associés et les versions des logiciels en fonctionnement.

```
nmap -sS -sV easycompta.fewnu.app
```

Cette commande combine deux options puissantes de Nmap pour effectuer une analyse approfondie de l'hôte spécifié. Voici une explication détaillée de ce que chaque option fait et ce que la commande accomplira dans son ensemble :

- -sS : SYN Scan (Scan furtif)
  Cette option réalise un scan SYN, souvent appelé "scan furtif". Il envoie des paquets SYN (synchronisation) aux ports cibles et attend des réponses pour déterminer l'état des ports (ouvert, fermé, filtré). C'est une technique de scan rapide et discrète qui n'établit pas de connexion complète (TCP handshake) avec la cible, ce qui la rend moins détectable par les systèmes de surveillance;

- -sV : Version Detection (Détection de version)

  Cette option permet de déterminer les versions des services qui s'exécutent sur les ports ouverts. Nmap envoie des requêtes spécifiques aux services pour obtenir des informations sur les logiciels et leurs versions, ce qui peut aider à identifier des vulnérabilités spécifiques aux versions de logiciels.

Nmap commence par résoudre **easycompta.fewnu.app** en une adresse IP. Envoie des paquets SYN aux ports cibles de l'adresse IP résolue . Attend des réponses (SYN/ACK, RST, ou aucun retour) pour déterminer l'état des ports :

- SYN/ACK : Le port est ouvert;
- RST : Le port est fermé;
- Aucune réponse ou filtré : Le port peut être filtré par un pare-feu.

Pour les ports ouverts identifiés lors du scan SYN, Nmap envoie des requêtes supplémentaires spécifiques aux services pour collecter des informations sur les logiciels et leurs versions.

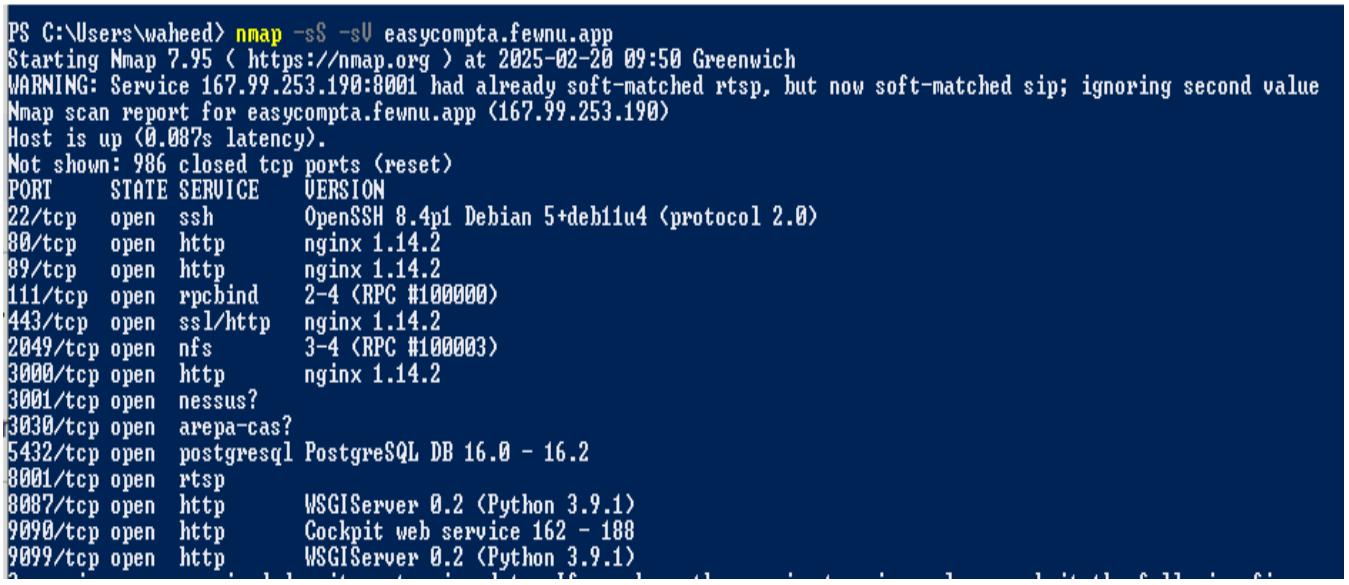


Figure 8 : Scan réseau avec Nmap



Figure 9 : Scan réseau avec Nmap (suite)

Une interprétation des résultats :

- L'hôte cible, **easycompta.fewnu.app** (résolue en adresse IP : 167.99.253.190), est actif;
- **Port 22 (SSH) :** La version utilisée est OpenSSH 8.4p1 Debian;
- **Port 80, 89, 443, 3000, 8087, 9090, 9099 (HTTP/HTTPS)** : Tous ces ports sont ouverts avec **Nginx 1.14.2** comme serveur web;
- **Port 5432 (PostgreSQL)** : Ce port est ouvert, ce qui signifie qu'une base de données PostgreSQL est accessible. Elle pourrait être un vecteur potentiel pour une injection SQL;
- **Port 111 (RPC)** et **Port 2049 (NFS)** : Ces services sont ouverts, ce qui peut aussi être un point d'entrée si mal configuré;
- **Port 3001 (Nessus)** : Il est possible que ce port soit utilisé par **Nessus**, un outil de gestion des vulnérabilités. Cela pourrait indiquer que des scans de sécurité sont effectués sur cette machine;
- **Port 3030 (Arepa-cas)** : Ce service n'est pas bien reconnu, mais il pourrait être spécifique à une application customisée. Il peut valoir la peine d'être exploré;
- **Port 8001 (RTSP)** : Ce port semble être utilisé pour le **Real-Time Streaming Protocol (RTSP)**, potentiellement utilisé pour du streaming vidéo.

4.4 Utilisation des outils d’analyse des vulnérabilités

L’analyse des vulnérabilités repose sur l'utilisation combinée d'outils automatisés et de tests manuels afin d’identifier et de valider les failles de sécurité présentes dans l’environnement cible. Cette section détaille les différentes approches utilisées pour détecter les vulnérabilités sur le système.

Pour cette analyse, l'outil principal utilisé est **OWASP ZAP** , un scanner de sécurité open-source spécialisé dans l'analyse des applications web.

4.4.1 Scan de vulnérabilités avec OWASP ZAP

Les scanners de vulnérabilités automatisés permettent d’identifier rapidement des failles connues en comparant la configuration et les versions des composants du système avec des bases de données de vulnérabilités.

OWASP ZAP (Zed Attack Proxy) est un outil puissant permettant d'analyser la sécurité des applications web en détectant les vulnérabilités courantes comme les injections SQL, les failles XSS, ou les erreurs de configuration.

L'écran d'exploration manuelle nous permet de lancer le navigateur de notre choix afin de pouvoir explorer toute l'application tout en utilisant le proxy ZAP. Sélectionnez le navigateur souhaité dans le menu déroulant et appuyez sur le bouton « Lancer le navigateur ».

Les navigateurs lancés de cette manière seront configurés pour utiliser le proxy via ZAP et ignoreront les erreurs de certificat. Vous n'aurez donc pas besoin d'importer le certificat d'autorité de certification racine ZAP.

ZAP se souviendra du dernier navigateur que vous avez choisi lorsque vous le redémarrerez.

Un bouton sera également ajouté à la barre d'outils qui lancera le dernier navigateur choisi. L'icône changera pour représenter le navigateur concerné

