République Algérienne démocratique et populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences& technologie Blida 1

Faculté des Sciences

Département de l'informatique

Filière : Informatique

Spécialité : Systèmes de sécurité informatique « SSI »

Projet de fin d'étude pour l'obtention
Du diplôme de Master en SSI

Thème : Audit d'une application Web JBoss

Thème Proposé par :
Naftal SPA

Réalisé par : OUKACI Fella
BENALIA Abdelwahab

Présenté devant les membres du jury :

Mme.OUKID Saliha	Professeur	Présidente du jury
Mme.OUKID Lamia	Maitre de conférences	Examinatrice
Mme.DAOUD Hayat	Maitre assistante	Promotrice
Mme.MAIZ Amel	Docteur	Encadreur

Année universitaire : 2022/2023

Résumé

Les applications web ont connu une évolution exponentielle pendant les dernières décennies, cela est dû à plusieurs raisons parmi les quelles leurs facilité d'utilisation, leur accessibilité et bien plus, les services fournis par ces applications sont devenus indispensable dans la vie quotidienne de beaucoup de personnes. Pour cela le Web est aujourd'hui le vecteur d'attaque le plus prisé des attaquants, avec des milliers d'utilisateurs et de nombreuses d'application Web à disposition le terrain est doté d'actions malveillantes illimitées. C'est pourquoi il est indispensable et primordial de consacrer des efforts et du temps pour sécuriser nos applications Web. L'audit de sécurité est l'une des techniques essentielles dans le processus de sécurisation.

Notre projet consiste d'abord à réaliser un audit de sécurité sur une des applications Web de la structure d'accueil Naftal SPA et établir par la suite un rapport d'audit bien détaillé. Par la suite, dans notre solution, nous proposons d'implémenter une application qui permet la gestion facile de l'audit de sécurité.

Mots clés : Sécurité des applications Web, Audit de sécurité, scanner Web, Vulnérabilité.

Abstract

Web applications have experienced exponential growth over the past decades, due to several reasons including their ease of use, accessibility, and many more. The services provided by these applications have become indispensable in the daily lives of many people. Therefore, the Web has become the most preferred attack vector for attackers. With thousands of users and thousands of web applications available, the field is filled with unlimited malicious actions. That's why it is essential and crucial to dedicate efforts and time to secure our web applications. Security auditing is one of the essential techniques in the security process.

Our project aims to conduct a security audit on one of the web applications of Naftal SPA, our hosting organization, and subsequently produce a detailed audit report. Additionally, our solution proposes the implementation of an application that facilitates the easy management of security audits.

Keywords: Web Application Security, Security Audit, Web Scanner, Vulnerability.

ملخص

تطورت تطبيقات الويب بشكل هائل خلال العقود الأخيرة، وذلك بسبب عدة أسباب من بينها سهولة الاستخدام والوصولية والمزيد. أصبحت الخدمات التي توفرها هذه التطبيقات لا غنى عنها في حياة الكثير من الأشخاص اليومية. ولذلك، أصبح الويب هو المجال الأكثر رغبة للمهاجمين، مع وجود الآلاف من المستخدمين وآلاف تطبيقات الويب المتاحة، يكون المجال مليء بالأعمال الخبيثة غير المحدودة. لذا، من الضروري والأساسي أن نكرس الجهود والوقت لتأمين تطبيقات الويب الخاصة بنا. يعد التدقيق الأمني أحد التقنيات الأساسية في عملية الأمان.

يهدف مشروعنا في البداية إلى إجراء تدقيق أمني على أحد تطبيقات الويب في هيكل الاستضافة لمؤسسة "نفطال" ومن ثم إعداد تقرير تدقيق مفصل علاوة على ذلك، نقترح في حلنا تنفيذ تطبيق يسهل إدارة التدقيق الأمني بسهولة.

كلمات مفتاحية : أمان تطبيقات الويب, تدقيق الأمان, ماسح الويب, ضعف أمني.

Remerciement

Nous tenons tout d'abord à exprimer notre gratitude envers Dieu le Tout-Puissant et Miséricordieux, qui nous a donné la force, le courage, la volonté et la santé pour accomplir ce travail.

Nous souhaitons exprimer notre profonde gratitude envers notre promotrice, Madame DAOUD Hayat, pour avoir accepté de nous encadrer et pour sa disponibilité tout au long de la réalisation de ce mémoire. Ses précieux conseils nous ont permis de mener à bien ce travail, et nous avons bénéficié de son expérience et de sa sagesse.

Nous tenons également à remercier notre encadreur, Madame MAIZ Amel, pour la confiance qu'elle nous a accordée en proposant ce travail. Son encadrement attentif et le temps précieux qu'nous a consacré ont été d'une grande aide durant la réalisation de ce projet.

Nous exprimons notre profonde gratitude envers Monsieur ERRAHMANI Billel et Monsieur RAFA Tarek pour leur précieuse aide. Leur contribution a été d'une valeur inestimable et a joué un rôle essentiel dans la réalisation de ce travail.

Nos sincères remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en l'examinant minutieusement et pour leurs suggestions qui ont enrichi notre mémoire. Enfin, nous souhaitons exprimer nos remerciements les plus sincères à toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire. Votre soutien et votre collaboration ont été d'une importance capitale pour nous, et nous sommes extrêmement reconnaissants.

Dédicaces

À mes chers parents, à qui aucun hommage ne pourrait être à la hauteur de leurs sacrifices. Que Dieu leur procure bonheur et longue vie. Votre petite fille qui vous aime.

À mes grandes mères Zoubida et Sabiha que j'aime énormément.

À ma sœur Dalila, son mari Smail et mon neveu Rayan pour leurs encouragements permanents, et leur soutien moral.

À tous mes amis en particulier Zyneb, Maroua, Sirine, Ichrak et Idriss et à tous les Apôtres pour leur présence et leur soutien.

À mon binôme Abdelwahab pour son travail et effort tout au long du projet.

À toute l'équipe Naftal qui nous ont aidé à réaliser ce travail Billel, Tarek, Abdelhafid, Fatiha et Khira.

Et à toute personne qui, de près ou loin, m'a apporté soutien, conseil ou réconfort.

Mille mercis.

Fella

Dédicace

Je dédie ce modeste travail

D'abord je remercie Allah de m'avoir donné le désir d'avancer et de surmonter tout obstacle. Merci de m'avoir montré du bien parmi tous les maux de m'avoir donné la force de continuer.

Un grand merci pour mes chers parents de m'avoir appris à surmonter mes peurs, merci de m'avoir tant donné pour être reconnaissant.

A mon père d'avoir toujours été mon allié, rien que je puisse dire ne peut vraiment exprimer ce que je ressens. Rien ne montrera toute la gratitude que j'ai pour vous. Merci beaucoup pour les ailes que vous m'avez données, pour m'avoir appris à me lever et à élargir mes horizons vers les cieux.

A ma mère qui m'as donné l'espoir et le courage chaque instant. Merci d'être toujours là à mes côtés. A celle qui dans une parole comprend une symphonie, qui mas entouré d'amour et d'affection. Je me sens tellement respecté et béni de vous avoir dans ma vie. Merci d'être la maman attentionnée, aimante et attentionnée que vous êtes.

A mes sœurs Ilhem, Dalila, Soumia, Nour el houda, Yousra et Nawel de prêter votre épaule quand j'en ai de besoin, pour tous les bons moments qu'on a passés ensemble. Je vous remercie pour votre encouragement et votre aide.

Mes dédicaces s'adressent également à mon frères Mohamed.

A toute ma famille maternelle et paternelle.

A mes amies Majid et Imad qui m'ont toujours soutenu, avec qui j'ai passé des moments inoubliables.

À toute l'équipe Naftal qui nous ont aidé à réaliser ce travail Billel, Tarek et Abdelhafid.

A mon binôme Fella et a tous mes professeurs du primaire à l'université.

Abdelwahab

Table de matière

Introduction générale 1
1 Chapitre 1 : Etat de l'art 3
Introduction 3
1.1 Sécurité informatique 3
1.1.1 Définition de la sécurité informatique 3
1.1.2 Objectifs de la sécurité informatique 3
1.2 Application web 4
1.2.1 Définition d'une application web 4
1.2.2 Evolution des applications web 4
1.2.3 Comment fonctionne une application web 5
1.2.4 Les avantages des applications web 6
1.2.5 Sécurité des application web 7
1.3 Scanner Web 7
1.3.1 Définition 7
1.3.2 Type de Scanner web 8
1.3.3 Approche de Scanner Web 8
1.4 Serveur JBoss 9
1.4.1 Définition d'un serveur web 9
1.4.2 Aperçu JBoss 9
1.4.3 Pourquoi JBoss 10
1.5 Architecture trois tiers 10
1.6 Audit de sécurité 11
1.6.1 Définition 11
1.6.2 Objectifs de l'audit de sécurité 12
Conclusion 12
2 Chapitre 2 : Audit de l'application 12
Introduction 12
2.1 Les outils utilisés 12
2.2 Réalisation et implémentation 16
2.2.1 Préparation de l'environnement 16
2.2.2 Scan automatique 19

2.2.3 Rapport d'audit 24
Conclusion 28
3 Chapitre 3 : Etude conceptuelle 29
Introduction 29
3.1 Présentation globale de l'application 29
3.2 Etude conceptuelle de notre application 30
3.2.1 Diagramme de cas d'utilisation 30
3.2.2 Digramme de séquence 31
3.2.3 Digramme d'état-transition : 33
Conclusion 33
4 Chapitre 4 : Implémentation et réalisation 34
Introduction 34
4.1 Environnement de développement 34
4.2 Implémentation de l'application 35
4.3 Tests et résultat 36
4.3.1 Authentification 36
4.3.2 Lancer un scan 37
4.3.3 Consulter un scan 39
Conclusion 41
Conclusion générale 42
Références 44
Annexe 46

Liste des Figures

Figure 1.1 : Fonctionnement d'une application web. 6
Figure 1.2 : Fonctionnement d'une architecture trois tiers 11
Figure 2.3 : Réponse Postman a une requête HTTP. 16
Figure 2.4 : Wildfly Management Console. 17
Figure 2.5 : Résultat de l'API de test. 18
Figure 2.6 : Permission BD 18
Figure 2.7 : Activation du protocole TCP/IP. 19
Figure 2.8 : adresse IP de la machine virtuel. 19
Figure 2.9 : Environnement de scan. 20
Figure 2.10 : Interface OWASP ZAP. 20
Figure 2.11 : Manage Add-ons. 21
Figure 2.12 : Import OpenAPI. 21
Figure 2.13 : Chargement des URIs. 22
Figure 2.14 : Ajouter Token. 22
Figure 2.15 : Active scan. 23
Figure 2.16 : Resultat. 23
Figure 2.17 : Path traversal vulnérabilité 26
Figure 2.18 : SQL Injection Vulnérabilité 27
Figure 3.19 : Schéma global de l'application. 29
Figure 3.20 : Diagramme de cas d'utilisation. 31
Figure 3.21 : Diagramme de séquence cas d'utilisation lancer un scan. 32
Figure 3.22 : Diagramme d'état transition 33
Figure 4.23 : Environnement de développement. 34
Figure 4.24 : page de Login. 37
Figure 4.25: Lancer un scan -Etape 1- 38
Figure 4. 26: Lancer un scan -Etape 2- 38
Figure 4. 27: Lancer un scan -Etape 3- 39
Figure 4. 28 : Consulter résultat de scan. 40
Figure 4. 29 : Consulter un rapport. 40
Figure 4. 30 : Consulter Instance. 41
Figure 31 : Organigramme Direction Centrale Système d'information. 47

Liste des Tableaux

Tableau 1.1 : comparaison entre l'approche active et passive. 9
Tableau 3.2 : Description du diagramme de cas d'utilisation..... 31
Tableau 4.3 : Langage de programmation et technologie utilisé 35

Liste des abréviations

UML	Unified Modeling Language
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
TCP/IP	Transmission Control Protocol/Internet Protocol
URL	Uniform Resource Locator
HTTP	Hypertext Transfer Protocol
CSS	Cascading Style Sheets
API	Application Programming Interface
JSON	JavaScript Object Notation
URI	Uniform Resource Identifier

Introduction générale

De nos jours le nombre et la fréquence des attaques sur toute sorte de site web augmente en vitesse exponentielle causant des dégâts financiers immense, des dénis de service, des pertes de données confidentielles et importantes et bien plus, afin de lutter contre cela et diminuer les risques et menaces potentielle au maximum, il faut prendre des mesures préventives.

C'est pour cela que les responsables de sécurité doivent être à jour et au courant des vulnérabilités présente dans leur application en lançant des scans et en utilisant des outils valable sur le marché ou même d'une façon manuelle ou en auditant l'application d'une façon régulière et fréquente et travailler à éliminer et corriger les vulnérabilités résultantes , Ces mesures de sécurité peuvent parfois prendre beaucoup de temps et d'effort à l'équipe chargé de la sécurité vu qu'il doivent se faire de façon périodique et régulière.

Et même parfois la lecture et l'analyse du fichier résultant du scan peut s'avérer difficile et compliqué surtout si le fichier est volumineux et il contient trop de détails, donc ce processus va devoir nécessiter beaucoup de temps.

Pour les raisons citées au-dessus et dans le cadre de notre stage effectuer au sein de l'entreprise Naftal SPA on nous a été demandé d'effectuer premièrement un scan en utilisant un outil de scan puissant et établir un rapport qui décrit les résultats d'une façon simplifiée et ne contient que les données nécessaires et importantes qui aide à effectuer la remédiation des vulnérabilités et failles résultantes.

Un audit de sécurité inclut plusieurs étapes et phases parmi lesquelles on compte la partie scan de l'application web. Afin de scanner correctement une application, l'auditeur doit faire appel à plusieurs outils de scan et donc passer par plusieurs étapes, pour obtenir à chaque fois les résultats de chaque étape qui servira de support de scan à la prochaine étape. Et tout cela dans le seul but de voir le résultat affiché à la fin.

Ce qui nous a poussé à réfléchir à une solution optimale et plus pratique pour l'auditeur, qui nécessitera moins d'étapes pour un même résultat final et qui fait gagner du temps.

La solution étant de concevoir « un outil d d'automatisation du processus d'audit » :

Les principaux avantages de l'application sont :

- La facilité d'utilisation : l'application est présentée sur une interface claire, accessible, et agréable quant à la manipulation.
- Résultat lisible et clair : l'application affiche les résultats obtenus d'une façon très simple et lisible ce qui permet aux auditeurs et responsables de la sécurité de gagner en temps.
- L'automatisation des processus redondants : l'application permet une gestion facile car elle comporte des fonctionnalités d'automatisation de quelques tâches.

Afin de bien mener notre étude, nous avons organisé notre manuscrit sur trois chapitres comme suit :

- Chapitre 1 : nous avons abordé les notions de base ainsi que les concepts clés de la sécurité informatique et les applications web ainsi les scanners web, les bases de données, le serveur JBoss, qu'est-ce que c'est une architecture trois tiers et enfin l'audit de sécurité.
- Chapitre 2 : intitulé étude conceptuelle ou nous allons tout d'abord présenter notre solution et on termine par l'étude conceptuelle de notre application en utilisant des diagramme UML.
- Chapitre 3 : Le chapitre 3 de notre travail a été consacré à la réalisation de l'audit d'une application web. Nous avons débuté ce chapitre en présentant les outils que nous avons utilisés pour mener à bien notre travail, ensuite, nous avons détaillé les différentes étapes que nous avons suivies pour réaliser notre scan de sécurité, enfin, nous avons élaboré un rapport détaillé de notre scan de sécurité.
- Chapitre 4 : Dans ce chapitre nous avons abordé en détail l'implémentation de notre application d'automatisation du processus d'audit, ensuite nous avons détaillé le fonctionnement de chaque module de l'application.

Chapitre 1 : Etat de l'art

Introduction

Avec l'avènement de l'ère numérique et de la connectivité, les applications web sont devenues des éléments indispensables de notre vie quotidienne. Elles facilitent les transactions commerciales, les interactions sociales, et offrent un accès rapide et pratique à une variété de services en ligne. Cependant, cette ubiquité numérique présente également des risques en matière de sécurité.

Dans ce chapitre nous allons explorer les concepts clés de la sécurité informatique dans le contexte des applications web, les Scanner Web, le serveur d'hébergement JBoss.

1.1 Sécurité informatique

1.1.1 Définition de la sécurité informatique

La sécurité informatique est l'ensemble des moyens et outils mis en œuvre pour limiter les vulnérabilités contre les menaces. L'objectif de la sécurité informatique est d'assurer que les ressources matérielles et/ou logicielles d'un système informatique sont uniquement utilisées dans les limites prévues et par des personnes légitimes.

En autre terme, La sécurité informatique apparaît comme tous moyens humains, matériels, juridiques, techniques et organisationnels mis en œuvre pour assurer la protection de l'information. Cette sécurité peut être assurée par une politique qui vise à garantir, rétablir et préserver le fonctionnement de l'information [1].

1.1.2 Objectifs de la sécurité informatique

On peut citer cinq principaux objectifs de la sécurité informatique :

- **La confidentialité :** Elle signifie que seules les personnes approuvées et autorisées peuvent accéder aux données en question.
- **La disponibilité :** Elle a pour but de garantir l'accès à un système, une ressource ou une donnée à tout moment.
- **L'intégrité :** l'objectif de l'intégrité est de garantir que les données n'ont subi aucun changement non autorisé soit en transit, à distance et en cours d'utilisation.

  Cela veut dire qu'ils n'ont pas changer depuis leur création

- **La non-répudiation :** Aucun acteur dans un système d'information peut nier ses actions sur ce dernier.
- **L'authentification :** Elle consiste à s'assurer que chaque personne est bien celle qui elle prétend de l'être.

1.2 Application web

1.2.1 Définition d'une application web

Une application web est un programme informatique qui est composé principalement d'un ensemble de pages web hyperliées entre elles, conçues pour être consultées à l'aide d'un navigateur Web, publiées par un propriétaire (une entreprise, une administration, une association, un particulier, etc.) et hébergées sur un ou plusieurs serveurs [2].

Les développeurs conçoivent des applications web pour une large diversité d'utilisateurs et de clients ce qui les rendent compatibles avec la majorité des ordinateurs et systèmes d'exploitation.

Les exemples les plus courants des applications web sont les plateformes de messageries, les sites de ventes en ligne…

1.2.2 Evolution des applications web

Depuis plusieurs années le web a connu une évolution exponentielle et il est passé par différentes phases et étapes avant d'arriver à ce qui l'est de nos jours.

Tout au début, les pages web se limitaient à des pages HTML qui étaient statiques et affichaient du contenu simple et non interactif (texte et quelques images et illustrations) et la seule interaction possible était de cliquer sur un lien hypertexte pour naviguer sur une autre page HTML.

Au fil du temps, les exigences des utilisateurs ont augmenté et les développeurs devaient absolument trouver une solution pour satisfaire les besoins des clients, c'est ici que les sites web dynamiques ont fait surface en introduisant en plus du HTML du code JavaScript, PHP et les bases de données …

- **Internet :** la première apparition d'internet remonte aux années 1970 par les deux informaticiens américains BOB KAHN et VINT CERF qui ont développé le TCP/IP qui est un ensemble de protocoles qui gouvernent la transmission des données sur un réseau initialement connu sous le nom d'ARPANET (Advanced Research Projects Agency Network).

  Internet est un système mondial de réseaux informatiques connectés entre eux et le Web est qu'un des services acheminés sur ces réseaux. Il s'agit d'un ensemble de documents HTML (HyperText Markup Language) et d'autres ressources, liés entre eux par des URL (Uniform Resource Locator), accessibles par des navigateurs Web, depuis des serveurs Web [3].

- **WWW (World Wide Web) :** il a été introduit en 1989 par le physicien britannique Tim Berners-Lee du CERN, son principal objectif était de pouvoir partager des informations avec d'autres physiciens du monde entier.

  En 1990 le WWW a vu le jour en introduisant les trois technologies fondamentales du web (HTML, URI, HTTP). Tim a aussi créé le premier navigateur web le WorldWideWeb.app.

  En 1993, le CERN a abandonné tous les droits de propriété intellectuelle sur le World Wide Web, le rendant libre d'utilisation pour tous, sans qu'aucune redevance ne soit due au CERN [3].
- **Web 1.0 :** En 1995 le nombre d'utilisateurs du web a atteint les 50 millions, environ 0.8% de la population mondiale. Les sites web à l'époque étaient statiques c'est-à-dire des pages HTML qui contiennent du texte relié par des hyperliens en utilisant le protocole HTTP (Hypertext Transfer Protocol).

  Un client envoie une requête HTTP en demandant la page (http://www.exemple.com/index.html) le serveur cherche cette page dans sa mémoire et la renvoie au client ensuite le navigateur interprète la réponse du serveur et affiche la page demandée, ceci est appelé l'architecture client-serveur.

  Le résultat affiché était le même pour tous les utilisateurs et n'était pas personnalisé donc il y avait un manque d'interactivité ce qui a mené les développeurs à introduire les sites web dynamiques et le web 2.0.
- **Web 2.0 :** c'était en 1999 que le mot Web 2.0 est apparu pour la première fois dans un article intitulé Fragmented Future écrit par Darcy DiNucci [4] , la fonction majeure de cette nouvelle génération du web est d'introduire les utilisateurs et de fournir de l'interaction avec le système, étant donné que les sites statiques ne permettent pas d'assurer ces fonctionnalités les sites dynamiques sont mis en place , Un site dynamique fournit un contenu unique aux utilisateurs chaque fois qu'ils consultent le site. Cela se fait par une combinaison de scripts du côté du client et du serveur.Les logiciels génèrent des pages web dynamiques avec les langages PHP, ASP …

Les sites web dynamiques ont permis la création et la révolution des applications web d'aujourd'hui tels que les réseaux sociaux et les sites d'E-commerce …

1.2.3 Comment fonctionne une application web

Les applications web d'aujourd'hui fonctionnent avec une architecture client-serveur.

- **Architecture côté client :** l'opération de récupération d'une application web commence par l'envoi d'une requête de la part du client en utilisant le protocole HTTP (S).

Le client émet une requête (i.e. Appelle une URL) pour demander une ressource au serveur. Exemple : http://www.exepmple.com/index.html . Ici il ne peut pas savoir si la réponse qui vas recevoir est statique (page HTML simple) ou dynamique (générée par une application web) [5].

- **Architecture côté serveur :** le serveur reçoit la requête du client qui demande une certaine application web, il cherche cette dernière parmi toutes ses pages et la renvoie au client.
  Le script côté serveur est responsable du traitement de données comme la sauvegarde des informations entrées dans un formulaire par exemple.
  Voici une figure qui explique le fonctionnement d'une application web d'une façon simplifiée :

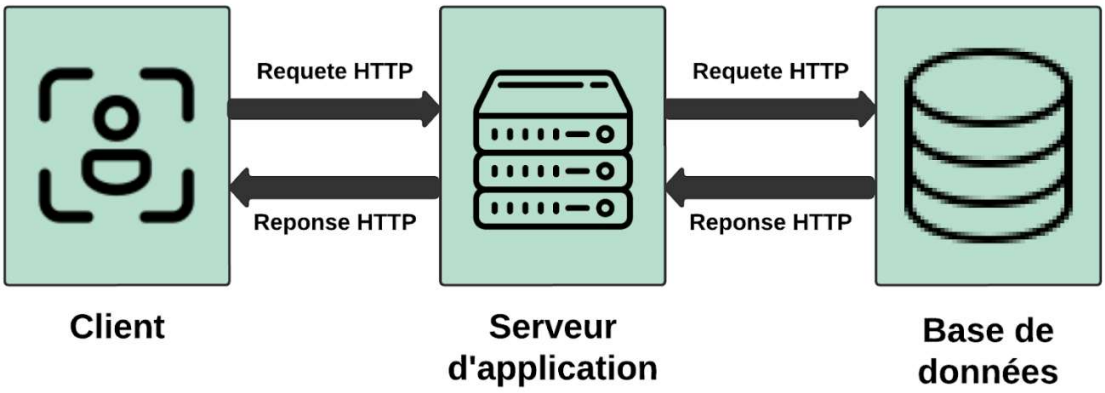

Figure 1.1 : *Fonctionnement d'une application web.*

1.2.4 Les avantages des applications web

- **L'accessibilité :** L'un des avantages majeurs d'une application web est qu'elle soit accessible et atteignable depuis n'importe où à n'importe quel moment, il suffit juste d'avoir une connexion internet et un ordinateur pour y accéder.
- **La compatibilité :** les applications web peuvent être interprétées sur plusieurs appareils électroniques (ordinateur, smartphone, tablette) quel que soit leurs systèmes d'exploitation en plus qu'ils n'ont pas besoin de beaucoup de ressources pour fonctionner il suffit tout simplement d'un navigateur web adéquat.
- **Facile à développer :** le cycle de développement d'une application web est relativement facile de nos jours vu la quantité de ressource disponible partout soit d'une façon payante ou même gratuite, il suffit d'une petite équipe d'informaticiens pour concevoir une application web simple et efficace qui répond au besoin des entreprises.

- **Facilité d'utilisation :** toute application web n'a besoin que d'un navigateur pour fonctionner, et aucune autre installation ou configuration n'est requise [6]**.**

1.2.5 Sécurité des application web

Comme nous l'avons vu précédemment les applications web semblent être impeccables et parfaites et qu'elles ne comportent aucune faute ou erreur, en fait c'est loin d'être le cas.

Comme tout système informatique, les applications web sont exposées à des vulnérabilités et menaces qui sont considérées comme la bête noire de ces dernières, une attaque fait perdre l'entreprise des grandes sommes d'argent, du temps et même la confiance des utilisateurs.

Puisque les applications web sont hébergées sur des serveurs distants et exposés au public en plus qu'elles contiennent, manipulent des données sensibles comme les numéros de carte bancaire … etc. Et les données doivent passer via internet avant d'atteindre le navigateur de l'utilisateur final, c'est pourquoi les applications web sont exposées à des risques et menaces beaucoup plus importantes et considérables. Les procédures et techniques de protection doivent être capables de détecter les actions suspectes et intrusions malveillantes.

Pour cela plusieurs méthodes sont valables comme l'utilisation de pare-feu d'application web (WAF) et beaucoup d'efforts sont déployés pour améliorer la barrière de sécurité et protéger les applications web.

1.3 Scanner Web

1.3.1 Définition

La numérisation des applications web consiste à effectuer des surveillances et des analyses automatisées pour vérifier les vulnérabilités ou les violations de sécurité. À mesure que les entreprises grandissent et que les sites web deviennent plus complexes, il est important d'identifier les failles sur votre site et de les corriger, quelle que soit leur importance. C'est pourquoi un scanner automatique d'applications web est un outil essentiel pour chaque site.

Avoir un scanner automatique permet d'analyser les éléments, la logique, les processus et les logiciels tiers qui composent votre site web. Ce faisant, le scanner identifie les problèmes tôt avant qu'ils ne s'aggravent ou que des pirates ne les exploitent. Certaines de ces vulnérabilités peuvent inclure des logiciels non patchés, un accès non autorisé, des attaques de type cross-site scripting (XSS), des attaques par injection, et bien d'autres encore [7].

On peut diviser les scanner Web en trois catégories majeures :

A. Scanner Black Box.
B. Scanner White Box.

C. Scanner Grey Box.

1.3.2 Type de Scanner web

A. Scanner White Box

Est une méthode d'évaluation de la sécurité des applications qui nécessite une connaissance de l'architecture, du code source et des détails internes de l'application.

Voici quelques caractéristiques importantes du scanner "White Box" :

- Le scanner White Box est autorisé à accéder au code source de l'application, ce qui lui permet d'effectuer une analyse en profondeur de l'application.
- Le scanner White Box utilise des techniques de scan statique pour analyser le code source et identifier les vulnérabilités potentielles.
- Étant donné qu'il a un accès complet à l'application, le scanner White Box peut effectuer des tests de vulnérabilité plus avancés et approfondis ce qui permet d'avoir des résultats plus pertinents et précis.

B. Scanner Black Box

- Le scanner de l'application teste l'état de la sécurité d'un point de vue extérieur.
- Explorer les perspectives de vulnérabilités d'un point de vue extérieur.
- Déduit que certaines vulnérabilités existent en envoyant les entrées à l'application et en faisant l'analyse des résultats.
- Pas d'accès au code source de l'application.

C. Scanner Grey Box

- La structure interne est partiellement connue.
- Les tests sont faits de la même manière que la méthode du scanner Black Box [8].

1.3.3 Approche de Scanner Web

Il existe deux approches principales de l'analyse des vulnérabilités : une approche passive et active.

- **Approche Passive :**

Est une méthode de recherche et collecte d'informations et de surveillance des réseaux et des systèmes sans toucher à leur fonctionnement normal. L'objectif principal d'un scan passif est d'observer et de recueillir des informations sur les données qui transitent sur le réseau.

- **Approche Active :**

Est une méthode d'évaluation de la sécurité des applications web qui implique l'interaction active avec les applications cibles. Un scan web actif envoie des requêtes et des données spécifiques aux applications web pour évaluer leurs vulnérabilités et leur comportement.

L'objectif majeur d'un scan web actif est d'identifier et chercher les vulnérabilités et les faiblesses des applications web, telles que les injections SQL, les cross-site Scripting (XSS).

Ici nous allons faire une comparaison rapide entre les deux approches en utilisant un tableau.

Tableau 1.1 : *comparaison entre l'approche active et passive.*

Approche Active	Approche Passive
Intrusive	Non-intrusive
Fournit des informations détaillés	Fournit des informations basics
Plus précis	Taux de faux positif élevé
Configuration complique	Configuration simple et facile
Long	Rapide

1.4 Serveur JBoss

1.4.1 Définition d'un serveur web

Un serveur web est une combinaison entre un serveur matériel et un autre logiciel qui travaillent ensemble.

- Le serveur matériel représente un sort d'ordinateur puissant qui possède des ressources physiques importantes qui stocke le serveur logiciel plus tous les documents composant un site web (les pages HTML, CSS, JS).
- Le serveur logiciel comprend plusieurs parties qui gèrent la manière dont les utilisateurs Web accèdent aux fichiers hébergés. Au moins, il s'agit d'un serveur HTTP (Un serveur HTTP est un logiciel qui comprend les URL) [9].

1.4.2 Aperçu JBoss

JBoss EAP (Enterprise Application Platform) est une plate-forme open source pour les applications Java hautement transactionnelles à l'échelle du Web. JBoss EAP combine les spécifications familières et populaires de Jakarta EE avec les dernières technologies, comme Eclipse MicroProfile, pour moderniser les applications Java EE traditionnel vers le nouveau monde de DevOps, du cloud, des conteneurs et des microservices.

JBoss EAP comprend tout ce dont un développeur a besoin pour créer, exécuter, déployer et gérer des applications Java d'entreprise dans plusieurs environnements, y compris des environnements sur site, virtuels et dans des clouds privés, publics et hybrides. JBoss EAP est basé sur le projet open source populaire WildFly [10].

1.4.3 Pourquoi JBoss

- **Certifié Java :** JBoss EAP plate-forme certifiée Java EE est conforme avec la majorité des spécifications de Java principalement Jakarta EE, Java SE (OpenJDK et OracleJDK) et Eclipse Microprofile.
- **Optimisation pour le Cloud :** JBoss EAP est idéal pour un déploiement simple et efficace dans quiconque environnement clouds virtuels, privés, publics et hybrides ainsi que les conteneurs.
- **Modulaire et léger :** JBoss EAP est une plateforme modulaire donc tout service est lancé ou démarré seulement au besoin ce qui améliore la vitesse du démarrage.
- **Sécurité :** JBoss EAP comporte un sous-système de sécurité appelé Elytron, il est utilisé pour configurer l'accès de gestion au serveur et également pour les applications déployées sur le serveur, il couvre les domaines suivants : Authentification, Autorisation, SSL/TLS, Stockage sécurisé des identifiants.
- **Gestion facile :** JBoss EAP peut être géré à partir de la CLI (command-line interface) cela permet une gestion, configuration et administration beaucoup plus facile et simple pour les administrateurs, ainsi il permet la gestion hors ligne des serveurs.
- **Productivité des développeurs :** JBoss EAP fait en sorte d'assurer la productivité et l'efficacité ainsi qu'il permet d'utiliser les dernières technologies disponibles sur le marché, les Frameworks web Jakarta EE tels que Spring, AngularJS, jQuery.
- **Open source :** JBoss EAP est un projet entièrement Open source et gratuit ce qui le rend très flexible [11].

1.5 Architecture trois tiers

Une architecture trois tiers appelée aussi architecture à trois niveaux ou architecture à trois couches, elle consiste à séparer une application en trois parties Individuelles chacune ses rôles et fonctionnalités. L'architecture est devisée comme cela :

- **La présentation des données :** c'est le premier niveau de ce modèle, il représente la partie IHM (Interface Homme Machine) ou tout ce qui est visuel menu, bouton, couleur, etc. il interagit directement avec l'utilisateur en affichant des données et en récupérant les entrées.

Les technologies couramment utilisé pour réaliser l'interface sont HTML, CSS, JavaScript ou des frameworks tel que Angular, react, etc.

- **Le traitement métier des données :** c'est le deuxième niveau, c'est ici que la partie logique de l'application est implémenté et le traitement et manipulation des données est effectué.
  Les technologies généralement dans cette couche sont les langages de programmation parmi eux PHP, Java, .NET, etc.
- **L'accès aux données persistantes :** elle représente la troisième et dernière couche de cette architecture, elle s'occupe de la manipulation, stockage, modification, récupération des données.
  Elle peut utiliser une base de données relationnelle tels que MySQL, SQL Server ou une base No SQL comme Mongo DB ou Oracle No SQL Database.

Cette architecture offre plusieurs avantages et fonctionnalités supplémentaires par rapport aux architectures classiques tels que la réutilisabilité, l'évolutivité, facilité de maintenance et elle est recommandée dans les applications modernes d'aujourd'hui.