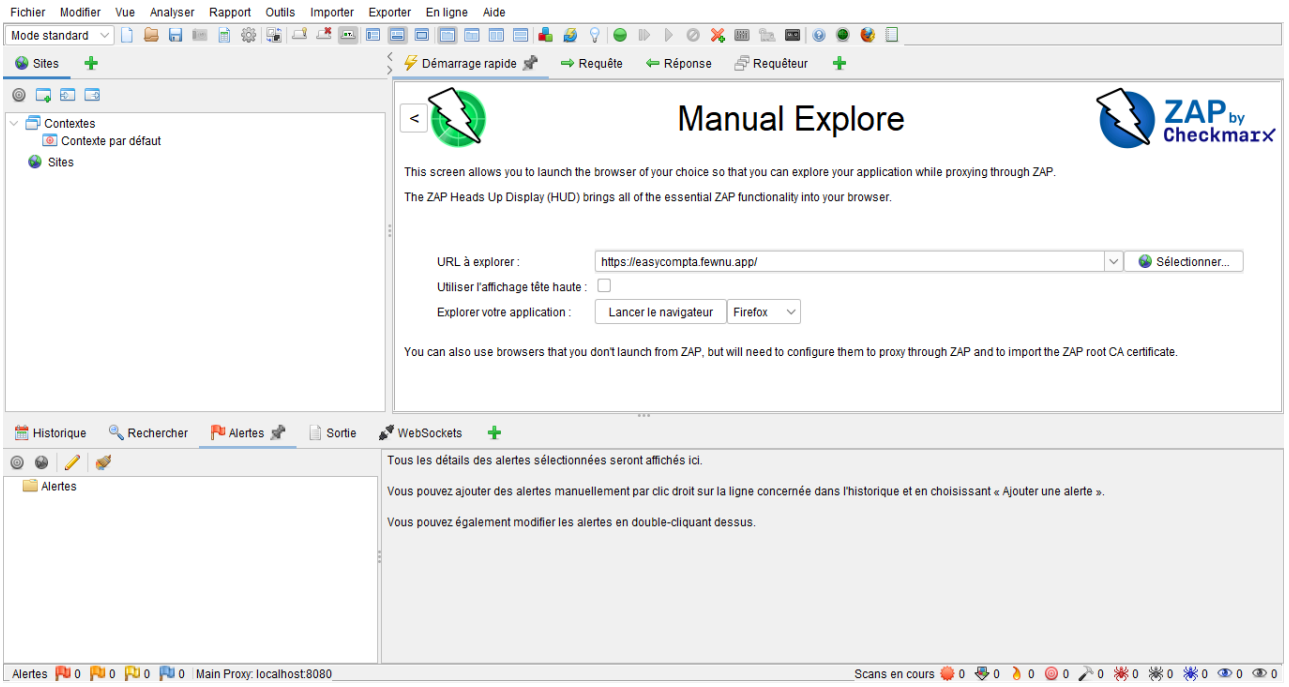


Figure 10 : Scan des vulnérabilités avec OWASP ZAP

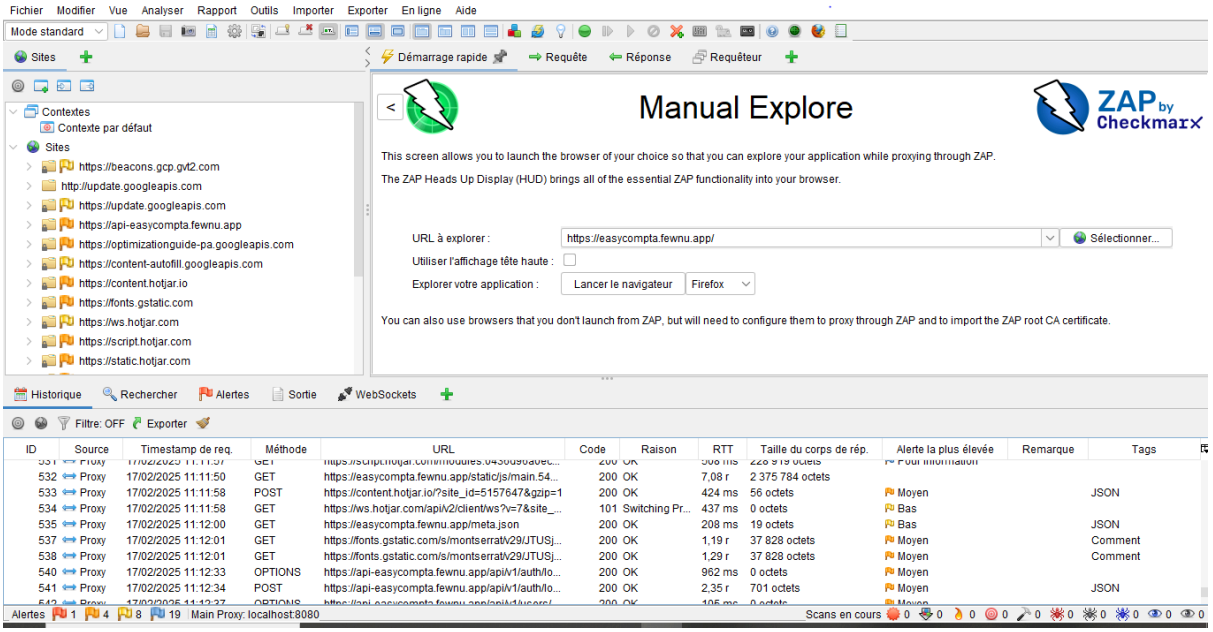


Figure 11 : Scan des vulnérabilités avec OWASP ZAP (la sortie de l'historique)

Cette capture montre les différentes failles que notre application a.

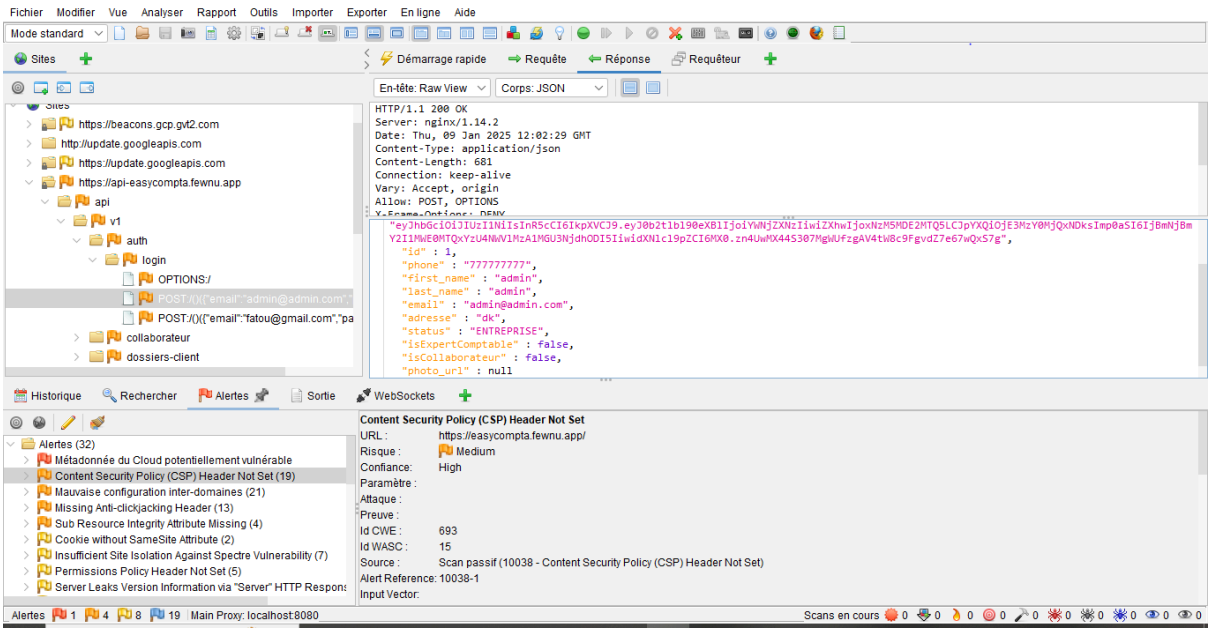


Figure 12 : Scan des vulnérabilités avec OWASP ZAP (les alertes)

Nous avons 32 alertes qui sont principalement liées à l'application, à l'API et à la configuration du serveur ainsi que des liens qu'ils ont été définis dans l'application.

Ce tableau indique le nombre d'alertes pour chaque niveau de risque et de confiance; les pourcentages entre parenthèses représentent le nombre en pourcentage du nombre total d'alertes, arrondi en décimale.

Tableau 5 : Nombre d'alertes pour niveau de risque et confiance

Risques	Confiance			
	Haut	Moyen	Faible	Total
Haut	0 (0,0 %)	0 (0,0 %)	1 (3,1 %)	1 (3,1 %)
Moyen	2 (6,2 %)	2 (6,2 %)	0 (0,0 %)	4 (12,5 %)
Faible	3 (9,4 %)	4 (12,5 %)	1 (3,1 %)	8 (25,0 %)
Pour plus d'informations	4 (12,5 %)	13 (40,6 %)	2 (6,2 %)	19 (59,4 %)
Total	9 (28,1 %)	19 (59,4 %)	4 (12,5 %)	32 (100%)

Ce tableau (Tableau 5) présente les résultats d'un scan de sécurité réalisé avec OWASP ZAP sur l'application Easy Compta . Il classe les vulnérabilités détectées selon deux critères :

- Le niveau de risque (Haut, Moyen, Faible, Pour plus d'informations).
- Le niveau de confiance attribué par l'outil (Haut, Moyen, Faible).

Nous pouvons analyser les données comme suit :

- Nombre total d'alertes détectées : 32
- Répartition des alertes selon le risque :
    - 1 alerte à risque élevé (3,1 %)
    - 4 alertes à risque moyen (12,5 %)
    - 8 alertes à risque faible (25 %)

- 19 alertes nécessitant plus d'informations (59,4 %)

La répartition des alertes selon le niveau de confiance est :

- 9 alertes avec un niveau de confiance élevé (28,1 %)
- 19 alertes avec un niveau de confiance moyen (59,4 %)
- 4 alertes avec un niveau de confiance faible (12,5 %)

4.4.2 Tests manuels pour la validation des vulnérabilités

En complément de l'analyse automatisée avec **OWASP ZAP** , des tests manuels ont été réalisés afin de vérifier la faisabilité et l'impact des vulnérabilités détectées. Ces tests ont révélé des faiblesses critiques, notamment une authentification faible permettant un accès non autorisé aux interfaces sensibles.

Accès aux routes protégées de l'application

Les tests ont permis d'accéder à l'ensemble des routes de l'application sans authentification robuste. Plusieurs points de terminaison sensibles ont été explorés, exposant des données critiques et des fonctionnalités réservées aux utilisateurs authentifiés.

Accès non autorisé à l'administration Django

Une vulnérabilité a été identifiée au niveau du **système d'authentification** , permettant l'accès à l'interface d'administration Django.

Cause principale :

- Le mot de passe administrateur, bien que stocké sous forme hachée, a été découvert;
- L'algorithme de hachage utilisé ne garantissait pas une résistance suffisante aux attaques;
- Un mot de passe faible a été utilisé, ce qui l'a rendu vulnérable à des attaques par **brute-force** et **rainbow table** .

Une fois le mot de passe découvert, l'accès a été obtenu via l'interface d'administration