La figure représente le fonctionnement d'une architecture trois tiers :

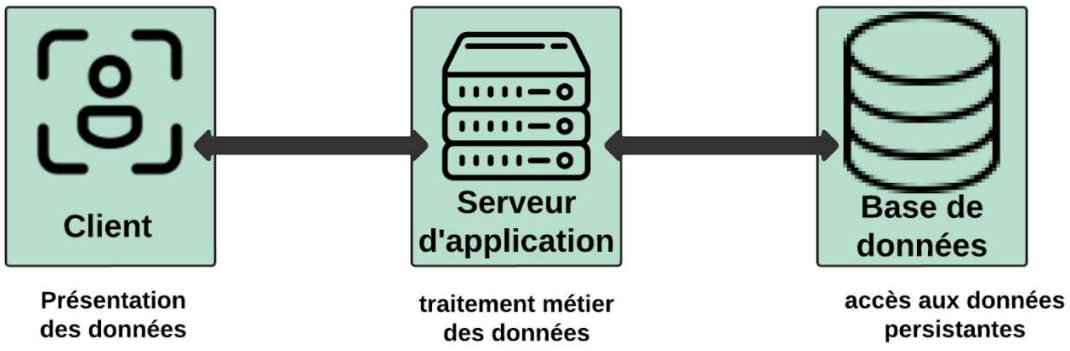

Figure 1.2 : Fonctionnement d'une architecture trois tiers

1.6 Audit de sécurité des applications web

1.6.1 Définition

La sécurité des applications Web est le processus de sécurisation des données confidentielles stockées en ligne contre tout accès et modification non autorisés. Ceci est accompli en appliquant des mesures politiques strictes. Les menaces de sécurité peuvent compromettre les données stockées par une organisation. Des pirates informatiques malveillants tentent d'accéder à des informations sensibles [12].

Un audit de sécurité des applications Web est une évaluation complète de vos applications Web et de leur infrastructure associée. Cela peut aller d'un site Web à des systèmes intranet d'entreprise, des API ou des systèmes de commerce électronique. Tout composant du Web relève de ce parapluie. [13]

1.6.2 Objectifs de l'audit de sécurité

La sécurité des applications Web a pour objectif d'identifier les éléments suivants :

- Atouts essentiels de l'organisation.
- Utilisateurs authentiques qui peuvent accéder aux données.
- Niveau d'accès fourni à chaque utilisateur.
- Diverses vulnérabilités pouvant exister dans l'application.
- Criticité des données et analyse des risques sur l'exposition aux données.
- Mesures correctives appropriées. [12]

1.7 Conclusion

Dans ce chapitre nous avons abordé tous les concepts clés dont nous allons avoir besoin dans la réalisation du projet dont la définition de la sécurité informatique ainsi que ces objectifs nous avons aussi parlé sur les applications web, le serveur JBoss, c'est quoi un audit d'application web.
L'audit de l'application web sera décrit en détail dans le chapitre suivant.

Chapitre 2 : Audit de l'application

Introduction

Le présent chapitre se concentre sur l'audit de notre application web, une étape cruciale dans le processus de sécurisation, nous aborderons également la préparation de l'environnement de travail nécessaire pour effectuer l'audit, en mettant en place les outils et les configurations requises, nous passerons en revue le scan automatique réalisé Enfin, nous examinerons le rapport d'audit.

2.1 Les outils utilisés

- **OWASP ZAP**

Zed Attack Proxy est un outil de sécurité des applications web gratuit et open-source géré par des volontaires, il est conçu pour effectuer des scans active et simuler des attaques sur l'application web cible et essayer de trouver des vulnérabilités.

ZAP offre une variété de fonctionnalités pour les tests de sécurité automatiques et manuels en particulier des scans actifs, passifs, spidering, fuzzing, tests d'authentification et de gestion de session.
OWASP ZAP comprend une interface graphique facile à manipuler et très complète ce qui le rend un outil assez puissant pour les débutants et les professionnels à la fois, une API vigoureuse et l'opportunité de l'inclure et l'associer à d'autres outils.

- **Wildfly**

Précédemment nommé JBoss application Server ou JBoss AS est un serveur d'application java développé par Red Hat, il sert à déployer et héberger des applications Java Enterprise Edition (EE) en plus d'être open source et totalement gratuit.
Wildfly assure la gestion de différentes spécifications Java EE principalement les servlets, les Java Server pages (JSP), les entreprises JavaBeans (EJB) ect.
Wildfly est très léger et souple d'utilisation, il comporte une console d'administrateur qui permet de générer le déploiement depuis une interface graphique simple et complète. Il est totalement adaptable aux besoins spécifiques en plus de la possibilité de faire appel à des différentes API et framework.

- **Postman**

Est un outil qui permet aux développeurs la création, le test, le débogage d'une API (Application Programming Interface), il peut être utilisé depuis une application bureau téléchargeable ou bien en ligne.
Le fonctionnement de Postman consiste à envoyer des requêtes HTTP et intercepter les réponses et afficher les détails, une requête peut être changé par le développeur en changeant l'en-tête de cette dernière.
Pour permettre plus de flexibilité et la détection efficace des erreurs. Postman permet aussi de collaborer avec d'autres personnes ce qui facilite le travail de l'équipe de développement, Postman propose un large choix de format de données (JavaScript, JSON, HTML, XML), en plus de contenir des fonctionnalités qui permet d'automatiser des tâches répétitives.
A la fin on peut dire que Postman est un outil très utile et puissant pour le développement d'une API ce qui le rend presque indispensable durant tout le cycle de vie d'une API en développement et même en production.

- **SQL Server**

Structured Query Language Server est un système de gestion de base de données relationnelle (SGBDR) de chez Microsoft. Il sert principalement à la manipulation des données structurées dont le stockage, la gestion et le traitement de ces dernières.

SQL Server prend en charge plusieurs fonctionnalités et voici quelques-unes :

1. **Gestion de Base De Données :** il permet de stocker, manipuler et développer des BD de petites comme de grandes tailles en utilisant ses fonctionnalités intégrées comme les tables, les vues, les triggers, les index … d'une façon rapide et simple
2. **T-SQL :** ou transact-SQL est une extension de programmation de Sybase et Microsoft ensemble. Le T-SQL est basé sur le SQL classique mais avec des fonctionnalités supplémentaires notamment la programmation procédurale …
3. **SQL Server Management Studio (SSMS) :** l'interface utilisateur graphique (GUI) de SQL Server facilite aux administrateurs de base de données leurs travail et les a fait gagner du temps en offrant une gamme très riche de fonctionnalités en seulement un clic
4. **Sécurité :** SQL Server prête beaucoup d'importance considéré l'aspect de la sécurité comme essentiel et primordiale en mettant en œuvre un large choix de fonctionnalités liées à la sécurité tels que l'authentification, l'autorisation, le chiffrement …

SQL Server comporte bien plus de fonctionnalités que celle listé au-dessus, il est l'un des SGBD les plus utilisés dans différents domaines tels que le web, l'informatique décisionnelle ainsi que sa flexibilité et rigueur le rendent un compétiteur très fort sur le marché.

- **Kali Linux**

Kali Linux (anciennement connu sous le nom de BackTrack Linux) est une distribution Linux open-source basée sur Debian conçue pour élaborer des tests de pénétration avancés et à l'audit de sécurité. Pour cela, il possède des outils, des configurations et des automatisations.

Kali Linux comprend des modifications spécifiques au domaine ainsi que plusieurs centaines d'outils dédiés aux diverses tâches de sécurité de l'information, telles que les tests d'intrusion, la recherche en sécurité, l'informatique judiciaire, l'ingénierie inverse, la gestion des vulnérabilités et les tests de l'équipe rouge [14].

Le but principal de Kali Linux est de fournir aux professionnels de cyber sécurité et aux utilisateurs une plateforme robuste qui regroupe tous les outils et programmes nécessaires au bon déroulement de quelconque projet (exploitation de vulnérabilité, test de sécurité, sécurisation des systèmes …) dans le domaine de la sécurité informatique dans un seul endroit.

- **VM ware Workstation**

Est un logiciel de bureau développé par VM Ware qui permet aux utilisateurs de créer et d'exécuter des machines virtuelles.

VM ware Workstation offre beaucoup de fonctionnalités et d'avantages voici quelques-unes :

1. **Exécuter plusieurs machines virtuelles sur un seul PC** : VM ware Workstation permet de créer et d'exécuter plusieurs machines virtuelles sur un seul PC physique. Chaque machine virtuelle est totalement indépendante des autres ce qui signifie que chacune a sa propre configuration, cela permet aux développeurs d'effectuer des tests sur des environnements différents.
2. **Support des systèmes d'exploitation :** VM Ware Workstation prend en charge la majorité des systèmes d'exploitation notamment Windows, linux et mac OS ainsi que leurs différentes distributions.
3. **Test** : VM Ware Workstation est utilisé en premier lieu pour permettre aux programmeurs d'effectuer des tests sur les logiciels dans des environnements isolés.
   Il permet aux développeurs de tester leurs applications en testant la compatibilité avec différents systèmes d'exploitation ce qui permet la réparation et la correction d'éventuelles erreurs.
4. **Sandbox et sécurité :** VM Ware Workstation offre la possibilité de créer des environnements isolés ou ce que on appelle des Sandbox, ces dernières permettent d'exécuter des applications dangereuses sans prendre le risque d'endommager les ressources physiques.

- **Ubuntu 20.04 LTS**

Ubuntu 20.04 connu aussi sous le nom "Focal Fossa" est l'une des versions Ubuntu les plus répandu grâce à sa flexibilité et stabilité Voici quelques caractéristiques et fonctionnalités clés d'Ubuntu 20.04 :

1. **Long terme support (LTS)** : Ubuntu 20.04 est une version LTS, cela veut dire qu'elle va bénéficier d'un support pendant un long terme d'une durée de cinq ans, elle recevra pendant ce temps des mises à jour liée à la sécurité plus celles qui contient des améliorations et des corrections.
2. **Interface utilisateur :** Ubuntu 20.04 est doté de GROME, un environnement desktop conçu pour donner une expérience plus agréable et simple même aux personnes non familières avec l'utilisation d'un PC.
3. **Linux Kernel 5.4 :** Ubuntu 20.04 utilise le noyau linux 5.4 qui dispose de plusieurs améliorations par rapport aux versions antérieures tels que l'amélioration du support hardware.

4. **Installation simplifiée :** l'installateur Ubuntu 20.04 offre une simplicité d'utilisation, la flexibilité et la possibilité de personnalisation selon les besoins spécifiques de chaque utilisateur.

2.2 Réalisation et implémentation

2.2.1 Préparation de l'environnement

L'architecture que nous avons adopté dans ce projet est une architecture trois tiers ou nous avons séparé les trois couches comme suit :

- **Couche présentation des données**

C'est la première couche de l'architecture trois tiers, elle est responsable du visuel de l'application, dans le cadre de notre projet nous n'avons pas pris en considération cette couche, nous nous sommes concentrés sur la couche traitement métier des données puisqu'elle ne fait pas parti des besoins de la structure d'accueil.

Pour tester le bon fonctionnement de l'application et afin d'observer les résultats obtenus par cette dernière nous avons utilisé l'outil de débogage Postman (voir la définition dans la partie outils utilisés), il suffit de lui fournir l'URL qui est composé de l'adresse IP où se trouve l'application suivi par l'URI qu'on veut interroger, le résultat obtenu représente la réponse rendu par l'API.

En interrogeant une API le code HTTP rendu doit être doit être égale à 200 est un code d'état dans le protocole http qui indique une demande réussie.

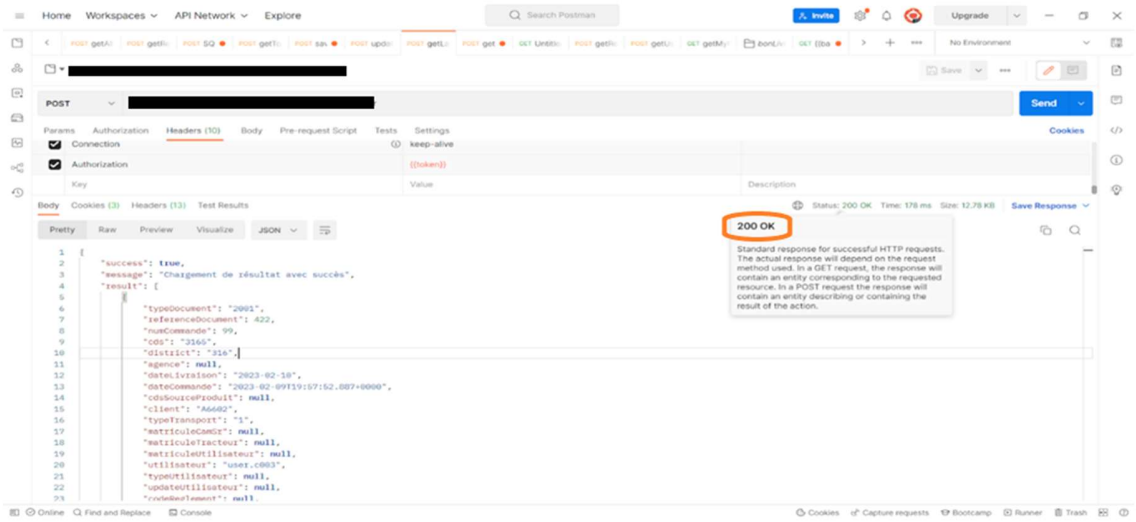

Figure 2.3 : Réponse Postman a une requête HTTP.

- **Couche traitement métier des données**

L'application sur laquelle nous avons réalisé notre scan est déployée dans une machine Ubuntu 20.04 sous le serveur web JBoss (actuellement Wildfly), la partie de déploiement est relativement facile sous JBoss et ne nécessite que très peu d'étapes.

Après l'installation du serveur WildFly du site web officiel de Red Hat, il faut accéder à la Management console avec le port 9990 et cliquer sur la rubrique "Deployments" → "Add Deployment " ensuite ajouter le WAR où se trouve le code source de notre application comme le montre la figure suivante :

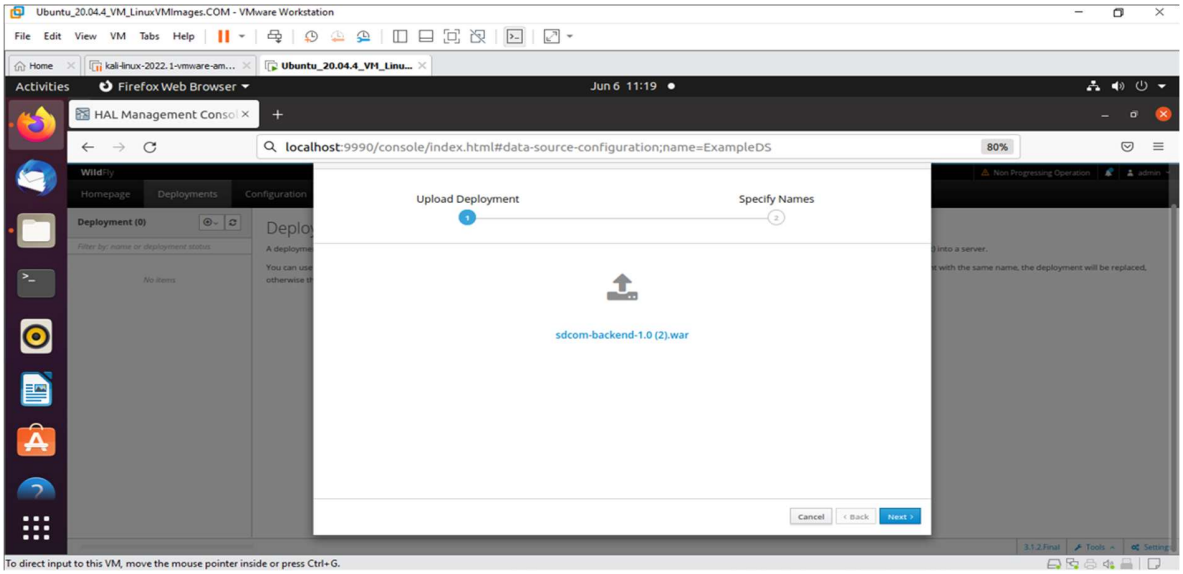

Figure 2.4 : *Wildfly Management Console.*

La partie déploiement est bientôt terminée, on poursuit en spécifiant le nom de ce déploiement et le tour est joué, notre application est maintenant prête pour être utilisée.

On peut vérifier que le serveur WildFly marche sans faute en tapant la commande *sudo systemctl status wildfly*, cette commande permet de s'assurer que notre serveur est fonctionnel.

Et pour vérifier que notre application a bien était déployé on tape le lien de l'API dédiée au test dans notre navigateur et on est censé avoir le résultat suivant.

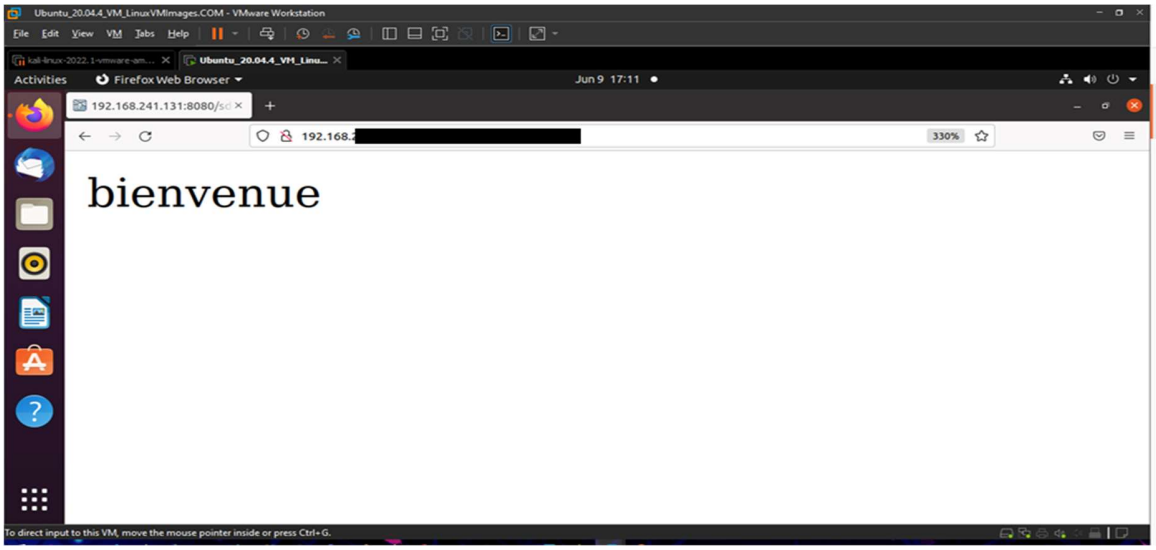

Figure 2.5 : Résultat de l'API de test.

- **Couche d'accès aux données persistantes**

Cette couche est responsable du traitement et de la manipulation des données. Dans notre cas, nous avons opté pour SQL Server 2019 comme SGBD (Système de Gestion de Base de Données).

La première étape étant de configurer la base de données dans nos machines locales qui a été fourni par la structure d'accueil dont les tables ainsi que les données.

Ensuite donner les permissions nécessaires à l'utilisateur pour pouvoir accéder au contenu des tables.

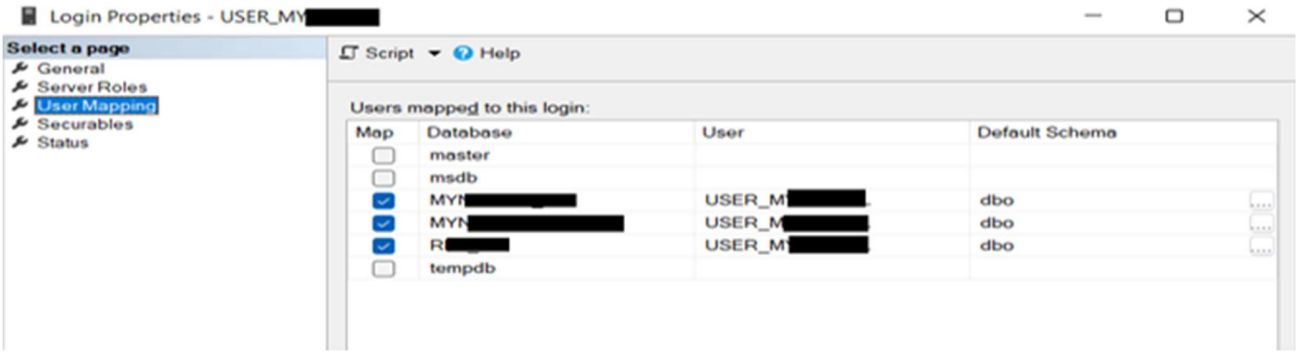

Figure 2.6 : Permission BD

La prochaine étape est de vérifier que le statut du protocole TCP/IP est active dans 'SQL Server Configuration Manager'

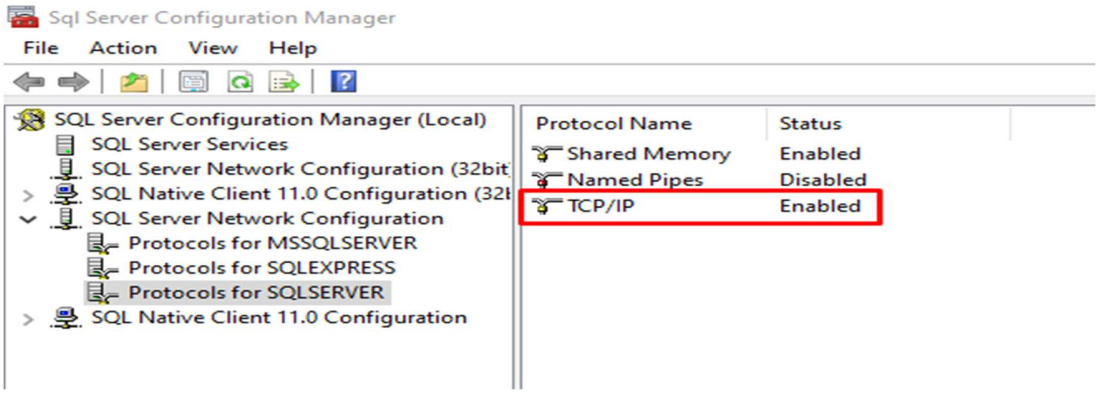

Figure 2.7 : Activation du protocole TCP/IP.

Ensuite on ajoute l'adresse IP de la machine virtuel où se trouve l'application pour que la connectivité entre les deux machines réussisse.

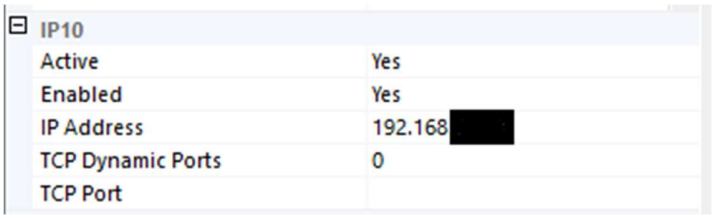

Figure 2.8 : adresse IP de la machine virtuel.

Maintenant que l'environnement est prêt et tout marche parfaitement bien on peut passer à l'étape qui suit qui est la réalisation du scan.

2.2.2 Scan automatique

Cette étape représente le cœur de notre étude dont le résultat est basé sur, Le scan a été effectué à l'aide de l'outil décrit précédemment dans le chapitre 2 « OWASP ZAP » qui est le scanneur open-source le plus utilisé supporter par une très grande communauté de volontaires.

Pour cela on doit préparer une machine d'attaque « attacker », on met en exécution le système Kali Linux installer virtuellement sure notre machine physique « host » exploité par Microsoft Windows 10, la figure suivante illustre l'environnement de travail.

Figure 2.9 : Environnement de scan.

Apres avoir préparer l'environnement on Installe ZAP sur la machine Kali par la commande **sudo apt install zaproxy,** à la fin on peut vérifier l'installation par la commande : **zaproxy -h.**

Une fois ZAP est lancé une petite fenêtre de dialogue s'affiche pour demander comment vous voulez choisissez de persister la session.

On prend pour exemple session nommé **test**, la figure suivante montre l'interface fraiche de l'outil ZAP.

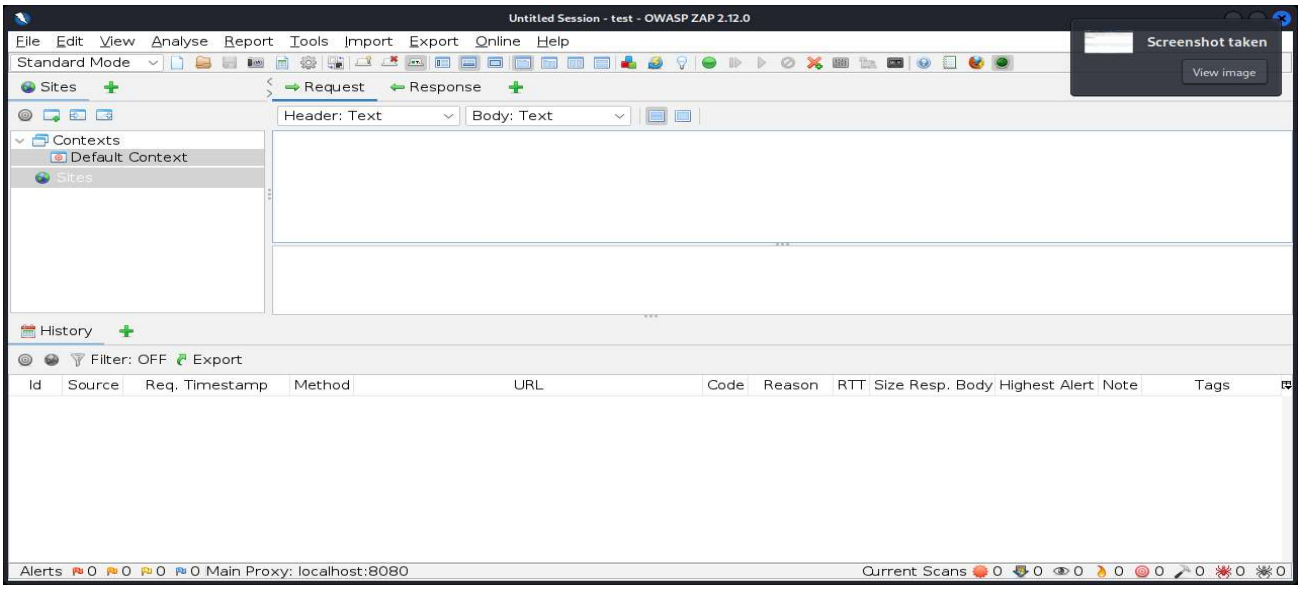

Figure 2.10 : Interface OWASP ZAP.

Au premier on doit ajouter quelque spécification relative à notre cas d'étude, comme la plateforme web et une API ont vérifient dont le manager des modules complémentaire du ZAP si : le module « OpenAPI Support » est installer et bien mise à jour, pour la possibilité de trouver les définitions de l'API.

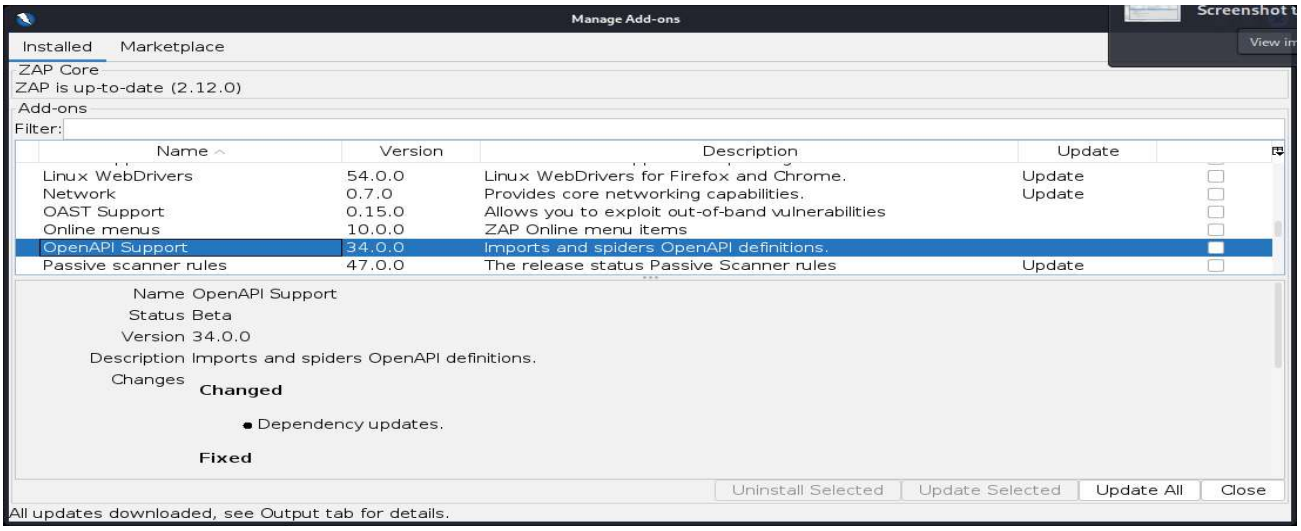

Figure 2.11 : *Manage Add-ons.*

Aussi le « Replacer » ce module donne la possibilité de remplacer des chaines de caractères dans les requêtes et les réponses. Dans notre cas c'est fondamental pour ajouter dans tout requête l'entête jeton d'autorisation d'accès à l'API « Authorization Token ».

Maintenant on charge les URIs, un fichier fournit par l'équipe de développement au format JSON qu'on a converti au format swagger.txt format supporter par ZAP.

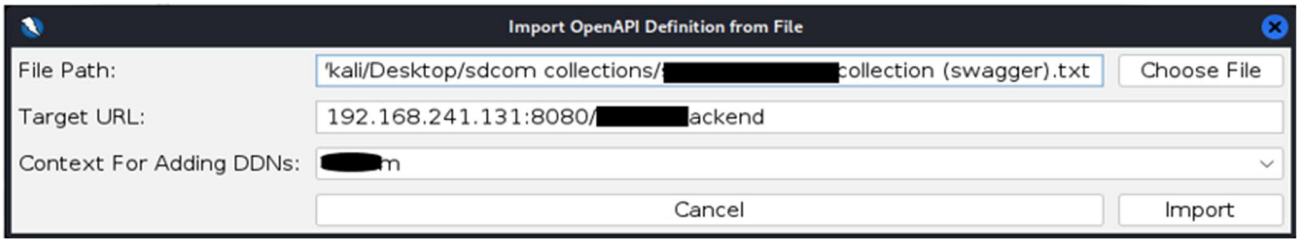

Figure 2.12 : *Import OpenAPI.*

A la fin du chargement des définitions de l'api une fenêtre de dialogue s'affiche indiquent le bon chargement des URIs, et on peut vérifier en déroulant le menu site à gauche de l'interface.

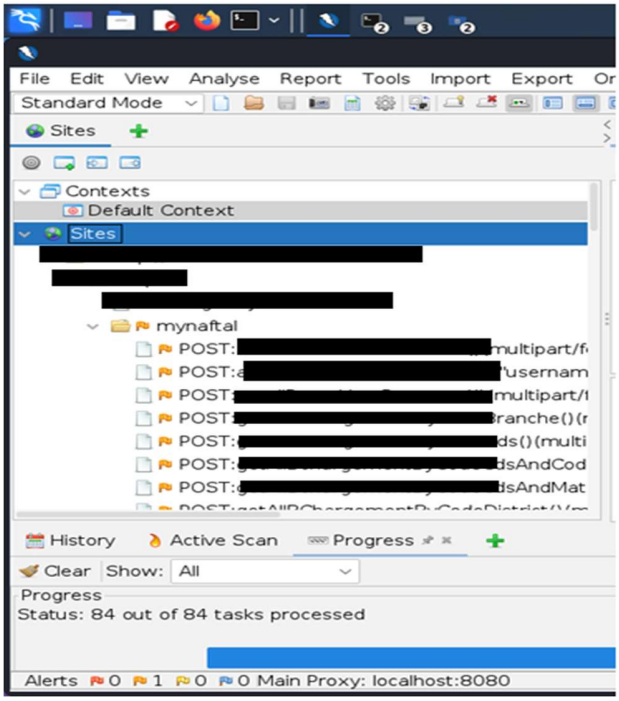

Figure 2.13 : *Chargement des URIs.*

La dernière étape et de charger la clé d'authentification de l'API « Auhtorization Token »

Dans le menu Tools >Options...>Replacer on choisit add et on remplit les informations nécessaires comme la montre la figure suivante :

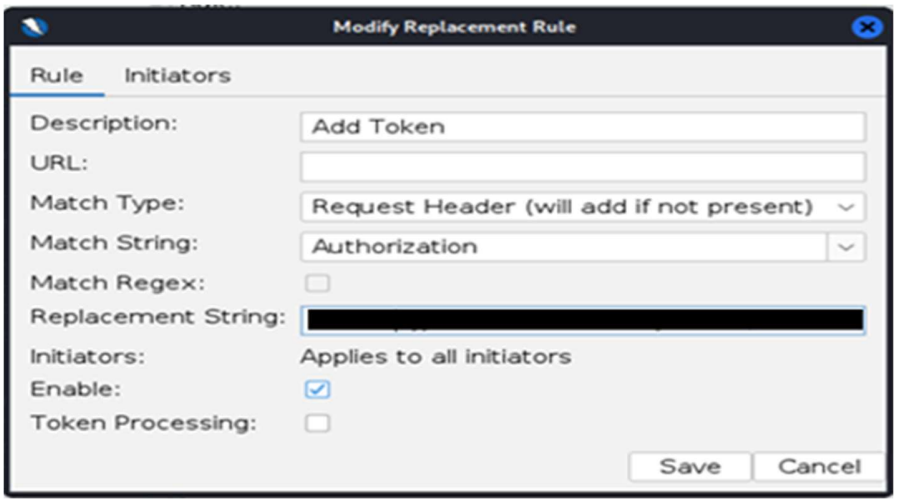

Figure 2.14 : *Ajouter Token.*

Maintenant tout est prêt pour lancer le scan.

On va aller directement tout on bas dans active scan en appuyant sur new scan ; en spécifiant le point de départ et c'est partie, le scan prend plusieurs minutes voire heures.