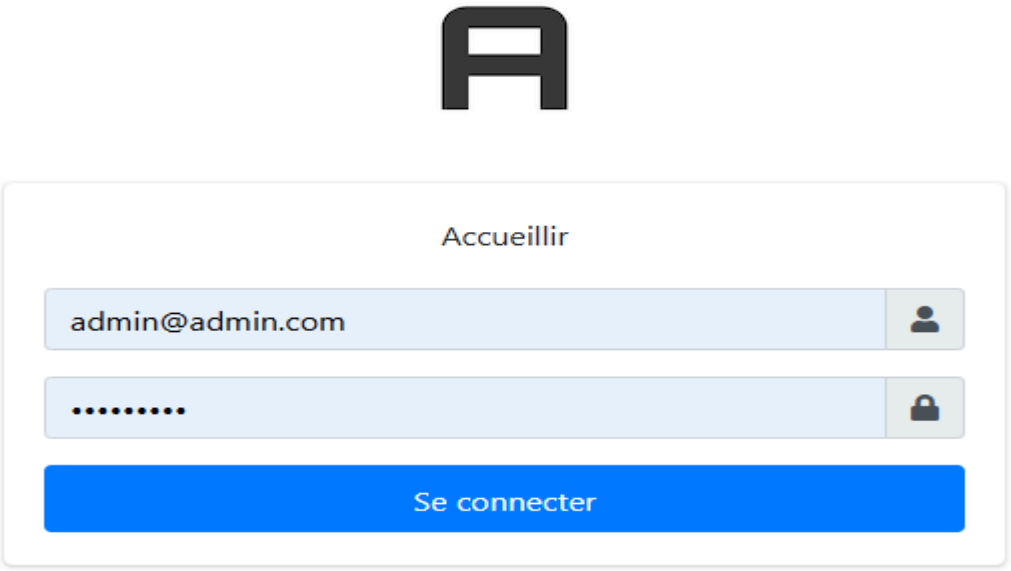


Figure 13 : Page d'authentification

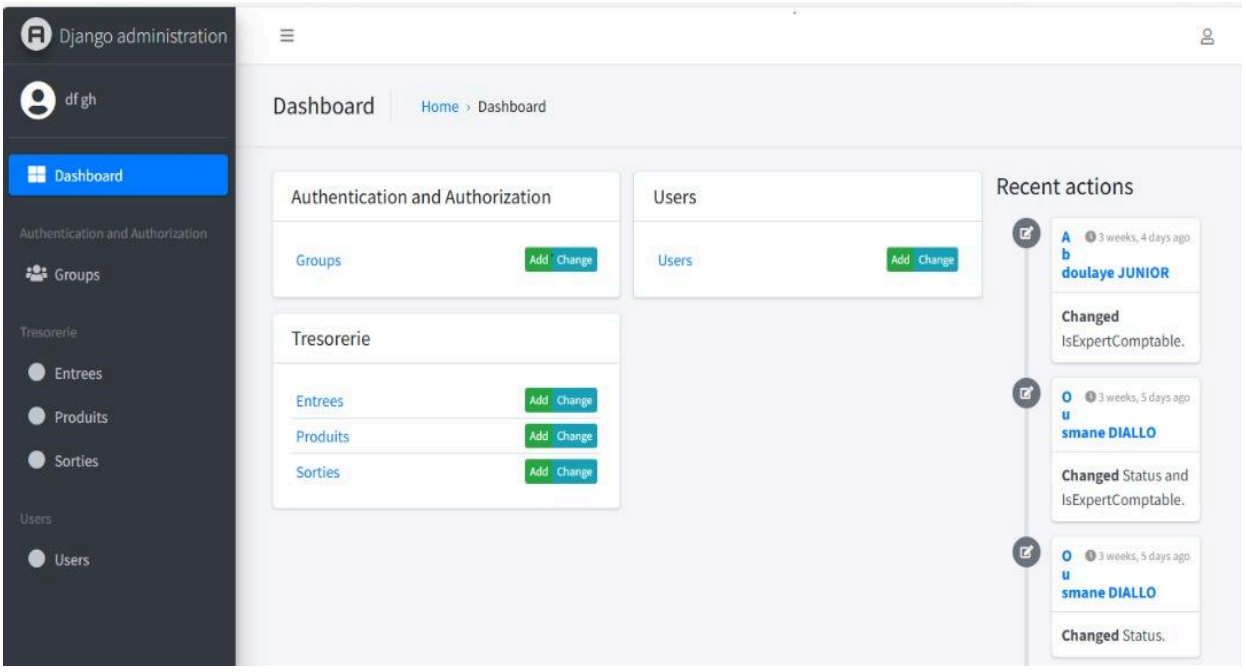


Figure 14 : Page d'administration django

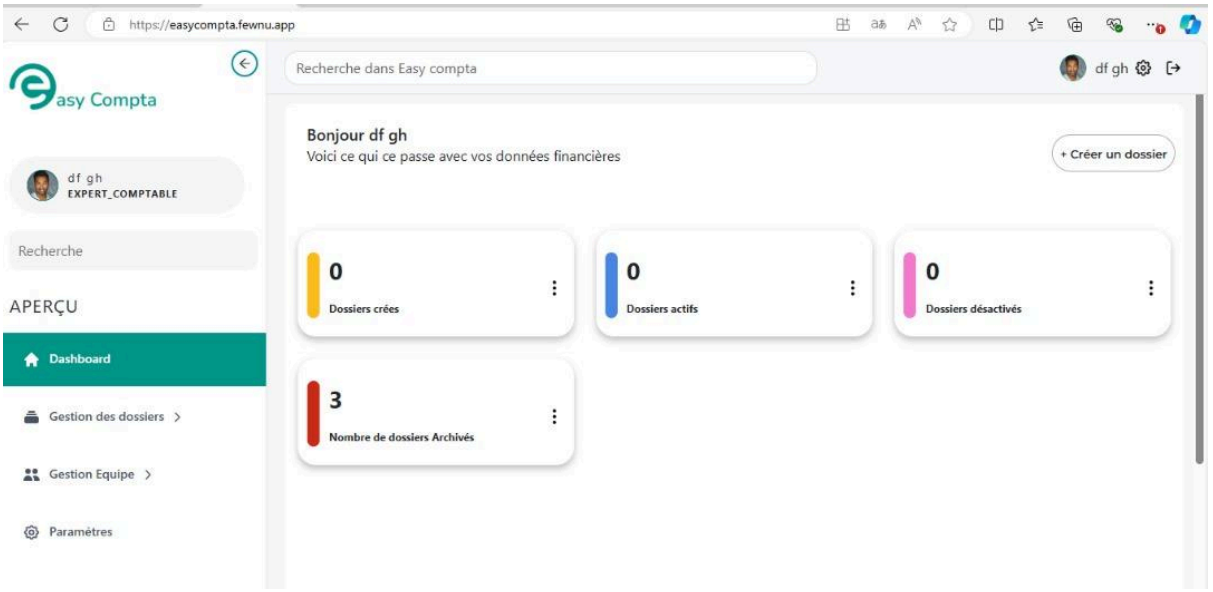


Figure 15 : Interface de l'admin django

L'authentification faible a permis un accès non autorisé aux routes protégées et à l'administration Django. La mise en place de **mesures de renforcement** est impérative pour sécuriser l'application, notamment à travers :

- Un **meilleur hachage des mots de passe** (bcrypt, Argon2);
- Une **politique de mots de passe robuste** et un **verrouillage des comptes après plusieurs échecs**;
- Une **restriction d'accès à l'administration Django** via des règles de pare-feu ou un VPN.

Ces mesures contribueront à améliorer significativement la sécurité de l'application et à réduire les risques d'intrusion.

4.5 Identification et classification des vulnérabilités

L'analyse des vulnérabilités permet d'identifier les failles de sécurité présentes dans le système et de les classer en fonction de leur nature, de leur impact et de leur exploitabilité. Cette phase repose sur les résultats de l'analyse avec **OWASP ZAP** , ainsi que sur les tests manuels effectués.

4.5.1 Identification des vulnérabilités

Les types de vulnérabilités identifiées incluent :

4.5.1.1 Métadonnée du Cloud potentiellement vulnérable

Le cloud est aujourd'hui un pilier essentiel de l'innovation. Son adoption massive témoigne des avantages qu'il offre en termes de flexibilité d'accès aux données et de capacité de stockage.La popularité du cloud s'explique également en partie par la complexité croissante des systèmes modernes, soutenus par des technologies comme l'IA ou l'IoT. Ces dernières nécessitent une infrastructure dynamique et évolutive ; besoins que les infrastructures cloud savent combler.

Les impacts de cette vulnérabilité sont :

- **Données d'Identifiants :** Les métadonnées peuvent contenir des identifiants d'accès, des clés API, ou des jetons d'authentification qui peuvent être exploités pour obtenir un accès non autorisé aux ressources Cloud;
- **Escalade de Privilèges :** Un attaquant ayant accès aux métadonnées peut obtenir des informations permettant l'escalade des privilèges, comme des identifiants avec des permissions élevées.

4.5.1.2 En-têtes HTTP de Sécurité non définies

Content Security Policy (CSP) Header Not Set

La politique de sécurité du contenu (CSP) est une couche de sécurité supplémentaire qui permet de détecter et d'atténuer certains types d'attaques, notamment les attaques Cross Site Scripting (XSS) et les attaques par injection de données. Ces attaques sont utilisées pour tout, du vol de données à la dégradation de site ou à la distribution de logiciels malveillants.

Les impacts de l'absence de CSP sont :

- **Attaques Cross Site Scripting (XSS)** : sans en-tête CSP, votre site Web est vulnérable aux attaques XSS [B1], où un attaquant peut injecter des scripts malveillants dans vos pages Web et voler des informations utilisateur sensibles ou effectuer des actions non autorisées au nom de l'utilisateur;

- **Attaques par injection de données :** les attaquants peuvent injecter du contenu malveillant dans votre site Web ou votre application, entraînant des violations de données, des accès non autorisés ou la distribution de logiciels malveillants à vos utilisateurs [B1].

Missing Anti-clickjacking Header

Le Clickjacking, aussi connu sous le nom de "UI redress attack", est une attaque où un utilisateur est trompé pour cliquer sur quelque chose différent de ce qu'il perçoit, permettant ainsi à un attaquant de voler des clics, des frappes de clavier ou d'autres interactions utilisateur.

Les impacts de l'absence de l'en-tête Anti-clickjacking sont :

- **Vulnérabilité aux attaques de Clickjacking** : Les utilisateurs peuvent être induits en erreur pour interagir avec des éléments invisibles ou déguisés sur une page, compromettant potentiellement la sécurité et la confidentialité;
- **Risque accru pour les opérations critiques** : Les actions sensibles, comme les transactions financières ou les changements de configuration, peuvent être réalisées sans le consentement de l'utilisateur.

Permissions Policy Header Not Set

L'en-tête de la politique d'autorisations est une couche de sécurité supplémentaire qui permet de restreindre l'accès ou l'utilisation non autorisés des fonctionnalités du navigateur/client par les ressources Web.

Les impacts de cette vulnérabilité :

Si l'en-tête de la politique d'autorisations n'est pas défini, les ressources Web peuvent avoir un accès illimité aux fonctionnalités sensibles du navigateur. Cela peut entraîner des violations potentielles de la confidentialité, une collecte de données non autorisée ou une utilisation abusive des ressources utilisateur. Les attaquants pourraient exploiter ces fonctionnalités pour capturer des informations sensibles, enregistrer de l'audio/vidéo sans consentement, suivre la localisation des utilisateurs ou forcer le navigateur à passer en mode plein écran sans interaction de l'utilisateur. Il est essentiel de définir l'en-tête de la politique d'autorisations pour atténuer ces risques et protéger la confidentialité des utilisateurs.

Strict-Transport-Security Header Not Set

HTTP Strict Transport Security (HSTS) est un mécanisme de politique de sécurité Web par lequel un serveur Web déclare que les agents utilisateurs conformes (tels qu'un navigateur Web) doivent interagir avec lui en utilisant uniquement des connexions HTTPS sécurisées (c'est-à-dire HTTP superposé sur TLS/SSL).

Les impacts de l'Absence de l'En-tête Strict-Transport-Security sont :

- **Vulnérabilité aux attaques Downgrade :** Sans HSTS, un attaquant peut exploiter des vulnérabilités dans le protocole HTTP pour forcer une connexion non sécurisée (HTTP) au lieu d'une connexion sécurisée (HTTPS). Cela peut permettre à l'attaquant de manipuler ou d'intercepter les données échangées;
- **Risque d'attaques de Type Man-in-the-Middle (MITM) :** Les connexions non sécurisées peuvent être interceptées par un attaquant, permettant l'exfiltration ou la manipulation de données sensibles, comme les informations d'identification des utilisateurs.