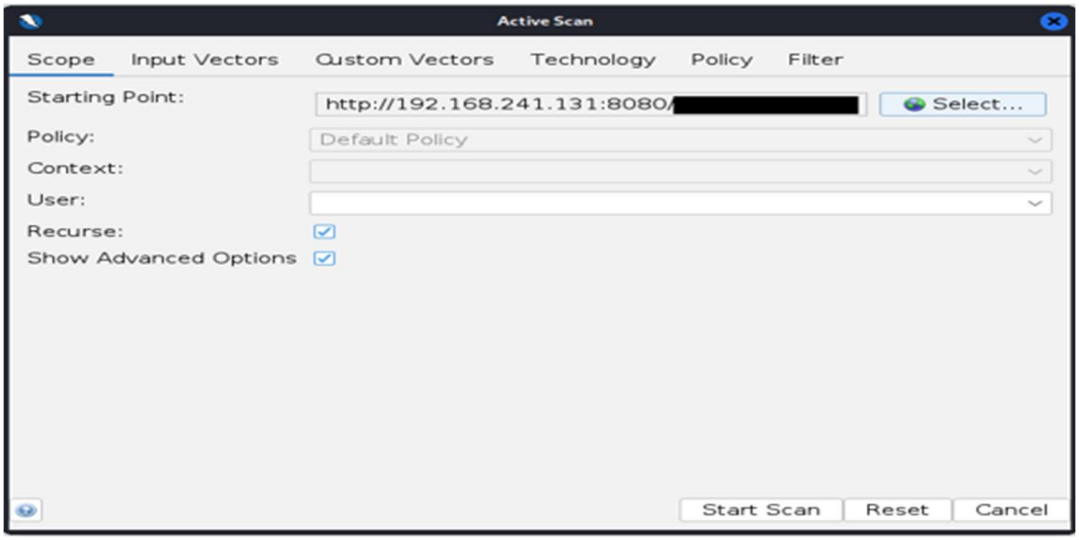

Figure 2.15 : *Active scan.*

Une fois le scan est lancé on peut voir la progression, pausée ou stopper le scan.

A la fin de scan on ouvre l'onglet des Alertes pour voir le résultat de scan, on peut même consulter pour chaque instance d'alerte la requête émis et la repense reçus.

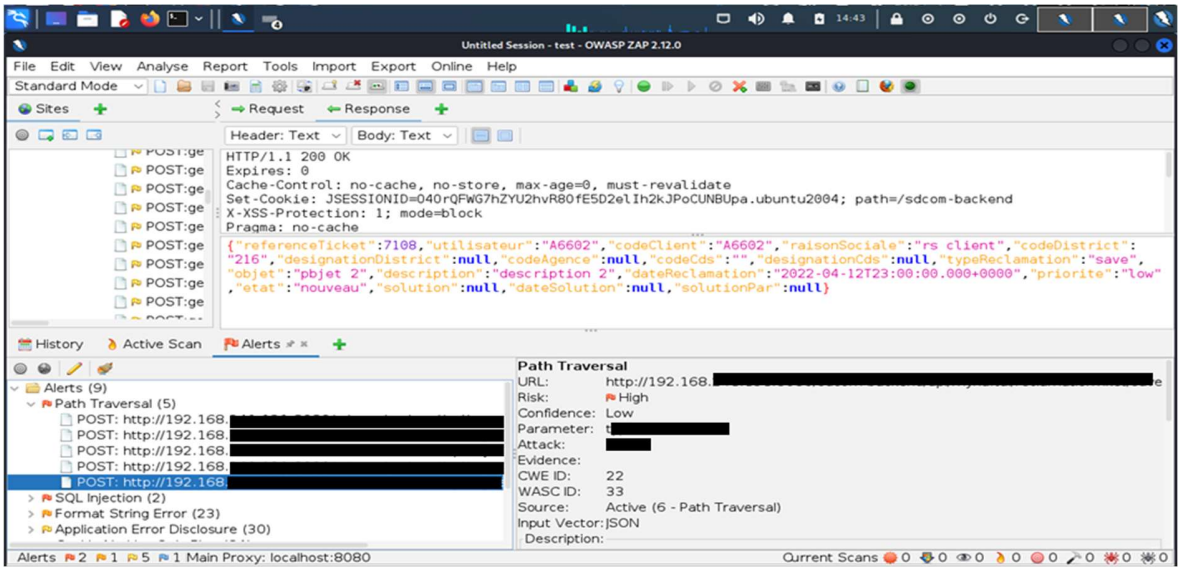

Figure 2.16 : *Resultat.*

La conclusion du scan est bien l'élaboration du rapport du scan, pour cela ZAP donne une variété de choix concernant la structure et le format désiré, comme l'HTML, XML, JSON….

Nous nous intéressons au format JSON

2.2.3 Rapport d'audit

- **Introduction**

Dans cette section nous allons présenter le rapport d'audit que nous avons fourni à la l'équipe de sécurité de la structure d'accueil Naftal pour qu'ils puissent passer à l'étape de remédiation et corriger les vulnérabilités trouvées :

Le présent rapport concerne l'analyse des vulnérabilités effectuée sur le système MyApp de l'entreprise Naftal.

Le scan d'audit a été réalisé le 06 Avril 2023, il a pour but d'évaluer l'état et le statut de sécurité actuelle de l'application ainsi d'identifier les éventuelles failles de sécurité présentes dans le système.

La méthode utilisée pour le scan consiste en une combinaison de scans automatiques. L'outil OWASP ZAP a été configuré et paramétré pour rechercher les vulnérabilités fréquentes telles que les injections SQL, les failles XSS, les problèmes d'authentification, etc. De plus, des tests de configuration ont été effectués pour vérifier la robustesse des paramètres de sécurité du système. Le scan a couvert les composants suivants :

- Application web : MyApp Web Application (version 1.0).
- Serveur de base de données : Microsoft SQL Server 2019.

Il est important de noter que l'audit effectué est basé sur les données et informations fournies par cette version seulement par conséquent il est possible que quelques vulnérabilités ne soient pas détectées ou identifiées en raison de limitations techniques ou de restrictions d'accès. Le scan de vulnérabilités a été effectué dans un environnement de test isolé et n'a eu aucun impact sur le système de production en cours d'exécution.

- **Résumé Exécutif**

Le présent rapport englobe les résultats du scan de vulnérabilités effectué sur le système MyApp de l'entreprise Naftal. L'objectif principal du scan était d'identifier les vulnérabilités potentielles et de mesurer le niveau de risque global du système. Les résultats du scan ont révélé plusieurs vulnérabilités critiques et un niveau élevé de risque associé. Il est recommandé de prendre des mesures de sécurité pour remédier à ces vulnérabilités afin de renforcer la sécurité du système et réduire le taux de risque et voici les principales conclusions :

- Un total de 13 vulnérabilités dont 4 ou le risque est considéré élevé qui représente plus de 30 % de l'intégralité du scan, 2 dont le risque est estimé moyen donc elles représentent un pourcentage de d'environ 15 %, 6 ou le risque est faible avec un pourcentage de 46 %.
- Les vulnérabilités critiques incluent path traversal, SQL Injection, SQL Injection - Oracle - Time Based, SQL Injection - SQLite.
- Les vulnérabilités moyennes comprennent Content Security Policy (CSP) Header Not Set, Format String Error.
- Les vulnérabilités mineures sont Application Error Disclosure, Cookie No HttpOnly Flag, Cookie without SameSite Attribute, Cross Site Scripting Weakness (Persistent in JSON Response, Information Disclosure - Debug Error Messages, X-Content-Type-Options Header Missing.

Recommandations

- Il est recommandé de remédier aux vulnérabilités critiques dans les plus brefs délais, car elles représentent un risque élevé pour la sécurité du système. Cela peut comprendre la mise à jour des composants logiciels pour bénéficier des corrections de sécurité qui sont apportées avec les versions récentes, la correction des configurations incorrectes et la mise en place de dispositifs de protection adéquats.
- Les vulnérabilités moyennes doivent également être traitées dans les meilleurs délais pour réduire le risque global. Cela peut impliquer la mise à jour des bibliothèques utilisées, la configuration sécurisée des composants du système et la mise en œuvre de bonnes pratiques de sécurité.
- Malgré que les vulnérabilités à faible risque présentent un risque relativement mineur, il est recommandé de les examiner et de les corriger pour maintenir un niveau de sécurité maximal.

Il est primordial de prendre ces mesures de remédiation pour réduire les risques associés aux vulnérabilités trouvées. L'amélioration de la sécurité du système MyApp garantira la confidentialité, l'intégrité et la disponibilité des données et des ressources.

➢ Méthodologie

La portée du scan de vulnérabilités comprenait l'évaluation du système MyApp ainsi que des applications associées. Les principaux éléments inclus dans la portée étaient l'application web et la base de données.

Pour réaliser le scan, l'outil de sécurité ZAP a été utilisé, la méthodologie suivie implique la configuration des paramètres du scan, la définition des adresses IP à analyser, la configuration de l'API header, l'ajout de la collection des API et l'ajout du authentication token. Le résultat du scan a été enregistré et analysé pour définir les vulnérabilités éventuelles.

Les trois types de scans lances sont :

- **Scan passive :** Il analyse et observe le trafic réseau entre le client et le serveur de l'application sans exécuter d'actions actives ou de requêtes directes.
- **Scan active :** Il implique une interaction directe avec l'application cible.

➢ **Vulnérabilités identifiées**

Ci-dessous, vous trouverez une liste des vulnérabilités principales et importantes identifiées lors du scan de vulnérabilités :

- **Path Traversal**

Définition : est une vulnérabilité qui permet à un attaquant d'accéder à des fichiers et des répertoires hors de la portée prévue d'une application. Cela a lieu lorsque les entrées fournies par l'utilisateur ne sont pas correctement validées ou filtrées.

Impact : elle permet à un attaquant de lire des fichiers confidentiels, de modifier des données, d'exécuter du code malveillant et d'obtenir un accès non autorisé au système.

Recommandation : pour remédier et éviter cette vulnérabilité il faut prendre en charge quelque mesure tels que la vérification et la validation des entrées, établir une liste blanche des chemins, Configurer les permissions du système de fichiers.

Voici le résultat rendu par OWASP ZAP pour la vulnérabilité Path Traversal

Figure 2.17 : Path traversal vulnérabilité

- **SQL Injection**

Définition : Cette vulnérabilité permet à un attaquant d'exécuter des requêtes SQL malveillantes et d'endommager la base de données.

Impact : Un attaquant pourrait avoir un accès non autorisé aux données sensibles de l'application.

Recommandation : Mettre en œuvre une vérification et un filtrage des entrées utilisateur.

Voici le résultat rendu par OWASP ZAP pour la vulnérabilité SQL Injection

Figure 2.18 : SQL Injection Vulnérabilité

- **Recommandation de remédiation**

Cette partie fournit des recommandations pour remédier aux vulnérabilités identifiées lors du scan de vulnérabilités. Les recommandations ont pour but d'atténuer les risques de sécurité.

- **Path Traversal**

Pour prévenir les attaques de path traversal, il est essentiel de mettre en œuvre des techniques de validation des entrées et d'encodage des sorties :

1. Filtrez toujours les entrées fournies par l'utilisateur avant de les utiliser pour construire des chemins de fichier ou accéder à des ressources externes.
2. Utilisez une technique qui repose sur le principe de liste blanche pour définir des modèles d'entrées valides et refuser toute entrée qui ne correspond pas à ces modèles.
3. Utilisez des techniques d'encapsulation dédiées pour isoler les entrées fournies par l'utilisateur des opérations sur le système de fichiers.
4. Conservez une liste blanche des chemins de fichier ou répertoires permis que l'application puisse accéder.

- **SQL Injection**

1. Mettre en place une validation des entrées utilisateur pour empêcher les injections SQL.

2. L'utilisation des requêtes paramétrées ou des ORM (Object-Relational Mapping).
3. Effectuez une revue de code pour identifier et corriger les vulnérabilités liées aux injections SQL.

Il est essentiel et primordial de prendre en considération les recommandations lister ainsi que faire appel à des professionnels en cas de difficulté à éliminer les vulnérabilités.

- **Conclusion et plan d'action**

Après avoir évalué les résultats du scan de vulnérabilités, les vulnérabilités importantes ont été identifiées dans le système ou l'application. Ces vulnérabilités peuvent potentiellement être exploitées par des attaquants pour des buts malveillants.

Pour assurer la sécurité du système, il est fortement suggéré de mettre en œuvre les mesures suivantes :

1. Classez les vulnérabilités en fonction de leur gravité et de leur probabilité d'exploitation afin de classer les actions à corriger par ordre de priorité.
2. Commencez par donner la priorité de remédiation aux vulnérabilités les plus critiques qui présentent un risque élevé et une probabilité d'exploitation élevée.
3. Mettez en place les recommandations de remédiation spécifiques pour chaque vulnérabilité.
4. Après la correction des vulnérabilités identifiées, mettez en place des mesures préventives telles que la sensibilisation à la sécurité, la formation des développeurs, et l'adoption de bonnes pratiques de développement sécurisé.
5. Réalisez des audits réguliers de sécurité pour identifier de nouvelles vulnérabilités et garantir que le système reste sécurisé au fil du temps et bien mis à jour.
6. Respectez les meilleures pratiques de sécurité en matière de configuration des serveurs, de validation des entrées utilisateur, de gestion des mots de passe, de contrôle d'accès.

2.3 Conclusion

Ce chapitre a exploré en détail l'audit de l'application web 'MyApp', en mettant l'accent sur les outils utilisés, la préparation de l'environnement de travail, le scan automatique et la génération du rapport d'audit. Nous avons examiné l'importance de l'audit d'une application web pour garantir sa sécurité et protéger les données sensibles des utilisateurs.

Chapitre 3 : Etude conceptuelle

Introduction

Dans ce chapitre nous allons présenter et simplifier au mieux la problématique grâce à une représentation détaillée des étapes conceptuelles.

Toute au début nous allons faire l'analyse des besoins qui réside à évaluer la situation afin de tenir compte des contraintes à la fin nous allons mettre en œuvre la conception grâce aux différents diagrammes.

3.1 Présentation globale de l'application

L'un des buts de notre projet est de réaliser une application qui permet d'automatiser quelques tâches tels que le lancement d'un scan sur une application web cible en utilisant l'outil de scan de vulnérabilités OWASP ZAP tout ça d'une façon simple et fluide pour l'utilisateur.

Notre application sera connue sous le nom de « Audit+ » pour « Audit Plus ».

Audit+ est une application qui permet aux membres de l'équipe de sécurité de lancer des scans sur une application web 'XYZ' et d'afficher les résultats d'une façon lisible, claire et d'estimer l'état de sécurité actuelle de l'application grâce aux statistiques fournies.

L'objet de réaliser cette application c'est de faciliter le travail de l'équipe de sécurité et gagner en matière de temps car elle permet de passer plusieurs étapes qui peuvent nécessiter beaucoup de temps, de configuration et de savoir-faire.

La figure illustre un schéma simple et global sur le fonctionnement de l'application Audit+.

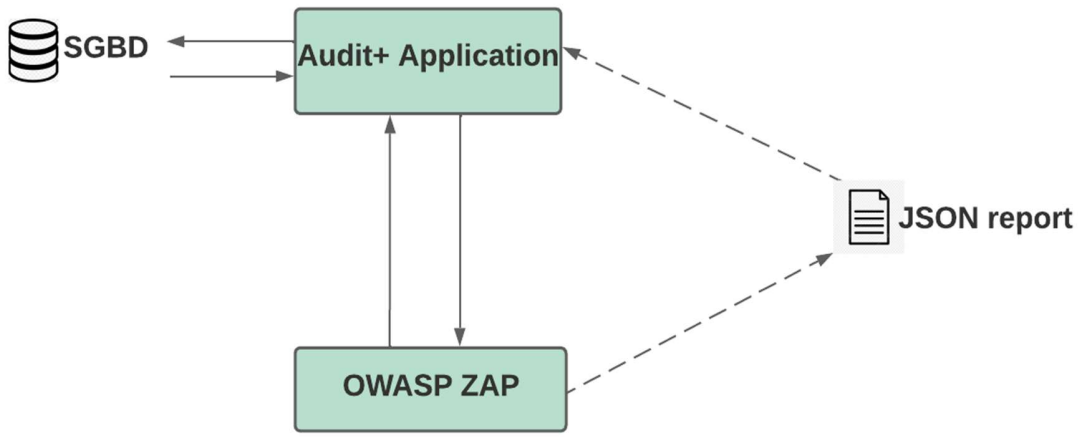

Figure 3.19 : *Schéma global de l'application.*

Explication :

- L'auditeur lance un scan depuis l'application Audit +.
- L'API de OWASP ZAP est appelée et lance le scan.
- Apres la fin du scan un rapport sous format JSON est généré.
- L'application Audit+ récupère le rapport, effectue un traitement bien précis.
- Stock chaque valeur dans sa table approprie dans la base de données.

3.2 Etude conceptuelle de notre application

3.2.1 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de décrire la relation entre les acteurs et les fonctionnalités du système, c'est-à-dire, de déterminer les possibilités d'interaction entre le système et les acteurs extérieurs.

Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur [15]**.**

Le modèle doit donc contenir deux identifications :

- **Un acteur :** est une entité externe au système logiciel que vous êtes en train de modéliser. Il représente un rôle joué par une personne ou une chose qui interagit avec le système.
- **Un cas d'utilisation :** Il représente une fonctionnalité ou une action que le système doit effectuer pour répondre à un besoin de l'utilisateur. Il décrit une interaction entre un acteur et le système, en mettant en œuvre les objectifs et les résultats attendus.

Dans le cadre de notre système nous avons identifié les acteurs et cas d'utilisation suivants :

- **Identification des acteurs :** Nous avons identifié un seul acteur dans le système qui est "l'auditeur". Les auditeurs : C'est les personnes qui peuvent accéder au système et exécuter les tâches tels que le lancement d'un scan, consulter les résultats d'un scan.
- **Identification des cas d'utilisations :** Les principaux cas d'utilisation sont les opérations qui concernent l'audit de sécurité informatique :
    - La création d'une tâche de scan.
    - La personnalisation d'une tâche de scan.
    - La consultation des résultats du scan.
    - La planification de lancement du scan a une date et heure bien précise.

Le diagramme suivant donne une vue générale sur les relations entre les acteurs et les cas d'utilisations essentiels du système,

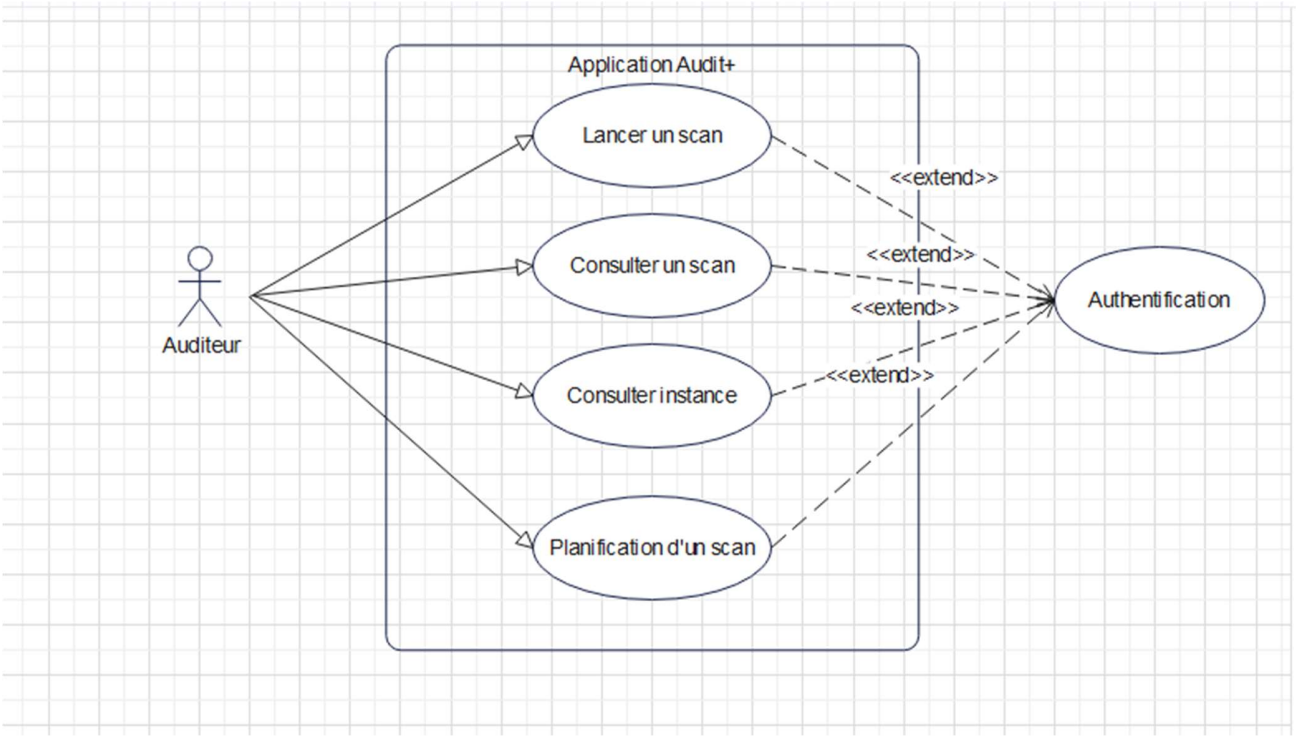

Figure 3.20 : *Diagramme de cas d'utilisation.*

Tableau 3.2 *: Description du diagramme de cas d'utilisation*

Cas d'utilisation	Acteur	Description
Lancer Un scan.	Auditeur	L'auditeur lance un scan OWASP ZAP par le biais de l'application.
Consulter résultat d'une alerte.		L'auditeur peut consulter et examiner le résultat d'une alerte spécifique.
Consulter résultat d'une instance.		L'auditeur peut consulter et examiner le résultat d'une instance spécifique.
Planifier le lancement d'un scan		L'auditeur peut planifier le lancement du scan a une date et heure bien précise

3.2.2 Digramme de séquence

Le diagramme de séquence est un diagramme UML qui fait partie des diagrammes comportementaux (dynamiques). Son objectif est de représenter les interactions entre les objets

et les acteurs, ou entre les objets uniquement, en indiquant la chronologie des échanges. Cette représentation peut se réaliser en considérant les différents scenarios associés [16].

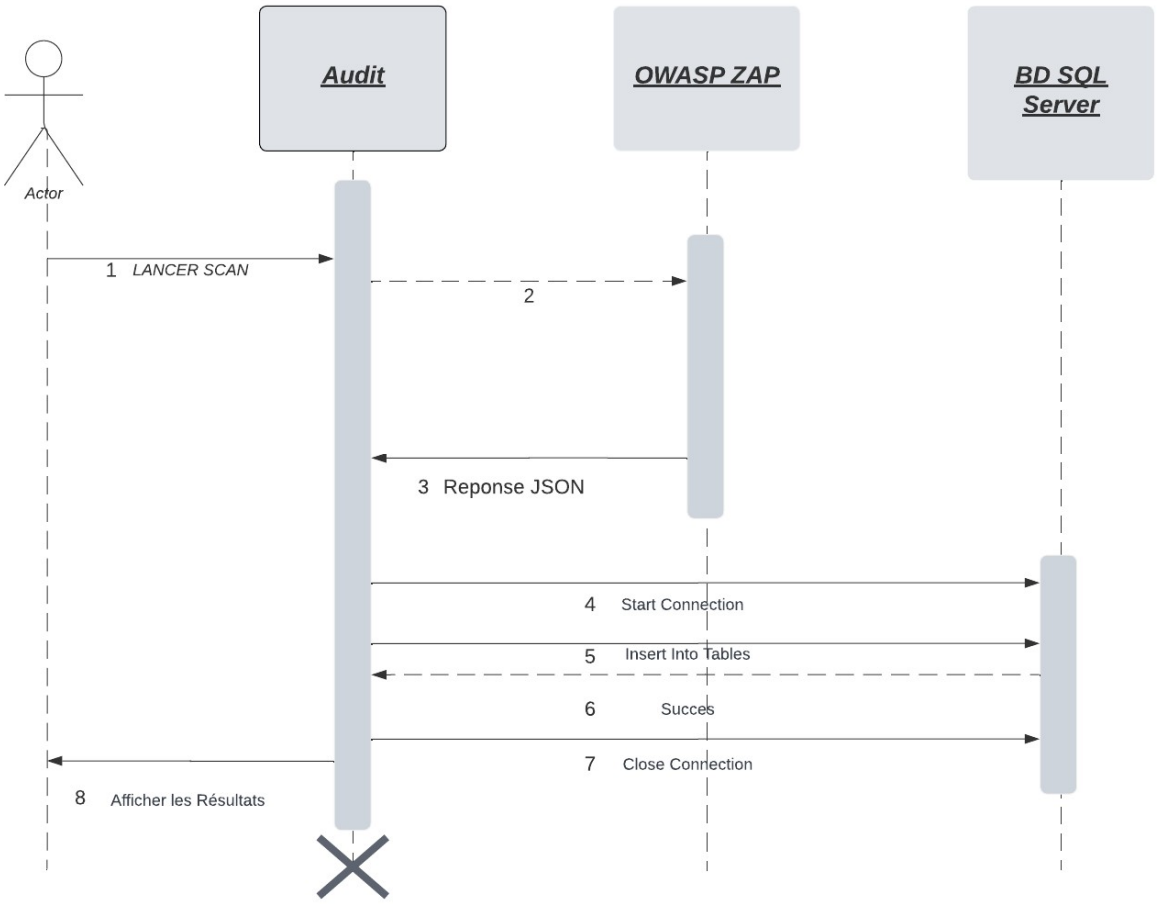

Figure 3.21 : *Diagramme de séquence cas d'utilisation lancer un scan.*

Explication :

1 L'auditeur commence par lancer un scan depuis l'application Audit+.
2 L'application fait appel à l'API de OWASP ZAP et lance le scan.
3 Apres la fin du scan un rapport JSON est généré et envoyer à l'application Audti+.
4 L'application initie une connexion avec la base de données.
5 L'application insert chaque valeur dans sa table appropriée.
6 Un message de succès est retourne à l'application.
7 La connexion est fermée.
8 L'application affiche les résultats d'une façon bien lisible et organisé.

3.2.3 Digramme d'état-transition :

Un diagramme d'état-transition, est une représentation graphique qui modélise le comportement d'un système en décrivant les états possibles du système, les transitions entre ces états et les événements qui déclenchent ces transitions.

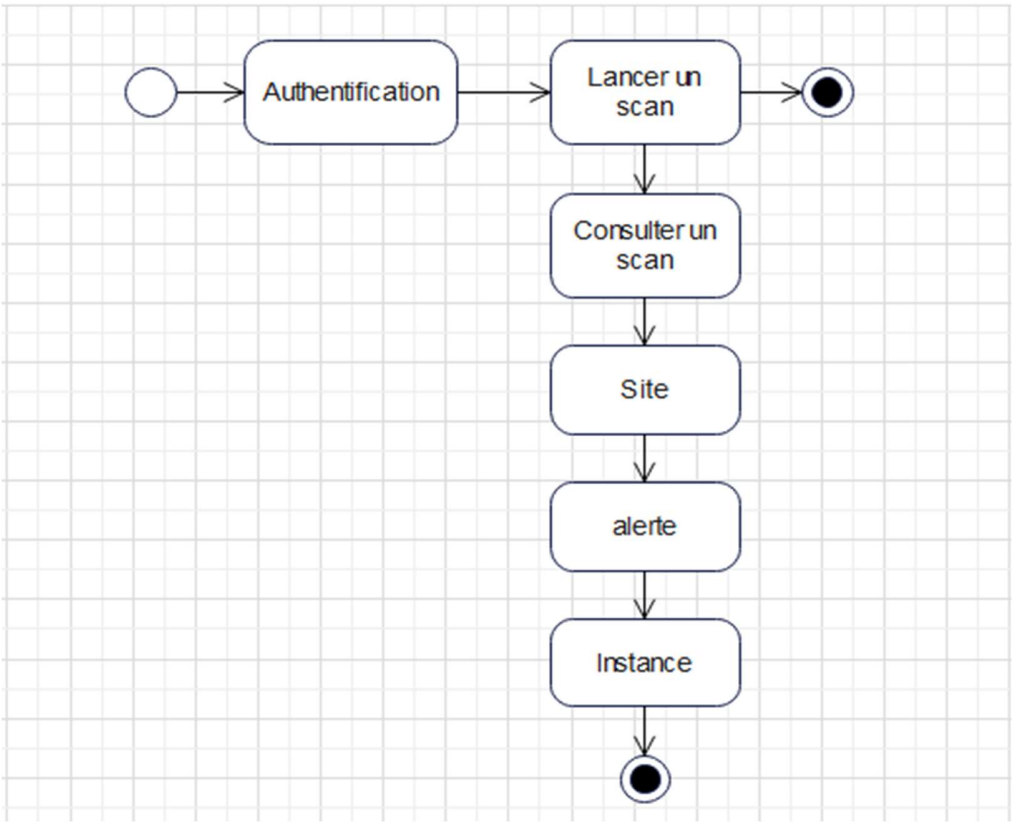

Figure 3.22 : Diagramme d'état transition

3.3 Conclusion

Dans ce chapitre Nous avons donné une vue globale sur l'application Audit+ en plus de présenter la conception de cette dernière.

Dans le prochain chapitre, nous allons aborder l'implémentation et la réalisation d'une application d'automatisation du processus d'audit. Nous explorerons les meilleures pratiques pour automatiser certaines étapes de l'audit et faciliter la gestion des résultats.

Chapitre 4 : Implémentation et réalisation

Introduction

Dans ce chapitre, nous aborderons la mise en place de l'environnement de développement nécessaire à la réalisation de notre application, ainsi que l'implémentation concrète de celle-ci. Nous passerons en revue les étapes du test réalisé sur l'application Audit+.

4.1 Environnement de développement

Dans le développement de notre application web nous avons utilisé une machine Windows qui contient le code ainsi que la base de données en plus d'une VM Kali Linux qui comporte le module qui va effectuer le scan 'OWASP ZAP'.

Nous avons devisé notre application en deux parties distinctes Front end et Back end chaque une est mise en œuvre grâce à plusieurs langages et technologies comme le montre Figure 4.12 et Tableau 4.3 sachant que les deux travails en coordination pour assurer le bon fonctionnement de l'application

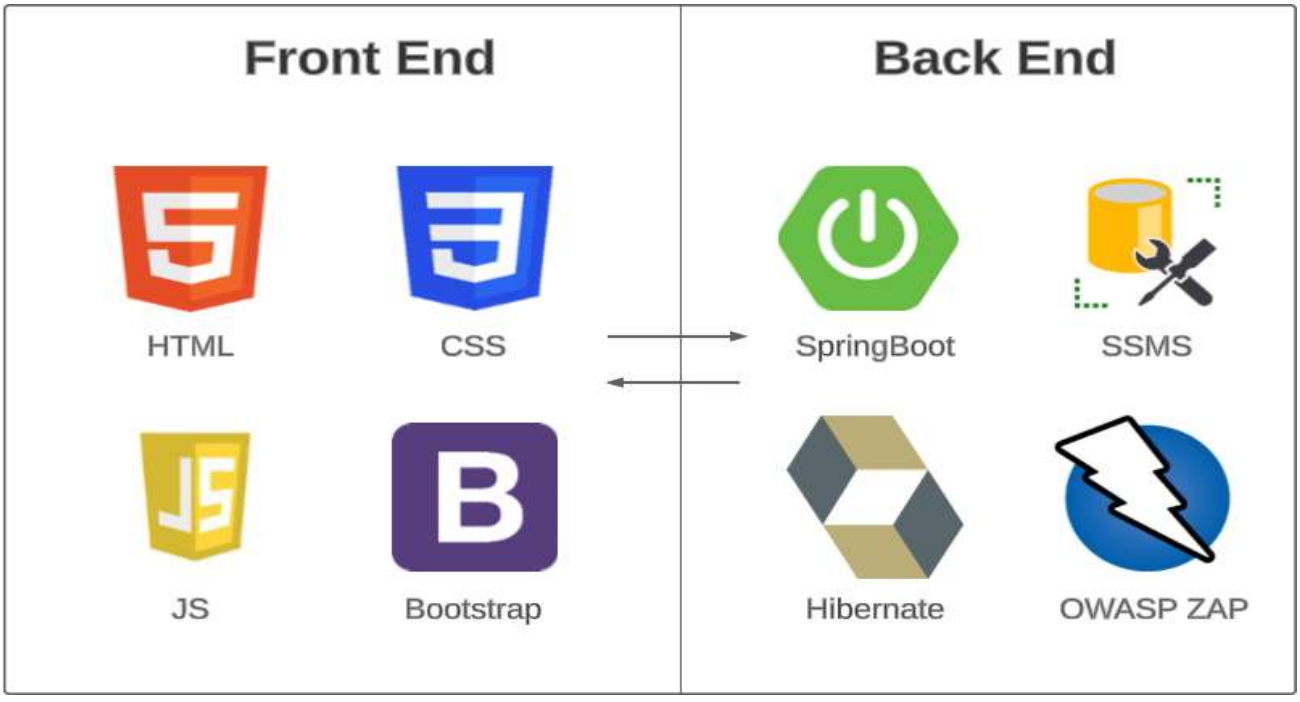

Figure 4.23 : *Environnement de développement.*

Tableau 4.3 : Langage de programmation et technologie utilisé

Langage	Description
HTML	(HyperText Markup Language) est un langage de balisage utilisé pour créer et structurer le contenu des pages web.
CSS	(Cascading Style Sheets) Langage de style permettant de définir l'apparence et la mise en forme des éléments HTML [17].
JS	(JavaScript) est un langage de programmation qui permet d'ajouter de l'interactivité et de la dynamique aux pages web.
Bootstrap	Collection d'outils utiles à la création du design, graphisme, animation et interactions avec la page dans le navigateur [18].
Spring Boot	Spring Boot est un Framework Java qui simplifie le développement d'applications web en fournissant des configurations par défaut intelligentes et en intégrant de nombreux composants essentiels.
SSMS	(SQL Server Management Studio) est un outil de gestion et d'administration de bases de données SQL Server.
Hibernate	Hibernate est un Framework de mapping objet-relationnel pour Java. Il facilite l'interaction entre une application Java et une base de données relationnelle en permettant de stocker, récupérer, mettre à jour et supprimer des objets Java directement dans la base de données, sans avoir à écrire de requêtes SQL.
OWASP ZAP	(Zed Attack Proxy) est un outil open-source de test de sécurité des applications web. Il est utilisé pour détecter les vulnérabilités et les failles de sécurité dans les applications web.

4.2 Implémentation de l'application

Notre application Audit+ a pour objectif :

- Authentification sécurisée.
- Lancer un scan depuis l'interface de l'application il suffit juste de préciser les adresses IP de la machine cible et la machine qui va effectuer l'attaque cela permet de faciliter la tâche dans le cas d'un réseau qui est compose de plusieurs machines.
- Choisir entre lancer un scan actif, passive.