X-Content-Type-Options Header Missing

L'en-tête HTTP X-Content-Type-Options est utilisé pour empêcher les navigateurs de faire des déductions ou des "sniffing" du type de contenu. Il force les navigateurs à respecter le type de contenu spécifié dans les en-têtes Content-Type.

Les impacts de l'Absence de l'En-tête X-Content-Type-Options sont :

- **Attaques de Type MIME Sniffing :** Sans cette en-tête, les navigateurs peuvent effectuer des "sniffing" des types MIME, ce qui peut entraîner l'interprétation incorrecte des fichiers;
- **Compromission de la Sécurité des Données :** L'absence de cette protection peut exposer les utilisateurs à des risques accrus de compromission des données et de perte de confidentialité.

4.5.1.3 Problèmes de stockage et de divulgation d'informations

Divulgation d'informations - Information in Browser localStorage / sessionStorage

Les objets de stockage Web **localStorage** et **sessionStorage** permettent d’enregistrer les paires clé/valeur dans le navigateur.

Ce qui est intéressant à leur sujet, c’est que les données survivent à une actualisation de la page (pour sessionStorage) et même à un redémarrage complet du navigateur (pour localStorage).

Les impacts de cette vulnérabilité sont :

- **Stockage de Données Sensibles en Clair :** Contrairement aux cookies, les données stockées dans **localStorage** et **sessionStorage** ne sont pas automatiquement sécurisées via les options HttpOnly et Secure. Cela signifie qu'un attaquant peut facilement y accéder via du JavaScript malveillant, notamment en cas de **Cross-Site Scripting (XSS)**;
- **Persistance des Données :** Les données restent stockées même après la fermeture du navigateur, ce qui augmente le risque d'exposition, notamment si un utilisateur se connecte sur un appareil partagé.

Information Disclosure - Sensitive Information in URL

La divulgation d'informations, également appelée fuite d'informations, se produit lorsqu'un site Web révèle involontairement des informations sensibles à ses utilisateurs. Selon le contexte, les sites Web peuvent divulguer toutes sortes d'informations à un attaquant potentiel.

Les impacts de cette vulnérabilité :

Les vulnérabilités liées à la divulgation d'informations peuvent avoir un impact direct ou indirect, en fonction de l'objectif du site Web et, par conséquent, des informations qu'un attaquant est en mesure d'obtenir.

En revanche, la fuite d'informations techniques, comme la structure des répertoires ou les frameworks tiers utilisés, peut n'avoir que peu ou pas d'impact direct. Cependant, entre de mauvaises mains, ces informations peuvent être essentielles pour construire un certain nombre d'exploits. La gravité de la situation dépend dans ce cas de ce que l'attaquant est capable de faire avec ces informations.

4.5.1.4 Problèmes liés au cache et à la mise en cache

Réexaminer les directives de contrôle du cache / Contenu stockable et cacheable

La vulnérabilité « Réexaminer les directives de contrôle du cache » est un problème courant rencontré dans les applications Web qui utilisent la mise en cache pour améliorer les performances du site Web. Lorsqu'elle n'est pas configurée correctement, la mise en cache peut exposer des informations sensibles, compromettre la confidentialité des utilisateurs ou même compromettre l'intégralité du site Web.

Les impacts de cette vulnérabilité sont :

- **Fuite d'informations sensibles** : Si des pages contenant des données confidentielles (ex. sessions utilisateur, données personnelles, informations de paiement) sont stockées dans le cache, un attaquant ou un utilisateur non autorisé pourrait y accéder via un historique de navigation ou un proxy intermédiaire;
- **Manipulation de contenu** : Des attaques comme le cache poisoning peuvent permettre à un attaquant d'injecter du contenu malveillant en exploitant des directives de mise en cache faibles ou absentes.

4.5.1.5 Gestion des sessions et de l'authentification

Requête d'authentification identifiée

D'un point de vue conceptuel, les vulnérabilités liées à l'authentification sont faciles à comprendre. Cependant, elles sont généralement critiques en raison de la relation évidente entre l'authentification et la sécurité.

Les impacts de l'authentification vulnérable :

Les vulnérabilités d'authentification peuvent avoir de graves conséquences. Si un attaquant contourne l'authentification ou s'introduit par force dans le compte d'un autre utilisateur, il a accès à toutes les données et fonctionnalités du compte compromis. S'il parvient à compromettre un compte hautement privilégié, comme celui d'un administrateur système, il peut prendre le contrôle total de l'ensemble de l'application et potentiellement accéder à l'infrastructure interne.

Même en compromettant un compte à faibles privilèges, un attaquant peut toujours accéder à des données auxquelles il n'aurait pas dû avoir accès, comme des informations commerciales sensibles. Même si le compte n'a accès à aucune donnée sensible, il peut toujours permettre à l'attaquant d'accéder à des pages supplémentaires, ce qui offre une surface d'attaque supplémentaire. Souvent, les attaques de haute gravité ne sont pas possibles à partir de pages accessibles au public, mais elles peuvent l'être à partir d'une page interne.

Réponse de gestion de session identifiée

L'analyse a révélé que la gestion des sessions de l'application pourrait présenter des failles de sécurité.

Les impacts de cette vulnérabilité sont :

- **Usurpation de session (Session Hijacking)** : Un attaquant peut intercepter l'identifiant de session et l'utiliser pour se faire passer pour un utilisateur légitime;
- **Fixation de session (Session Fixation)** : Un attaquant force un utilisateur à utiliser un identifiant de session prédéfini, lui permettant d'accéder à son compte après l'authentification.

4.5.1.6 Serveur & Technologies

Server Leaks Version Information via "Server" HTTP Response Header Field

Lors de l'analyse, il a été constaté que le serveur web révèle sa version via l'en-tête HTTP **Server** dans les réponses aux requêtes. Cette information peut être utilisée par des attaquants pour identifier précisément le type et la version du serveur utilisé.

Les impacts de cette vulnérabilité sont :

- **Facilitation des attaques ciblées :** Un attaquant peut exploiter des vulnérabilités spécifiques connues pour une version donnée du serveur.
- **Augmentation de la surface d'attaque :** Plus un attaquant dispose d'informations sur l'environnement, plus il peut affiner ses stratégies d'attaque.

Tech Detected - Amazon CloudFront / Google Font API / React / PWA / HTTP/3

L'analyse a permis d'identifier plusieurs technologies utilisées sur l'application :

- **Amazon CloudFront** : Un service de distribution de contenu (CDN) qui permet d'améliorer les performances et la disponibilité du site.
- **Google Font API** : Utilisé pour charger les politiques de caractères externes depuis les serveurs de Google.
- **React** : Un framework JavaScript utilisé pour le développement d'interfaces utilisateur dynamiques.
- **Progressive Web App (PWA)** : Une approche permettant aux applications web d'offrir une expérience similaire à une application mobile.
- **HTTP/3** : Une version récente du protocole HTTP qui améliore la vitesse et la sécurité des communications entre le client et le serveur.

Les impacts sur la sécurité de l’application sont :

- **Risques liés aux dépendances :** Si React et ses bibliothèques ne sont pas à jour, elles peuvent contenir des vulnérabilités exploitables (ex. : failles XSS, attaques par désérialisation).
- **Exposition aux attaques sur les CDN :** Une mauvaise configuration d'Amazon CloudFront peut entraîner une fuite de données ou un accès non autorisé à certaines ressources.
- **Fuites potentielles via Google Font API :** L'utilisation d'API externes implique que certaines informations, comme les requêtes de politiques, sont envoyées à Google, ce qui peut soulever des préoccupations en matière de confidentialité.

4.5.2 Classification des vulnérabilités

L'analyse des vulnérabilités a mis en évidence plusieurs failles critiques affectant la sécurité du système et de l'application. La classification de ces vulnérabilités permet de mieux cibler les mesures de remédiation et de prioriser les actions correctives pour minimiser les risques.

Tableau 6 : Classification des vulnérabilités

Type de vulnérabilités	**Description**
Métadonnée du Cloud potentiellement vulnérable	L'attaque des métadonnées du cloud tente d'exploiter un serveur NGINX mal configuré afin d'accéder aux métadonnées d'instance gérées par des fournisseurs de services cloud tels qu'AWS, GCP et Azure.
En-têtes HTTP de Sécurité non définies	Les en-têtes HTTP de sécurité sont des directives envoyées par le serveur web au navigateur pour renforcer la protection contre certaines attaques courantes, comme le **Cross-Site Scripting (XSS)** , le **Clickjacking** , ou encore les **attaques Man-in-the-Middle (MitM)** .
Problèmes de stockage et de divulgation d'informations	Les problèmes de stockage et de divulgation d'informations concernent la mauvaise gestion des données sensibles dans une application. Ils constituent un risque majeur pour la sécurité des applications.

Type de vulnérabilités	**Description**
Problèmes liés au cache et à la mise en cache	La mise en cache est une technique utilisée pour optimiser les performances d'une application en stockant temporairement les données fréquemment utilisées. Cependant, une mauvaise gestion du cache peut entraîner des risques de fuite de données et d'accès non autorisé à des informations sensibles.
Gestion des sessions et de l'authentification	La gestion des sessions et de l'authentification est un élément clé de la sécurité des applications. Une mauvaise configuration peut entraîner des critiques de vulnérabilités comme le **vol de session** , **l'usurpation d'identité** , ou encore des attaques par **force brute** et **replay** .
Serveur & Technologies	Le serveur révèle sa version exacte via l'en-tête HTTP **Server** dans la réponse. Cela donne aux attaquants une indication sur la version utilisée, ce qui facilite l'exploitation des vulnérabilités connues. Les informations révélées sur le serveur et les technologies utilisées peuvent aider un attaquant à cibler des vulnérabilités spécifiques.