- Consulter les résultat des scans d'une façon bien lisible, compréhensible et structuré en comparent avec le rapport JSON résultant de OWASP ZAP qui peut contenir parfois un nombre énorme de ligne ce qui rend le processus d'analyse du rapport compliqué et long.
- Programmer le lancement d'un scan d'une façon automatique a une heure et une datte bien précise. Cette fonctionnalité vient résoudre un problème très important qui est dû à la nature du scan active qui consiste à l'envoi de requêtes ou l'exécution d'actions qui simulent des techniques d'attaque réelles afin d'identifier les vulnérabilités d'un système cible. Bien que l'objectif soit de trouver les failles de sécurité, la nature agressive du scan actif peut parfois avoir des conséquences imprévues, Parmi elle on peut citer :
    1. Déni de service (DoS).
    2. Surcharge du réseau ou de l'infrastructure.
    3. Dysfonctionnement ou incompatibilité des techniques de scan.

  Pour ces raisons là les spécialistes de sécurité suggèrent d'effectuer ces scans actifs pendant les périodes de faible trafic.

4.3 Tests et résultat

Dans cette section, nous aborderons les tests effectués ainsi que les résultats obtenus.

Dans le cadre de notre projet, nous avons réalisé des tests au sein de l'entreprise Naftal SPA. Cet environnement de test a permis de simuler des situations réelles et de mettre à l'épreuve notre solution dans des conditions proches de la réalité.

En respectant les lois et les réglementations en matière de sécurité des données, de confidentialité et de protection des informations sensibles, nous nous sommes assurés de mener les tests dans le respect des politiques et des obligations légales en vigueur. Et c'est pourquoi, nous allons masquer quelque information sensible tels que les URI et api dans les figures de cette section où nous allons simuler les techniques choisies.

4.3.1 Authentification

L'accès à l'application Audit+ est soumis à une authentification préalable pour garantir la sécurité et la confidentialité des données. Avant de pouvoir accéder à l'application, l'auditeur doit fournir son adresse e-mail ainsi que son mot de passe. Ces informations sont ensuite comparées aux données stockées dans la base de données, Si les données sont identiques l'auditeur est autorisé à accéder à

l'application. Cette mesure de sécurité garantit que seules les personnes autorisées et légitimes peuvent accéder à l'application et effectuer des activités d'audit. Cela renforce la confiance dans l'intégrité et la confidentialité des données d'audit.

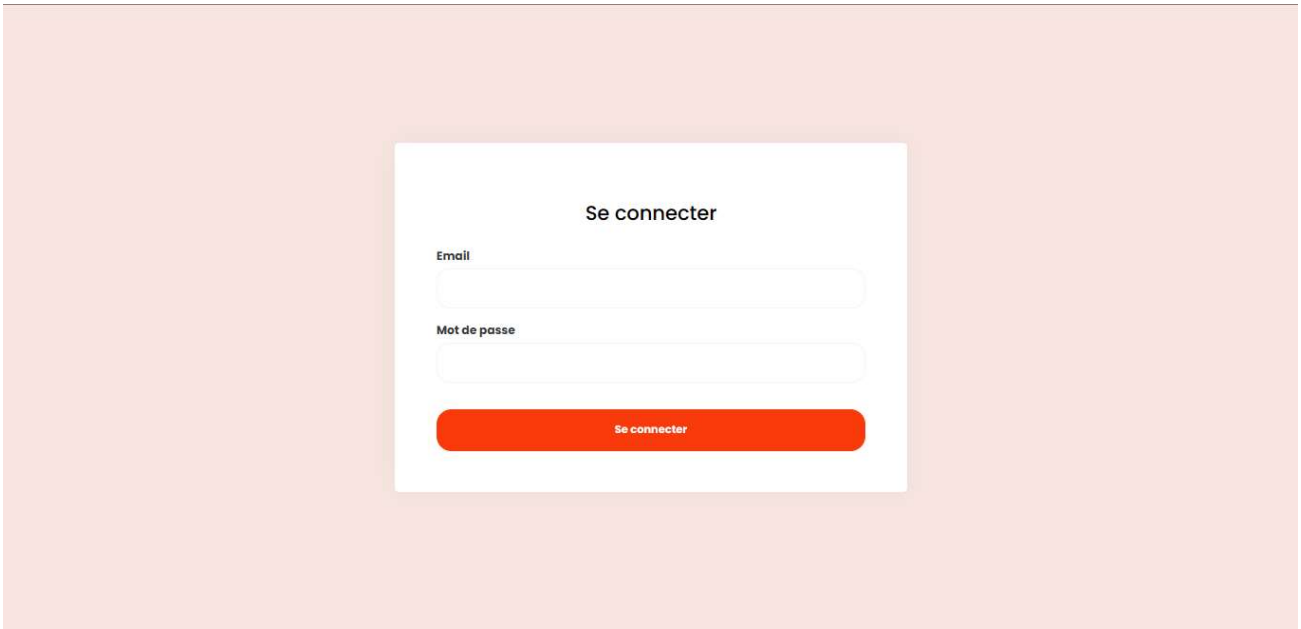

Figure 4.24 : page de Login.

4.3.2 Lancer un scan

Une fois que l'auditeur s'est authentifié avec succès, il est dirigé vers la page "Lancer un scan" de l'application. Sur cette page, il a la possibilité de lancer un scan automatique de l'application cible directement depuis l'interface de l'application Audit+. Cette fonctionnalité offre à l'auditeur une grande commodité, lui permettant d'initier le processus de scan sans avoir à utiliser à des outils externes ou à des procédures complexes.

- **Etape 1**

La première étape du lancement automatique d'un scan consiste à entrer les coordonnées de la machine cible. L'auditeur doit préciser l'adresse IP de la machine cible, le numéro de port sur lequel l'application est hébergée et l'URI qui représente le chemin d'accès à l'application Ces informations permettent à l'application Audit+ de se connecter à la machine cible et de cibler spécifiquement l'application à auditer.

De plus, si la machine cible nécessite un token d'authentification pour accéder à l'application, l'auditeur doit également fournir ce token lors du lancement du scan automatique.

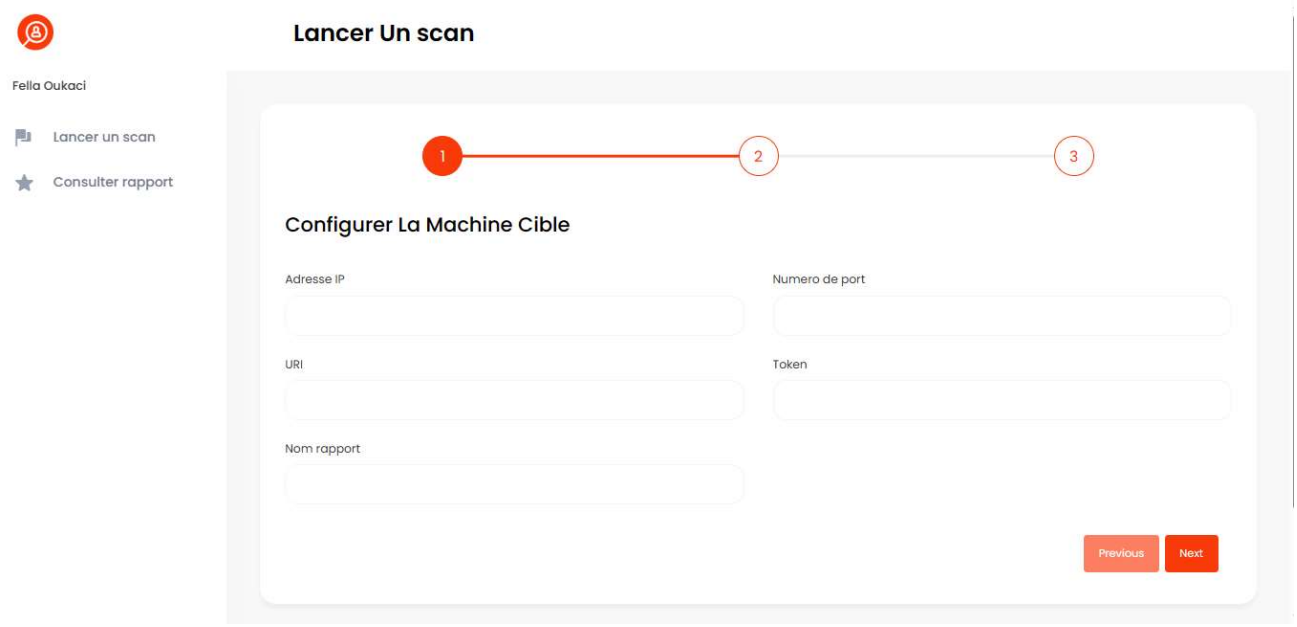

Figure 4.25: *Lancer un scan -Etape 1-*

La deuxième étape du lancement automatique d'un scan consiste à fournir les informations de la machine qui va effectuer l'attaque, ainsi que l'API Key si elle est requise.

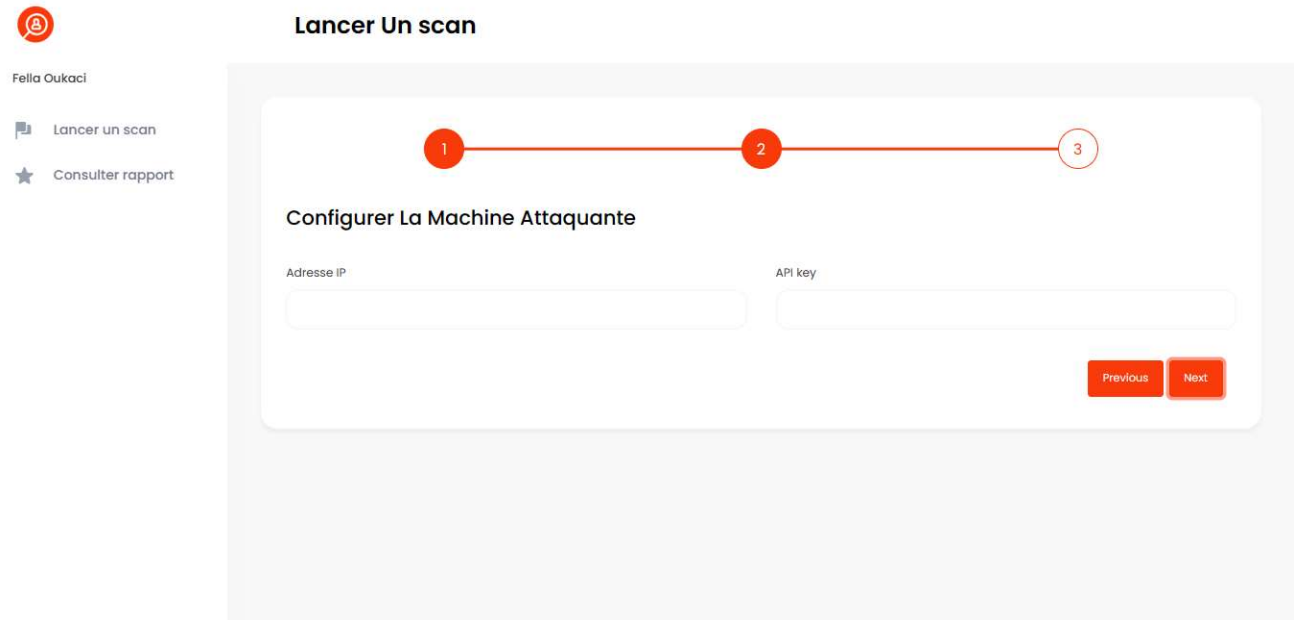

Figure 4. 26: *Lancer un scan -Etape 2-*

- **Etape 3**

Dans la dernière étape, l'auditeur a la possibilité de choisir le type de scan à réaliser, soit un scan actif ou un scan passif. Une fois que l'auditeur a choisi le type de scan souhaité, il peut lancer le scan à partir de l'application d'audit ou bien le programmer a une date et heure bien précise. Cela déclenche le processus d'analyse de l'application cible en utilisant les paramètres spécifiés, tels que l'adresse IP, le numéro de port, l'URI et les informations d'authentification si nécessaire.

Cette dernière étape permet l'auditeur lancer le scan choisi pour évaluer la sécurité de l'application et identifier les éventuelles vulnérabilités. Ces résultats seront ensuite utilisés pour effectuer une analyse de sécurité approfondie et prendre les mesures nécessaires pour renforcer la sécurité de l'application.

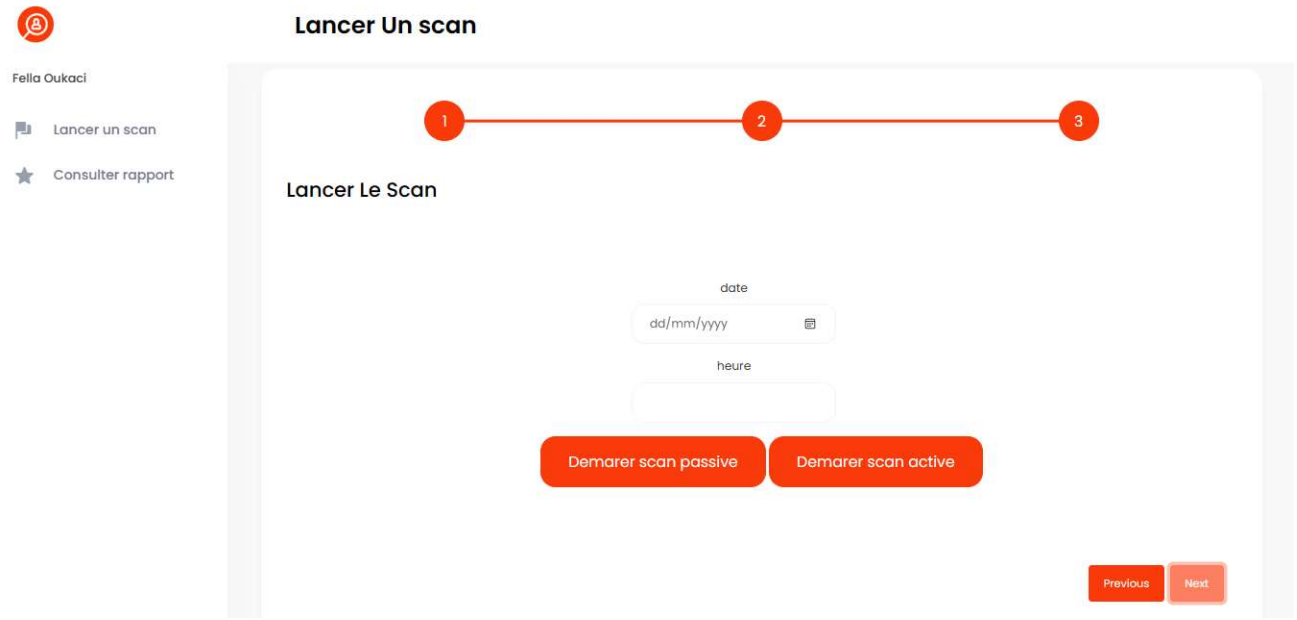

Figure 4. 27: Lancer un scan -Etape 3-

4.3.3 Consulter un scan

Dans l'application Audit+, l'auditeur dispose d'une fonctionnalité essentielle qui lui permet de consulter les résultats des scans précédents de manière lisible et bien organisée. Cela lui permet d'avoir une vision claire des vulnérabilités identifiées, des failles de sécurité potentielles et des mesures recommandées pour renforcer la sécurité de l'application. Les résultats des scans sont généralement présentés sous forme de rapports détaillés, comprenant des informations telles que les vulnérabilités détectées, leur impact potentiel, leur priorité, et les actions recommandées pour les corriger. L'auditeur peut naviguer à travers ces rapports et accéder aux détails spécifiques de chaque vulnérabilité détectée.

L'interface de l'application Audit+ est conçue de manière à faciliter la consultation des résultats. Les informations sont organisées de manière claire et structurée, avec des sections dédiées aux différentes catégories de vulnérabilités.

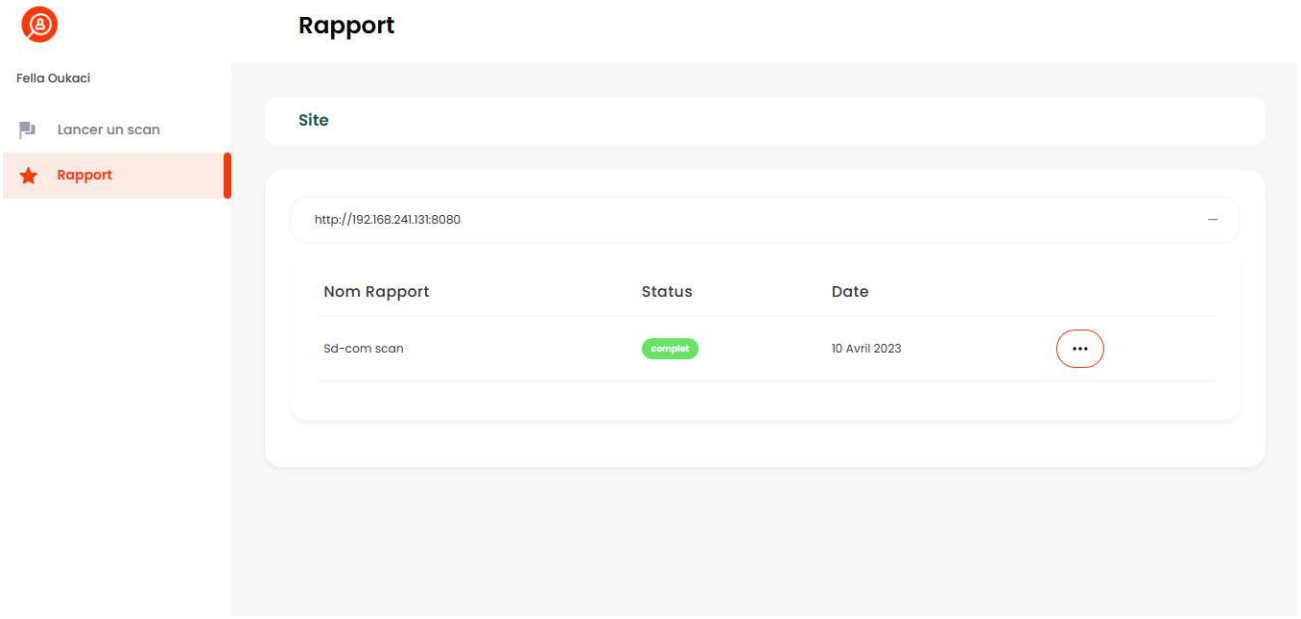

Figure 4. 28 : Consulter résultat de scan.

Si l'auditeur veut consulter les détails du scan 'sd-com' par exemple il appui sur le bouton 'Consulter' et il sera redirigé vers une autre page qui contient toutes les alertes et leurs détails.

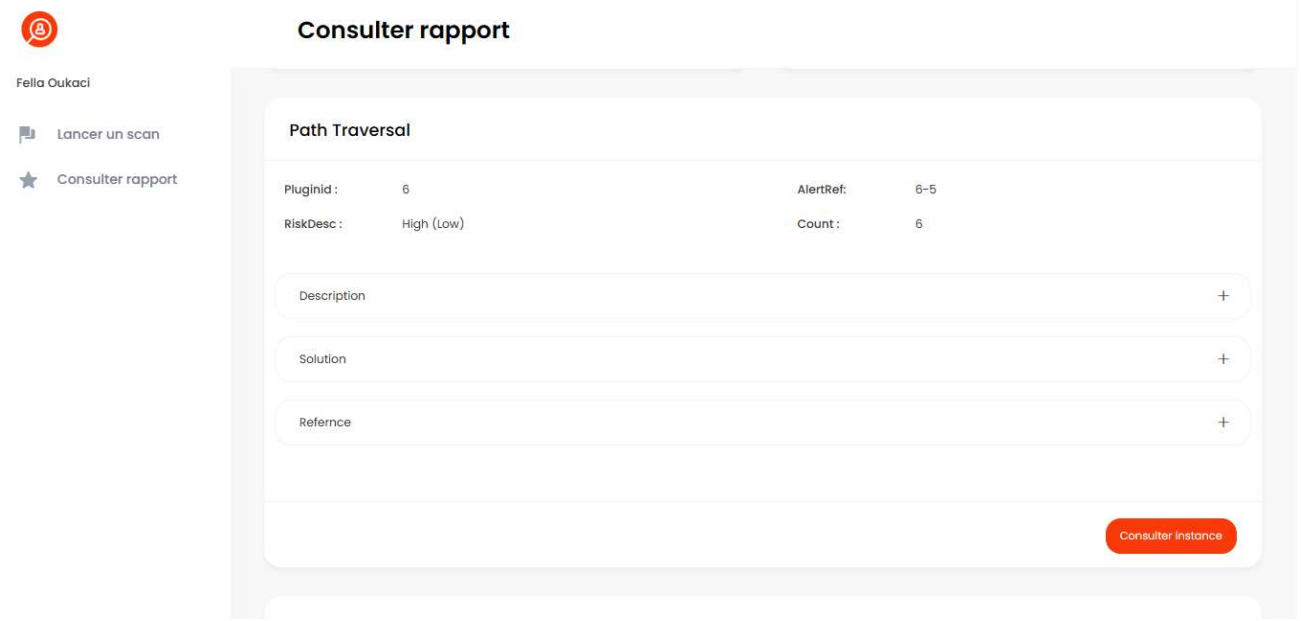

Figure 4. 29 : Consulter un rapport.

Lorsqu'une alerte est identifiée, l'auditeur peut sélectionner cette alerte dans la liste des résultats, ensuite, il peut accéder à une page dédiée qui affiche toutes les instances spécifiques de cette alerte sur cette page, l'auditeur peut explorer les détails de chaque instance.

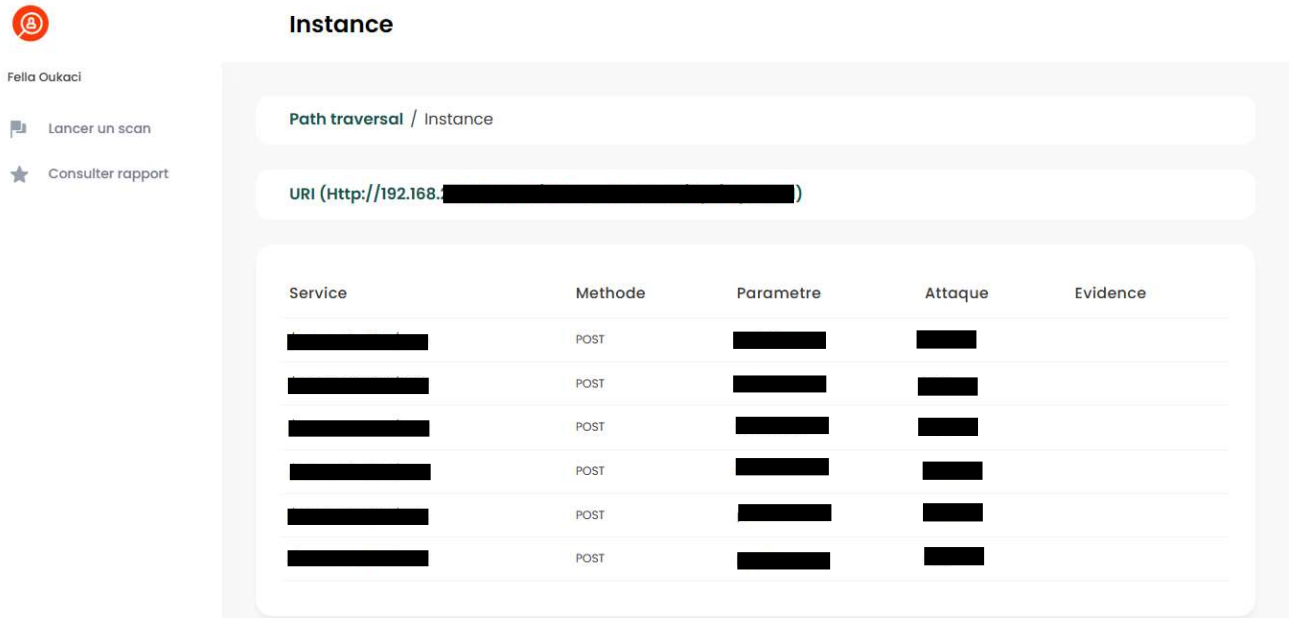

Figure 4. 30 : Consulter Instance.

4.4 Conclusion

En conclusion, ce chapitre a fourni une vue d'ensemble de notre application en mettant l'accent sur ses fonctionnalités, son environnement de développement et de test, ainsi que ses interfaces.

Conclusion générale

En conclusion, ce projet de fin d'études qui consiste à réaliser un audit sur l'application MyApp ensuite l'implémentation de notre application Audit+, qui vise à simplifier et à automatiser ce processus a été mené en suivant une approche méthodique et rigoureuse. Nous avons réalisé une étude bibliographique approfondie pour acquérir les connaissances nécessaires à la conception de notre solution. Dans notre premier chapitre nous avons introduit les concepts clés de la sécurité informatique et se concentre sur la sécurité des applications web ainsi que l'importance de la réalisation d'audits de sécurité régulière. Les outils de scan web, le serveur JBoss et l'architecture trois tiers sont également abordés pour fournir une compréhension approfondie du sujet. Dans le deuxième chapitre nous avons présenté et détaillé notre solution, l'application Audit+, nous avons examiné l'environnement dans lequel l'application sera mise en œuvre et explore les besoins et les fonctionnalités clés de notre solution. Des diagrammes UML sont utilisés pour illustrer l'architecture et le fonctionnement de l'application. Le troisième chapitre se concentre sur la réalisation de l'audit de sécurité d'une application web, nous avons commencé par présenter les outils que nous avons utilisés pour mener à bien notre travail, ensuite, nous avons détaillé les différentes étapes que nous avons suivies pour réaliser notre scan ainsi que le rapport. Dans ce dernier chapitre, nous avons abordé l'implémentation de l'application Audit+ et son fonctionnement. On détaille chaque module de l'application, expliquant comment ils contribuent à automatiser le processus d'audit de sécurité. Le chapitre met en avant les avantages de l'application, tels que sa facilité d'utilisation, la clarté des résultats affichés.

En conclusion, ce mémoire a permis de mettre en évidence l'importance de l'audit de sécurité des applications web et de présenter notre solution, l'application Audit+, qui vise à simplifier et à améliorer ce processus. Notre application offre aux auditeurs un moyen plus efficace de détecter et de remédier aux vulnérabilités de sécurité.

Parmi les exigences que nous avions fixées dès le début du projet, nous sommes parvenus à respecter l'objectif d'extensibilité de la solution proposée. Cependant, il est important de souligner que tout projet peut toujours bénéficier d'améliorations, Des perspectives d'amélioration et d'extension peuvent donc être envisagées afin d'enrichir et améliorer, dans l'avenir, notre solution.

Conclusion générale

Nous proposons :

- Ajouter une documentation afin d'expliquer comment utiliser cette application par L'utilisateur.
- Améliorer les interfaces afin d'offrir aux utilisateurs une meilleure expérience.
- Améliorer le développement sécurisé de l'application et renforcer les lignes de sécurité.
- Intégrer d'autre outil de scan comme Open VAS, Nmap.
- Ajouter d'autre type et méthode de scan.

Références

[1] C. W. Laurent Bloch, chez *Sécurité informatique Principes et méthode* , p. 9.

[2] N. Khireddine, «Memoire de Master Gestion et implementation d'une application web sur virus Corona dans le monde».

[3] J. Flynn-York, « A Brief History of the Web,» [En ligne]. Available: https://blog.keepsite.com/a-brief-history-of-the-web-809509ba23df . [Accès le 20 Avril 2023].

[4] A. Rhillane, «Site dynamique vs site statique : Lequel choisir ? (Source: https://www.sortlist.fr/blog/site-dynamique-vs-statique/),» 11 Aout 2022. [En ligne]. Available: https://www.sortlist.fr/blog/site-dynamique-vs-statique/. [Accès le 5 Mai 2023].

[5] T. A. Khimoud Salem, «Mise en oeuvre et securisation d'une application de gestion de la formation professionnelle specialisee (FPS)».

[6] «Quels sont les avantages et les inconvénients d'une application web ?,» [En ligne]. Available: https://agence-scroll.com/blog/avantages-et-inconvenients-dune-application-web. [Accès le 22 Mai 2023].

[7] «Analyse des applications Web : tout ce que vous devez savoir,» 11 May 2022. [En ligne]. Available: https://www.oshyn.com/blog/web-application-scanning. [Accès le Juillet 2023].

[8] D. H. DEMIAI Ahmed, «Memoire de Master intitule 'Realisation d'un systeme d'audit de securite pour les applications web',» Blida, 2018.

[9] «What is a web server?,» [En ligne]. Available: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server. [Accès le 15 Avril 2023].

[10] «Overview,» [En ligne]. Available: https://developers.redhat.com/products/eap/overview. [Accès le 2023].

[11] «Features and benefits,» [En ligne]. Available: https://www.redhat.com/en/technologies/jboss-middleware/application-platform/features. [Accès le 2023].

[12] «Qu'est-ce que la sécurité des applications Web? - définition de techopedia,» 2023. [En ligne]. Available: https://fr.theastrologypage.com/web-application-security. [Accès le juillet 2023].

[13] «Audit de sécurité des applications Web,» 10 Aout 2022. [En ligne]. Available: https://loughtec.com/web-application-security-audit/. [Accès le Juillet 2023].

[14] «What is Kali Linux?,» [En ligne]. Available: https://www.kali.org/docs/introduction/what-is-kali-linux/. [Accès le 12 Avril 2023].

Reference

[15] R. GILLES, UML2 en action, de l'analyse des besoins en action, Presses de l'Université de Québec, 2009 , première édition.

[16] D. G. Joseph Gabay, UML 2 Analyse et conception Mise en oeuvre guidée avec études de cas, Dunod, 2008.

[17] «CSS : Feuilles de style en cascade,» 21 Septembre 2022. [En ligne]. Available: https://developer.mozilla.org/fr/docs/Web/CSS. [Accès le 1 Juin 2023].

[18] «Bootstrap Get Started,» [En ligne]. Available: https://www.w3schools.com/bootstrap/bootstrap_get_started.asp. [Accès le 1 Juin 2023].

Annexe :

Structure d'accueil

Présentation de Naftal

Fondée en 1982 et filiale 100 % du Groupe Sonatrach, Naftal est une société par actions (SPA) au capital social de 15 650 000 000 DA. Elle est rattachée à l'activité commercialisation.

Mission

NAFTAL a pour mission principale, la distribution et la commercialisation des produits pétroliers sur le marché national.

Elle intervient dans les domaines de :

- L'affûtage des GPL
- La formulation de bitumes
- La distribution, stockage et commercialisation des carburants, GPL, lubrifiants, bitumes pneumatiques, GPL/carburant, produits spéciaux.
- Transport des produits pétroliers.

Département d'accueil (DCSI)

Missions des quatre (04) directions de la DCSI et du département sécurité & conformité

- **Direction infrastructures**

Définir et mettre en œuvre l'architecture des systèmes, des bases de données et réseaux d'infrastructure du Système d'Information.

- **Direction solutions métiers**

Concevoir et réaliser des solutions informatiques qui répondent aux besoins opérationnels de l'ensemble des structures de la Société.

- **Direction opérations**

Veiller au bon fonctionnement des plateformes monétiques et décisionnelles.

Garantir la disponibilité du matériel informatique dédié aux utilisateurs finaux du système d'information de la société.

- **Direction informations documentaires**

Mettre en œuvre la politique de la Société en matière d'archivage des documents de la société.

Assurer la gestion du système documentaire en matière d'acquisition, traitement et diffusion d'informations.

- **Département sécurité & conformité**

Concevoir et mettre en place un dispositif permettant la sécurité et la pérennité des systèmes d'information mis en place.

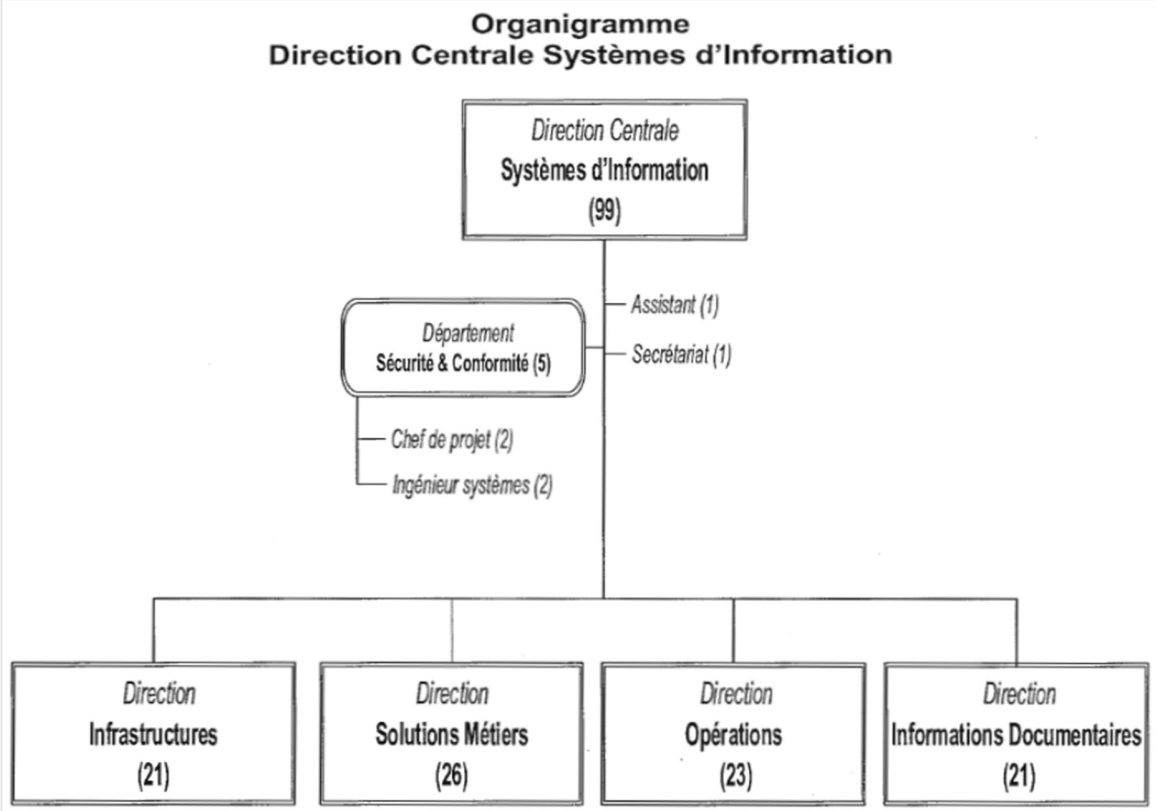

Figure 31 *: Organigramme Direction Centrale Système d'information.*