4.6 Évaluation des risques

L'évaluation des risques permet de mesurer l'impact potentiel des vulnérabilités identifiées et de déterminer les priorités d'atténuation. Cette évaluation repose sur plusieurs critères, notamment la gravité, la probabilité d'exploitation et l'impact sur les systèmes.
Voici un tableau de matrice d’évaluation des risques que ces vulnérabilités ont sur la sécurité du système et de l’application :

Tableau 7 : Matrice d’évaluation des risques

Vulnérabilités	**Probabilité d'exploitation**	**Impact**	**Niveau de risque**
Métadonnée du Cloud potentiellement vulnérable	Élevée	Élevé	Critique
En-têtes HTTP non définis	Moyenne	Moyen	Moyen
Stockage d'informations sensibles	Élevée	Élevé	Critique
Problèmes de cache	Moyenne	Moyen	Moyen
Mauvaise gestion des sessions	Élevée	Élevé	Critique
Vulnérabilités serveur	Élevée	Élevé	Elevé

La **matrice d'évaluation des risques** (Tableau 8) est un outil essentiel en cybersécurité permettant de **classer et prioriser les vulnérabilités** en fonction de leur impact et de leur probabilité d'exploitation. Elle facilite la prise de décision en identifiant les vulnérabilités les plus critiques et en définissant l'ordre des actions correctives à entreprendre.

Elle croise **deux facteurs principaux** pour évaluer chaque vulnérabilité :

1. **Probabilité d'Exploitation**
    - **Faible** : Exploitation difficile, requiert un attaquant expérimenté et des conditions spécifiques.
    - **Moyenne** : Exploitable avec des outils spécialisés, nécessite un certain niveau de compétence.
    - **Élevée** : Vulnérabilité facilement exploitable avec des outils courants et peu de connaissances techniques.
2. **Impact sur le Système**
    - **Faible** : Dommages mineurs, peu d'effets sur les opérations.
    - **Moyen** : Perturbation de certaines fonctionnalités, fuites d'informations non critiques.

- **Élevé** : Compromission totale du système, fuite de données sensibles, arrêt des services.

Les vulnérabilités classées "Critique" doivent être corrigées immédiatement , car elles présentent un risque majeur pour la sécurité du système.
Les vulnérabilités "Élevé" doivent être corrigées rapidement , mais peuvent être traitées après les risques critiques.
Les vulnérabilités "Moyen" doivent être corrigées dès que possible , mais elles ont un impact moindre sur la sécurité globale.

La matrice d'évaluation des risques aide à rationaliser l'effort de correction en concentrant les ressources sur les menaces les plus dangereuses en premier.

4.7 Conclusion

L'analyse des vulnérabilités existantes réalisées dans ce chapitre a permis d'identifier et d'évaluer les principales failles de sécurité affectant le système étudié. Grâce à une approche méthodique et à l'utilisation d'outils spécialisés, nous avons pu classer ces vulnérabilités en fonction de leur gravité, de leur potentiel d'impact et de leur exploitabilité. Cette analyse a mis en évidence des risques critiques qui pourraient nuire à la confidentialité, à l'intégrité et à la disponibilité des données et services.

Les résultats obtenus soulignent l'importance d'une gestion proactive des vulnérabilités, incluant des mises à jour régulières, des tests de sécurité continue et l'application de bonnes pratiques en matière de cybersécurité. La mise en place de mesures correctives et préventives adaptées est essentielle pour atténuer ces risques et renforcer la résilience du système face aux cybermenaces.

En conclusion, cette analyse démontre que la sécurité d'un système ne peut être garantie de manière absolue, mais qu'une surveillance constante et une continuité des mécanismes de protection permettent de réduire considérablement les risques. Il est donc recommandé d'intégrer ces pratiques dans une politique de sécurité globale, afin de garantir une meilleure protection des actifs numériques et une plus grande confiance des utilisateurs et partenaires.

CHAPITRE 5 : PROPOSITION DE MÉCANISMES DE PROTECTION

5.1 Introduction

Dans un contexte où les cybermenaces sont de plus en plus sophistiquées, la mise en place de mécanismes de protection efficaces est essentielle pour garantir la sécurité des systèmes d'information. Après avoir identifié et analysé les vulnérabilités existantes dans le chapitre précédent, il est crucial de proposer des solutions adaptées afin de réduire les risques et d'améliorer la résilience du système face aux attaques.

Ce chapitre présente une approche structurée visant à renforcer la sécurité des infrastructures, des applications et des données. Il repose sur des principes fondamentaux tels que la défense en profondeur, la gestion des accès, la surveillance des menaces et la sensibilisation des utilisateurs. Les différentes solutions proposées s'appuient sur les meilleures pratiques en matière de cybersécurité, en tenant compte des normes et recommandations internationales.

5.2 Approche générale de la protection

L'approche générale de la protection repose sur des stratégies de sécurité éprouvées permettant de réduire les risques liés aux vulnérabilités identifiées. Elle vise à établir un cadre robuste pour la défense des systèmes, en combinant plusieurs niveaux de protection. Cette approche repose sur les principes suivants :

5.2.1 Stratégie de défense en profondeur

La **défense en profondeur** est une approche qui consiste à superposer plusieurs couches de sécurité afin de ralentir ou d'empêcher une attaque. Elle repose sur l'idée qu'aucune mesure de protection unique n'est suffisante pour garantir une sécurité totale. Les principaux niveaux de défense comprennent :

- **Sécurité périmétrique** : Pare-feux, filtrage du trafic, segmentation réseau.
- **Sécurité des hôtes** : Correctifs de sécurité, antivirus, chiffrement des données.

- **Sécurité des applications** : Protection contre les injections SQL, XSS, authentification renforcée.
- **Sécurité des données** : Chiffrement des bases de données, sauvegardes régulières.
- **Surveillance et détection** : SIEM, journaux d'audit, systèmes IDS/IPS.

L'objectif est d'empêcher une attaque d'atteindre des couches critiques du système, même si l'une des protections est contournée.

5.2.2 Modèle de gestion des risques

Avant de mettre en place des mesures de protection, il est essentiel d'évaluer les risques en fonction des critères suivants :

- **Probabilité d'exploitation d'une vulnérabilité** : Quelle est la probabilité qu'un attaquant exploite cette faille ?
- **Impact potentiel** : Quelles seraient les conséquences d'une exploitation réussie (perte de données, interruption de service, compromission d'informations sensibles) ?
- **et pratique des contre-mesures** : Quel est le coût des solutions de protection par rapport aux bénéfices apportés ?

Cette gestion des risques permet de **prioriser les mesures de protection** et d'optimiser l'allocation des ressources en fonction du niveau de criticité des vulnérabilités.

5.3 Mécanismes de protection proposés

l'objectif est de présenter des solutions ou stratégies que nous avons proposées pour sécuriser le système, l'application ou l'environnement en réponse aux vulnérabilités identifiées. Nous allons décrire les mécanismes de sécurité qui peuvent être mis en place pour contrer les menaces et atténuer les risques de sécurité.

5.3.1 Sécurisation des métadonnées du Cloud

Les solutions pour cette vulnérabilité sont :

- **Activer IMDSv2** :

```
aws ec2 modify-instance-metadata-options --instance-id i-xxxxxxxxx --http-tokens required
```

- **Restreindre l'accès aux métadonnées** via les règles de pare-feu :
    - Désactiver l'accès non nécessaire (169.254.169.254).
- **Ne jamais stocker des credentials dans les métadonnées**.

5.3.2 Définition des en-têtes HTTP de sécurité

Par défaut, Django et Nginx ne traitent pas certaines en-têtes HTTP de sécurité. Il faut donc les ajouter pour **protéger contre XSS, clickjacking et MITM** [B2].

Les solutions [B3] [B4] pour définir les en-têtes sont :

- Installer le paquet django-csp :

```
pip install django-csp
```

- Ajouter **csp.middleware.CSPMiddleware** dans le MIDDLEWARE dans settings.py :

```
MIDDLEWARE = [
   …
   'csp.middleware.CSPMiddleware',  # Ajout de CSP
   …
]
```

- Configurer Django pour envoyer des en-têtes sécurisés dans settings.py :

```
# Content Security Policy (CSP)
CSP_DEFAULT_SRC = ("'self'",)
CSP_IMG_SRC = ("'self'", "data:")
CSP_STYLE_SRC = ("'self'", "https://fonts.googleapis.com")
CSP_SCRIPT_SRC = ("'self'", "https://apis.google.com")
CSP_FONT_SRC = ("'self'", "https://fonts.gstatic.com")
CSP_CONNECT_SRC = ("'self'",)
CSP_FRAME_SRC = ("'none'",)

# Clickjacking Protection
X_FRAME_OPTIONS = 'DENY'
```

```
# HSTS
SECURE_HSTS_SECONDS = 31536000
SECURE_HSTS_INCLUDE_SUBDOMAINS = True
SECURE_HSTS_PRELOAD = True

# X-Content-Type-Options
SECURE_CONTENT_TYPE_NOSNIFF = True

# Feature-Policy (Permissions-Policy)
SECURE_FEATURE_POLICY = {
    'geolocation': "'none'",
    'camera': "'none'",
    'microphone': "'none'",
}
```

- Configurer Nginx pour forcer HTTPS et ajouter les headers dans nginx.conf :

```
server {
    listen 443 ssl;
    server_name yourdomain.com;

    # En-têtes de sécurité

    add_header Content-Security-Policy "
        default-src 'self';
        script-src 'self' https://apis.google.com;
        style-src 'self' https://fonts.googleapis.com;
        font-src 'self' https://fonts.gstatic.com;
        img-src 'self' data:;
        connect-src 'self';
        frame-ancestors 'none';
    " always;
```

```
    add_header X-Frame-Options "DENY";
    add_header X-Content-Type-Options "nosniff";
    add_header X-XSS-Protection "1; mode=block";
    add_header Strict-Transport-Security "max-age=31536000; includeSubDomains;
preload";
    add_header Referrer-Policy "no-referrer" always;
    add_header Permissions-Policy "geolocation=(), microphone=(), camera=()" always;

}
```

5.3.3 Problèmes de stockage et de divulgation d'informations

La sécurisation des fichiers sensibles dans Django [B2] **:**

- Utiliser un fichier **.env** sécurisé avec python-decouple :

```
pip install python-decouple
```

- Dans **.env**, ajouter ces directives :

```
SECRET_KEY=super_secret_key DATABASE_URL=postgres://user:password@host/db
```

- Dans settings.py, ajouter ces directives :

```
from decouple import config
SECRET_KEY = config("SECRET_KEY")
DATABASE_URL = config("DATABASE_URL")
```

- Désactiver le mode DEBUG en production :

```
DEBUG = False
```

- Configurer Django pour ne pas exposer les erreurs aux utilisateurs :

```
ALLOWED_HOSTS = ["yourdomain.com"]
```

- Configurer Nginx pour bloquer l'accès aux fichiers sensibles :

```
location ~* \.(env|log|sql|sqlite|ini|sh|pem|bak|old|swp)$ {
  deny all;
}
```

La sécurisation des logs d'erreur :

- Ne jamais exposer des erreurs aux utilisateur

```
LOGGING = {
  "version": 1,
  "disable_existing_loggers": False,
  "handlers": {
    "file": {
      "level": "ERROR",
      "class": "logging.FileHandler",
      "filename": "/var/log/django_errors.log",
    },
  },
  "loggers": {
    "django": {
      "handlers": ["file"],
      "level": "ERROR",
      "propagate": True,
    },
  },
}
```

5.3.4 Problèmes liés au cache et à la mise en cache

Les solutions spécifiques à cette vulnérabilité sont [B3] **:**

- Empêcher la mise en cache des réponses sensibles dans Django dans views.py :

```
from django.views.decorators.cache import never_cache
```

```
@never_cache
def my_sensitive_view(request):
    return render(request, "my_template.html")
```

- Configurer les en-têtes Cache-Control pour React.js dans Nginx :
  Ajoutez ces directives dans nginx.conf :

```
location / {
    add_header Cache-Control "no-store, no-cache, must-revalidate, max-age=0";
    add_header Pragma "no-cache";
    expires off;
}
```

5.3.5 Gestion des sessions et de l'authentification

Les solutions spécifiques à cette vulnérabilité sont :

- Sécuriser les cookies de session Django dans settings.py :

```
SESSION_COOKIE_SECURE = True  # Activer uniquement sur HTTPS
SESSION_COOKIE_HTTPONLY = True  # Empêcher l'accès JavaScript
SESSION_COOKIE_SAMESITE = "Strict"  # Protection CSRF
CSRF_COOKIE_SECURE = True
CSRF_COOKIE_HTTPONLY = True
```

- Améliorer la sécurité des JWT avec Django Rest Framework
  Configurer la durée de vie des tokens :

```
SIMPLE_JWT = {
    "ACCESS_TOKEN_LIFETIME": timedelta(minutes=15),
    "REFRESH_TOKEN_LIFETIME": timedelta(days=1),
    "ROTATE_REFRESH_TOKENS": True,
    "BLACKLIST_AFTER_ROTATION": True,
```

```
}
```

- Stocker les JWT dans httpOnly cookies côté React.js :

```
document.cookie = `access_token=${token}; HttpOnly; Secure; SameSite=Strict`;
```

5.3.6 Sécurisation du serveur et des technologies utilisées

La sécurisation du serveur Django :

- Utiliser Gunicorn pour exécuter Django au lieu du serveur de développement

```
pip install gunicorn
gunicorn --bind 0.0.0.0:8000 myproject.wsgi
```

- Limiter l'accès aux ports avec UFW

```
sudo ufw allow 443
sudo ufw allow 80
sudo ufw deny 8000  # Bloquer le port Django
sudo ufw enable
```

La sécurisation de Nginx (React.js + Django API)

- Ajouter ces directives dans nginx.conf :

```
server {
    listen 443 ssl;
    server_name yourdomain.com;

    ssl_certificate /etc/ssl/certs/your_cert.pem;
    ssl_certificate_key /etc/ssl/private/your_key.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
    ssl_prefer_server_ciphers on;
}
```

5.3.7 Surveillance et détection des incidents

La mise en place de mécanismes de surveillance est un élément essentiel pour garantir la sécurité, la disponibilité et la performance des applications. Une surveillance proactive permet non seulement de détecter rapidement les incidents, mais aussi d'anticiper les éventuelles menaces en identifiant des comportements anormaux ou des tentatives d'attaques.

Dans le cadre de ce mémoire, **UptimeRobot** a été utilisé pour surveiller l'état de l'application **easycompta.fewnu.app** et assurer un suivi en temps réel de sa disponibilité.

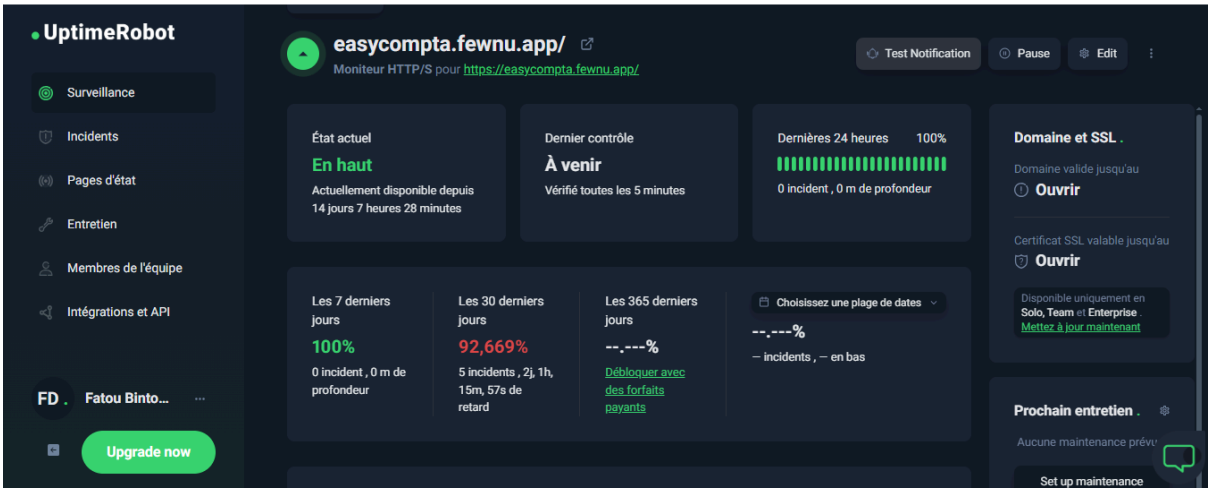


Figure 16 : Surveillance avec UptimeRobot

UptimeRobot est un service de surveillance en ligne qui permet de vérifier l'état d'un site web ou d'une application toutes les **5 minutes**. Il envoie des alertes en cas de panne et fournit un historique des performances de disponibilité.

Dans notre cas (Figure 14) les données actuelles sont :

- **État actuel** : L'application est **"En haut"**, disponible depuis **14 jours, 7 heures et 28 minutes**.
- **Dernier contrôle** : À venir, vérification toutes les **5 minutes**.
- **Disponibilité sur les 7 derniers jours** : **100%**, aucun incident détecté.
- **Disponibilité sur les 30 derniers jours** : **92,669%**, avec **5 incidents** ayant entraîné **2 jours, 1 heure, 15 minutes et 57 secondes de retard**.

- **Disponibilité sur les 365 derniers jours** : Non disponible dans la version actuelle du compte.

5.4 Conclusion

Dans ce chapitre, nous avons exploré une série de mécanismes de protection essentiels pour sécuriser les systèmes, applications et infrastructures face aux vulnérabilités identifiées dans les chapitres précédents. Ces mécanismes couvrent des aspects cruciaux de la sécurité, allant de la gestion des accès et de l'authentification à la sécurisation des communications, en passant par la prévention des attaques par injection, la gestion des sessions et la sécurisation des infrastructures.

La mise en œuvre de ces mesures offre une protection robuste contre une large gamme de menaces potentielles, allant des attaques internes à celles provenant de l'extérieur. Elles permettent de garantir la confidentialité, l'intégrité et la disponibilité des données, tout en améliorant la résilience du système face aux attaques sophistiquées.

Il est cependant important de noter que la sécurité est un processus continu et qu'aucun mécanisme n'est infaillible. La vigilance constante, la mise à jour régulière des systèmes et l'adaptation aux nouvelles menaces émergentes sont essentielles pour maintenir un environnement sécurisé. Les mécanismes de protection proposés doivent donc être intégrés dans une démarche proactive de gestion de la sécurité, comprenant des tests réguliers, des audits de sécurité et une veille continue sur les vulnérabilités.

CHAPITRE 6 : RENFORCEMENT DES CAPACITÉS EN SÉCURITÉ APPLICATIVE

6.1 Introduction

Dans un contexte où les cyberattaques se multiplient et deviennent de plus en plus sophistiquées, la sécurité applicative constitue un enjeu stratégique pour toute organisation. Les failles de sécurité dans les applications peuvent compromettre l'intégrité des données, exposer des informations sensibles et engendrer des pertes financières considérables.

Afin de réduire ces risques, il est essentiel de renforcer les capacités en sécurité applicative à travers des actions ciblées. Cela passe par la **sensibilisation des parties prenantes**, notamment les développeurs et les administrateurs système, afin qu'ils adoptent des pratiques sécurisées dès les premières phases du développement et de l'exploitation des applications. De plus, la **documentation des procédures et bonnes pratiques** joue un rôle clé dans la mise en place d'une gestion proactive de la sécurité, garantissant ainsi une protection continue contre les menaces potentielles.

Ce chapitre propose des stratégies concrètes pour renforcer la sécurité applicative, en mettant l'accent sur la formation, la sensibilisation et la mise en place de processus documentés pour une meilleure résilience face aux cyberattaques.

6.2 Sensibilisation des Parties Prenantes

La sécurité des identités numériques ne repose pas uniquement sur les technologies et les mécanismes de protection. L'un des maillons les plus critiques reste **le facteur humain**. Une sensibilisation efficace et une formation continue des parties prenantes (développeurs, administrateurs système, utilisateurs finaux) sont essentielles pour garantir la robustesse des mesures de protection mises en place.

6.2.1 Sensibilisation des Développeurs

Les développeurs jouent un rôle fondamental dans la sécurisation des applications et des systèmes. Une mauvaise pratique de développement peut introduire des vulnérabilités exploitables par des attaquants. Il est donc essentiel de les former aux bonnes pratiques de **développement sécurisé** afin de minimiser les risques et de garantir la robustesse des systèmes.

6.2.1.1 Connaissance des Vulnérabilités et des Menaces

La première étape de la sensibilisation consiste à comprendre les principales menaces qui pèsent sur les applications et les systèmes. Parmi celles-ci, on retrouve :

- **Les failles courantes du Top Ten OWASP** :
    - Injection SQL (SQLi);
    - Cross-Site Scripting (XSS);
    - Cross-Site Request Forgery (CSRF);
    - Exposition de données sensibles;
    - Mauvaise gestion de l'authentification et des sessions;
    - Contrôle d'accès inadéquat;
    - Injections de commandes et de code.
- **Les attaques ciblant les identités** :
    - Attaques par force brute et **credential stuffing;**
    - Vol de session et détournement d'authentification;
    - Hameçonnage (**phishing**) visant les développeurs (ex. compromission des comptes GitHub).

6.2.1.2 Adoption des Bonnes Pratiques de Développement Sécurisé

Pour réduire la surface d'attaque et éviter les erreurs courantes, les développeurs doivent intégrer **des principes de sécurité dès la conception** (Security by Design) et appliquer des bonnes pratiques tout au long du cycle de développement.

Les principes de développement sécurisé sont :

- **Appliquer le principe du moindre privilège (PoLP)** : accorder uniquement les permissions nécessaires;

- **Sanitiser et valider les entrées utilisateurs** afin d'éviter les injections de code;
- **Utiliser des mécanismes d'authentification et d'autorisation robustes** (ex. OAuth 2.0, OpenID Connect);
- **Protéger les données sensibles** par chiffrement et stockage sécurisé;
- **Éviter le stockage d'informations sensibles en clair** (mots de passe, tokens, clés API);
- **Restreindre l'exposition des endpoints et interfaces** en limitant les permissions et en utilisant des pare-feu applicatifs (WAF).

Les revues de code et tests de sécurité sont :

- **Intégrer des revues de code avec une approche sécurité** : chaque nouvelle fonctionnalité doit être analysée pour détecter d'éventuelles vulnérabilités;
- **Automatiser l'analyse statique et dynamique du code** avec des outils comme **SonarQube (SAST), OWASP ZAP (DAST), GitHub Dependabot, Snyk**;
- **Mettre en place des tests de pénétration réguliers** pour identifier et corriger les failles avant la mise en production.

6.2.1.3 Sécurisation des Données et des Communications

Un développement sécurisé passe également par **la protection des données** et **la sécurisation des échanges** entre systèmes et utilisateurs. Nous avons :

- **Chiffrement des données sensibles** avec AES-256 pour le stockage et TLS 1.3 pour les communications;
- **Utilisation des cookies sécurisés** (HttpOnly, Secure, SameSite) pour éviter le vol de sessions;
- **Implémentation d'un contrôle strict des permissions** en utilisant RBAC (Role-Based Access Control) ou ABAC (Attribute-Based Access Control);
- **Gestion des erreurs et des journaux** sans divulguer d'informations sensibles aux attaquants potentiels (ex. suppression des messages d'erreur détaillés en production).

6.2.2 Formation des Administrateurs Systèmes et Réseaux

Les administrateurs systèmes et réseaux jouent un rôle clé dans la mise en place et le maintien de la sécurité des infrastructures informatiques. Une mauvaise configuration des

systèmes, un manque de mises à jour ou une gestion inadéquate des accès peuvent exposer l'organisation à des cyberattaques. Il est donc essentiel de leur fournir une **formation continue** et de les sensibiliser aux **meilleures pratiques en matière de cybersécurité**.

6.2.2.1 Compréhension des Menaces et des Risques

Les administrateurs doivent être conscients des principales menaces qui pèsent sur les systèmes et les réseaux. Parmi celles-ci, on retrouve :

- **Les attaques réseau** :
    - **DDoS (Déni de Service Distribué)** : indisponibilité des ressources;
    - **Man-in-the-Middle (MITM)** : interception des communications;
    - **ARP Spoofing** et **DNS Poisoning** : manipulation des tables réseau.
- **Les attaques ciblant les identités** :
    - Phishing et attaques par hameçonnage;
    - Credential stuffing et brute-force;
    - Attaques sur les protocoles d'authentification (Kerberoasting, Pass-the-Hash).
- **Les malwares et ransomwares** :
    - Propagation via des failles non corrigées;
    - Exploitation des accès non restreints;
    - Chiffrement des données critiques avec demande de rançon.

6.2.2.2 Sécurisation des Systèmes et des Accès

Un administrateur doit appliquer **les bonnes pratiques de sécurisation des systèmes** pour réduire les risques liés aux cyberattaques.

La gestion des comptes et des privilèges doit être :

- **Appliquer le principe du moindre privilège (PoLP)** :
    - Limiter les comptes administrateurs aux seules tâches nécessitant des privilèges élevés;
    - Éviter l'utilisation du compte root ou administrateur pour des tâches courantes.
- **Utiliser l'authentification multifactorielle (MFA)** pour les accès critiques;

- **Segmenter les comptes** : comptes utilisateur, comptes de service, comptes administrateurs.

La sécurisation des serveurs et des postes de travail est :

- **Appliquer des politiques de mise à jour automatique** (OS, firmwares, logiciels);
- **Désactiver les services inutilisés** pour réduire la surface d'attaque;
- **Implémenter des politiques de durcissement des systèmes** (ex. CIS Benchmarks, STIG);
- **Configurer les journaux et surveiller des événements** pour détecter les activités suspectes.

La gestion des accès à distance est :

- **Restreindre l'accès SSH et RDP** aux seules adresses IP autorisées;
- **Utiliser des VPN sécurisés** avec chiffrement fort (IPsec, WireGuard, OpenVPN);
- **Désactiver l'authentification par mot de passe** pour SSH et privilégier les clés SSH.

6.2.2.3 Sécurisation du Réseau et des Communications

Un administrateur doit mettre en place des stratégies de protection réseau pour empêcher les intrusions et surveiller le trafic.

La segmentation et isolation des réseaux est de :

- **Mettre en place des VLANs** pour isoler les différents services et départements;
- **Utiliser des pare-feux (Firewalls)** pour filtrer le trafic entrant et sortant;
- **Implémenter des zones de sécurité (DMZ, réseau interne, réseau invité).**

La protection contre les attaques réseau est de :

- **Déployer des systèmes de détection et prévention des intrusions (IDS/IPS)** comme **Snort, Suricata**;
- **Limiter les ports et services exposés sur Internet** avec des ACLs (Access Control Lists);
- **Appliquer des règles strictes de filtrage des paquets** sur les routeurs et pare-feux.

Pour la sécurisation des protocoles de communication, il faut :

- **Désactiver les protocoles obsolètes et vulnérables** (ex. Telnet, SSLv3, SMBv1);
- **Forcer l'utilisation de TLS 1.3 et SSH pour les connexions sécurisées;**
- **Appliquer des certificats SSL/TLS valides** pour sécuriser les échanges.

6.2.2.4 Surveillance, Détection et Réponse aux Incidents

Une mise en place d'un système de monitoring pour :

- **Surveiller les logs système et applicatifs** avec des solutions comme **ELK (Elasticsearch, Logstash, Kibana), Splunk, Graylog**;
- **Implémenter une solution SIEM (Security Information and Event Management)** pour corréler les événements de sécurité;
- **Activer les logs d'audit** sur les systèmes critiques (ex. connexions, modifications de fichiers sensibles).

Pour la gestion des alertes et réponse aux incidents, il faut :

- **Définir des procédures de gestion des incidents de sécurité**;
- **Effectuer des simulations d'attaques** pour tester la réactivité;
- **Mettre en place une cellule de réponse aux incidents (CSIRT)** avec des outils comme **TheHive, Cortex**.

6.2.2.5 Formation Continue et Certification

Les administrateurs doivent se former en continu pour rester à jour sur les nouvelles menaces et techniques de défense.

Les formations et certifications recommandées sont :

- **Sécurité Système et Réseau** :
    - **Certified Ethical Hacker (CEH);**
    - **CompTIA Security+;**
    - **Cisco CCNP Security.**
- **Gestion des Incidents et SOC** :
    - **Certified Incident Handler (GCIH);**

    - **Certified Information Systems Security Professional (CISSP).**
- **Cloud et Infrastructure Sécurisée** :
    - **AWS Certified Security - Specialty;**
    - **Microsoft Certified: Azure Security Engineer Associate.**

Une participation aux communautés et veille sécurité pour :

- **Suivre les alertes de sécurité** (ex. CERT, CVE, NIST);
- **Participer aux forums spécialisés** (ex. SANS, Offensive Security);
- **Tester des outils et techniques en environnement contrôlé (Lab)** avec **Kali Linux, Metasploit, CTF (Capture The Flag)**.

6.2.3 Sensibilisation des Utilisateurs Finaux

Les utilisateurs finaux représentent souvent le maillon faible de la sécurité informatique. Une simple négligence, comme l'utilisation d'un mot de passe faible ou le clic sur un lien malveillant, peut compromettre l'ensemble du système d'information. La sensibilisation des utilisateurs est donc une **composante essentielle** des mécanismes de protection.

6.2.3.1 Compréhension des Enjeux de Sécurité

Les utilisateurs doivent comprendre les **principales menaces** auxquelles ils sont exposés au quotidien :

- **Hameçonnage (Phishing)** : emails frauduleux incitant à divulguer des informations sensibles;
- **Ingénierie sociale** : manipulation psychologique pour obtenir des accès non autorisés;
- **Malwares et ransomwares** : exécution de fichiers malveillants entraînant une compromission du système;
- **Utilisation de mots de passe faibles** ou réutilisation des mêmes mots de passe sur plusieurs services;
- **Fuites de données** dues au partage involontaire ou à la mauvaise gestion des informations confidentielles.

Il est important de **vulgariser** ces concepts et d’expliquer leur **impact** sur la sécurité de l’entreprise et des données personnelles.

6.2.3.2 Adoption des Bonnes Pratiques de Sécurité

Les utilisateurs doivent être formés à appliquer des **bonnes pratiques** simples mais efficaces.

Pour la gestion des mots de passe, il faut :

- **Utiliser des mots de passe complexes** avec au moins 12 caractères, mélangeant lettres, chiffres et symboles;
- **Ne pas réutiliser les mots de passe** sur plusieurs comptes;
- **Utiliser un gestionnaire de mots de passe** (ex. Bitwarden, 1Password, KeePass);
- **Activer l'authentification à double facteur (2FA/MFA)** sur les comptes critiques (email, VPN, services cloud).

Pour la sécurité des emails et de la navigation, il faut :

- **Ne jamais cliquer sur des liens suspects** dans les emails ou messages instantanés;
- **Vérifier l'adresse de l'expéditeur** avant de répondre à une demande sensible;
- **Se méfier des pièces jointes non sollicitées**, même si elles proviennent d'un contact connu;
- **Utiliser des navigateurs sécurisés** et mettre à jour régulièrement les extensions.

Pour la protection des données sensibles, il faut :

- **Ne pas partager d'informations sensibles** via des plateformes non sécurisées (WhatsApp, email personnel);
- **Vérifier les paramètres de confidentialité** sur les réseaux sociaux et les services en ligne;
- **Utiliser des connexions chiffrées (VPN, HTTPS)** lors de l'accès à des données professionnelles à distance;
- **Ne pas stocker des fichiers sensibles sur des clés USB non sécurisées ou des services cloud publics.**

Pour la sécurisation des postes de travail et des périphériques, il faut :

- **Verrouiller son poste de travail** en cas d'absence;
- **Ne pas installer de logiciels non autorisés** sur son ordinateur professionnel;

- **Éviter les connexions aux réseaux Wi-Fi publics** ou utiliser un **VPN** en cas de nécessité;
- **Ne pas utiliser de supports amovibles inconnus (clés USB trouvées, disques durs externes non vérifiés).**

6.2.3.3 Mise en Place de Programmes de Sensibilisation

Pour la mise en place de programmes de sensibilisation, Il faut :

- **Organiser des sessions de formation régulières** sur la cybersécurité;
- **Mettre en place des campagnes de sensibilisation** (affiches, emails informatifs, guides de bonnes pratiques);
- **Utiliser des plateformes d'apprentissage interactives** (ex. KnowBe4, CyberSec Awareness, Google Jigsaw);
- **Simuler des attaques et exercices pratiques;**
- **Créer un point de contact sécurité.**

6.3 Documentation des Procédures et Bonnes Pratiques

La documentation joue un rôle crucial dans la gestion proactive de la sécurité applicative. Une approche bien structurée permet aux équipes de développement, d'exploitation et de sécurité d'adopter des pratiques cohérentes et d'assurer une protection efficace contre les menaces. Cette section met l'accent sur l'élaboration de guides et de politiques de sécurité pour encadrer les pratiques des parties prenantes.

6.3.1 Élaboration d'un Guide de Sécurité pour les Développeurs

Les développeurs sont en première ligne lorsqu'il s'agit d'assurer la sécurité des applications. Un guide de sécurité bien documenté leur permet d'intégrer les bonnes pratiques dès la phase de conception :

- **Principes fondamentaux du développement sécurisé :**
    - Intégration des principes de **"Security by Design"** et **"Privacy by Design"**;
    - Respect des standards de codage sécurisé (OWASP, NIST, ISO 27001);
    - Implémentation du principe du **moindre privilège** pour limiter les accès aux ressources sensibles.

- **Bonnes pratiques de codage sécurisé :**
    - Éviter les injections SQL, XSS, CSRF et autres vulnérabilités courantes;
    - Utilisation de fonctions de hachage et de chiffrement pour protéger les données sensibles;
    - Validation stricte des entrées utilisateur pour éviter toute exploitation malveillante.
- **Check-list de sécurité avant la mise en production :**
    - Vérification des dépendances et bibliothèques pour détecter d'éventuelles failles;
    - Mise en place de mécanismes de journalisation et de surveillance des activités suspectes;
    - Exécution de tests de sécurité automatisés (SAST, DAST).

6.3.2 Documentation des Procédures de Sécurisation des Infrastructures

Les administrateurs systèmes et réseaux doivent disposer de documents de référence pour garantir la sécurité des infrastructures sous leur responsabilité :

- **Durcissement des serveurs et bases de données :**
    - Désactivation des services inutiles et limitation des ports ouverts;
    - Configuration des pare-feu et des règles de filtrage du trafic;
    - Chiffrement des communications avec TLS/SSL et SSH.
- **Gestion des accès et des privilèges :**
    - Application de l'authentification multifactorielle (MFA);
    - Restriction des privilèges d'accès aux ressources critiques;
    - Rotation et mise à jour régulière des mots de passe et clés d'accès.
- **Surveillance et audit des infrastructures :**
    - Utilisation d'outils de monitoring (Prometheus, ELK, Wazuh) pour détecter les anomalies;
    - Configuration des alertes en cas d'activités suspectes;
    - Réalisation régulière d'audits de sécurité et de tests d'intrusion.

6.3.3 Mise en Place d'une Politique de Gestion des Vulnérabilités

La gestion des vulnérabilités est un élément clé de la sécurité applicative. Une politique claire permet d'assurer une réaction rapide face aux menaces :

- **Procédures de gestion des incidents de sécurité :**
    - Définition des rôles et responsabilités en cas d'incident;
    - Plan de réponse aux incidents incluant les actions correctives et préventives;
    - Communication interne et externe en cas de compromission des données.
- **Mise à jour et correction des failles de sécurité :**
    - Surveillance des CVE (Common Vulnerabilities and Exposures) et mise à jour régulière des logiciels;
    - Automatisation des correctifs avec des outils comme **Dependabot, WSUS, ou Ansible**;
    - Application de correctifs de sécurité dès leur publication pour limiter les risques d'exploitation.
- **Utilisation d'outils d'analyse et de suivi des vulnérabilités :**
    - Analyse régulière du code avec **SonarQube, OWASP ZAP, Snyk**;
    - Scans de vulnérabilités réseau avec **Nmap, Nessus, OpenVAS**;
    - Mise en place de tableaux de bord pour suivre l'état de la sécurité et les correctifs en attente.

6.4 Conclusion

La sécurité applicative ne peut être efficace sans une approche globale intégrant des mécanismes de protection adaptés, une sensibilisation continue des parties prenantes et une documentation rigoureuse des bonnes pratiques. Ce chapitre a mis en évidence l'importance de la prévention à travers l'adoption de standards de développement sécurisé, la configuration optimale des infrastructures et l'application de politiques de gestion des vulnérabilités.

L'implication de tous les acteurs, développeurs;administrateurs systèmes et réseaux;ainsi que les utilisateurs finaux, est essentielle pour réduire les risques de compromission et assurer une protection robuste contre les menaces. La formation continue et la mise à jour régulière des procédures de sécurité doivent être considérées comme un processus dynamique, en constante évolution pour répondre aux nouvelles vulnérabilités et techniques d'attaque.

En mettant en place ces bonnes pratiques, les organisations peuvent améliorer la résilience de leurs systèmes et garantir une meilleure protection des données et des services applicatifs. La sécurité ne doit pas être perçue comme une contrainte, mais comme un élément fondamental garantissant la pérennité et la confiance dans les applications et infrastructures informatiques.

Conclusion Générale

Avec l'essor des technologies numériques, la sécurité des applications web est devenue un enjeu majeur pour les entreprises et les organisations. La multiplication des cyberattaques ciblant les plateformes en ligne met en évidence la nécessité d'adopter des stratégies de protection robustes pour garantir l'intégrité, la confidentialité et la disponibilité des données.

Dans ce mémoire, nous avons mené une **étude des vulnérabilités existantes dans l'application Easy Compta**, une plateforme destinée à la gestion comptable. En nous appuyant sur des méthodologies éprouvées, incluant des **tests d'intrusion, des analyses statiques du code ainsi que des outils spécialisés**, nous avons pu identifier des failles de sécurité potentielles et évaluer leur impact sur le système.

L'analyse des vulnérabilités nous a permis de comprendre les principales menaces auxquelles l'application est exposée, notamment celles répertoriées dans l'**OWASP Top 10**. Face à ces risques, nous avons proposé des **mécanismes de protection adaptés**, tels que le **renforcement de l'authentification, la sécurisation des entrées utilisateur, l'amélioration des politiques d'accès et l'intégration de solutions de surveillance** pour détecter et prévenir les menaces en temps réel.

Dans cette optique, plusieurs **perspectives d'amélioration** peuvent être envisagées : Comment l'intelligence artificielle peut-elle être intégrée dans l'application pour détecter automatiquement les menaces et vulnérabilités en temps réel ? Quels outils et pratiques DevSecOps peuvent être utilisés pour renforcer la sécurité de l'application tout au long de son cycle de vie ? En quoi la blockchain pourrait-elle contribuer à renforcer la protection des données sensibles dans une application web comme Easy Compta ?

En conclusion, ce travail a permis d'améliorer la sécurité de l'application Easy Compta tout en mettant en lumière les **défis et enjeux liés à la cybersécurité des applications web**. Nous espérons que cette étude contribuera à une **meilleure prise de conscience des bonnes pratiques de développement sécurisé** et encouragera l'adoption de stratégies de protection adaptées aux nouvelles menaces numériques.

Webographie et Bibliographie

Webographie

[W1] Qu'est ce qu'un vps **consulté le 05 janvier 2025**

[W2] Fonctionnement d'un vps **consulté le 05 janvier 2025**

[W3] VPS **consulté le 09 mars 2025**

[W4] Serveur dédié **consulté le 21 janvier 2025**

[W5] Hébergement partagé **consulté le 21 janvier 2025**

[W6] NMAP **consulté le 01 mars 2025**

[W7] Fonctionnement de nmap **consulté le 01 mars 2025**

[W8] Masscan **consulté le 01 mars 2025**

[W9] Comparaison Nmap et Masscan **consulté le 01 mars 2025**

[W10] Unicornscan **consulté le 01 mars 2025**

[W11] Fonctionnalités de Unicornscan **consulté le 01 mars 2025**

[W12] Qu'est ce que OWASP ZAP **consulté le 01 mars 2025**

[W13] ZAP **consulté le 01 mars 2015**

[W14] Burp Suite **consulté le 01 mars 2025**

[W15] Nessus **consulté le 01 mars 2025**

[W16] Avantages de Nessus **consulté le 01 mars 2025**

[W17] UptimeRobot **consulté le 05 août 2024**

[W18] Présentation de UptimeRobot **consulté le 25 février 2025**

[W19] Prometheus **consulté le 05 août 2024**

[W20] Nagios **consulté le 28 février 2025**

Bibliographie

[B1] https://hal.science/hal-00736013 - Présentation des failles de sécurité - Guillaume Harry

[B2] https://docs.stackhawk.com/vulnerabilities.html - les vulnérabilités des applications web

[B3] https://scotthelme.co.uk/hardening-your-http-response-headers/ - Renforcer les en-tetes de sécurité

[B4]https://www.justegeek.fr/activer-len-tete-hsts-sur-apache-pour-proteger-son-site-web - activation de HSTS

[B5] GUELMI_ZAID_F5 - Mémoire de Fin d'études de Master - Juin 2022 https://dspace.univ-guelma.dz/jspui/bitstream/123456789/13042/1/GUELMI_ZAID_F5.pdf

[B6] https://www.f5.com/fr_fr/glossary/web-application-security - Qu'est-ce que la sécurité des applications Web ?

[B7] https://arxiv.org/abs/2204.07107 - étude sur les vulnérabilités des applications Web

[B8] TheseMakiou - Abdelhamid MAKIOU - Doctorat ParisTech - Sécurité des applications WEB - https://pastel.hal.science/tel-01668540/file/TheseMakiou.pdf