Manuel Technique
3PROJ

Réalisé par :
BERRADA Yahia

SOMMAIRE

La documentation technique qui va suivre explique toutes les caractéristiques technologique et informatique utilisé pour réaliser ce projet.

Le projet consistait en une architecture de sécurité pouvant détecter et identifier les attaques reçues afin de centraliser et de gérer toutes ces données de façon automatique.

1) Introduction

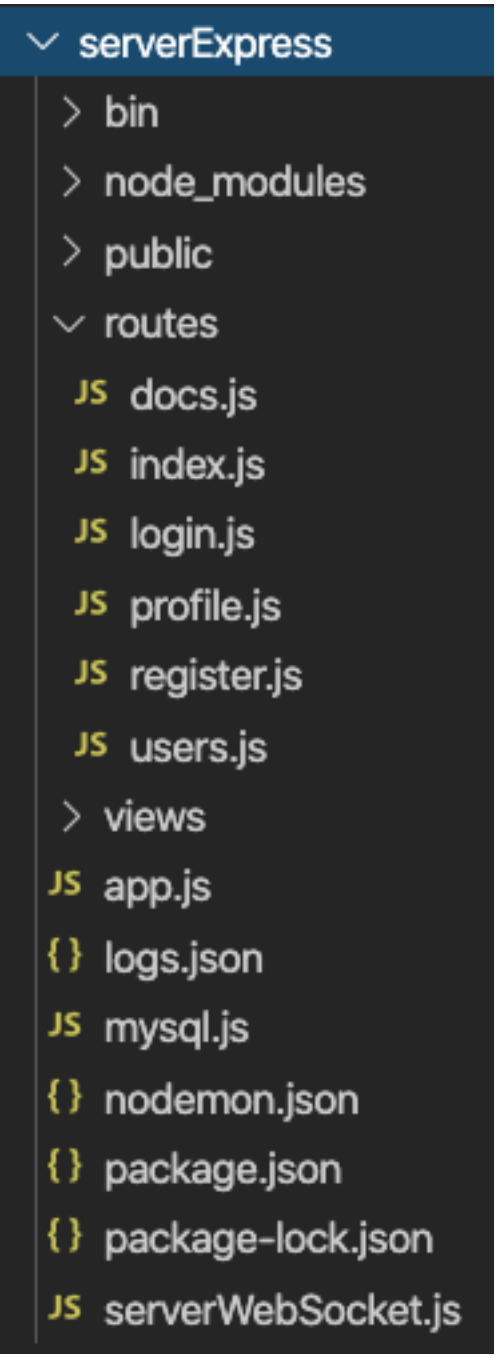
a. Architecture Client-Serveur

Le projet qui va vous être présenté dans cette documentation a principalement été réalisé avec une architecture Client-Serveur.

Cette architecture est très connue et très appréciée dans le monde professionnel.

L'architecture Client-Serveur consiste en un mode de communication à travers un réseau dans lequel le client envoie une requête et le serveur répond à celle-ci.

2) Serveur



Dans ce projet, le serveur joue plusieurs rôles, en effet ce serveur permet de lancer :

- Un serveur http pour le site web de l'entreprise
- Un serveur websocket pour gérer la communication
- Un serveur fs pour gérer les différents fichiers/documents

Ce serveur a entièrement été réalisé en Nodejs et plus précisément avec ExpressJS qui est framework Nodejs permettant de construire des applications web avec le modèle MVC.

L'utilisation de cette technologie pour le serveur a été un choix évident car Nodejs dispose de nombreuses librairies et d'une très forte documentation qui permet de mener à bien des projets de cette envergure.

Ce serveur est dans le dossier serverExpress du projet.

a. Serveur Websocket

Le Websocket permet d'établir une connexion avec le protocole réseau TCP pour les navigateurs web.

Le serveur websocket de ce projet permet de faire tourner l'application web du monitoring que l'on verra plus tard dans cette documentation.

Il permet en effet de tester la connexion afin de pouvoir afficher différents éléments sur la page web.
Ce serveur websocket permet également la détection des attaques qui seront réalisés par le client.

Suite à la détection de ces attaques, différents traitements seront alors actés comme : les mises à jour de la base de données concernant les attaques et les sections relatives à ces attaques, l'insertion des logs dans un fichier JSON et dans la base de données.

Le serveur Websocket de ce projet utilise le port 8080.

b. Serveur Fs

Le serveur fs qui signifie file system est un serveur qui va permettre de gérer le stockage de fichiers et de documents.

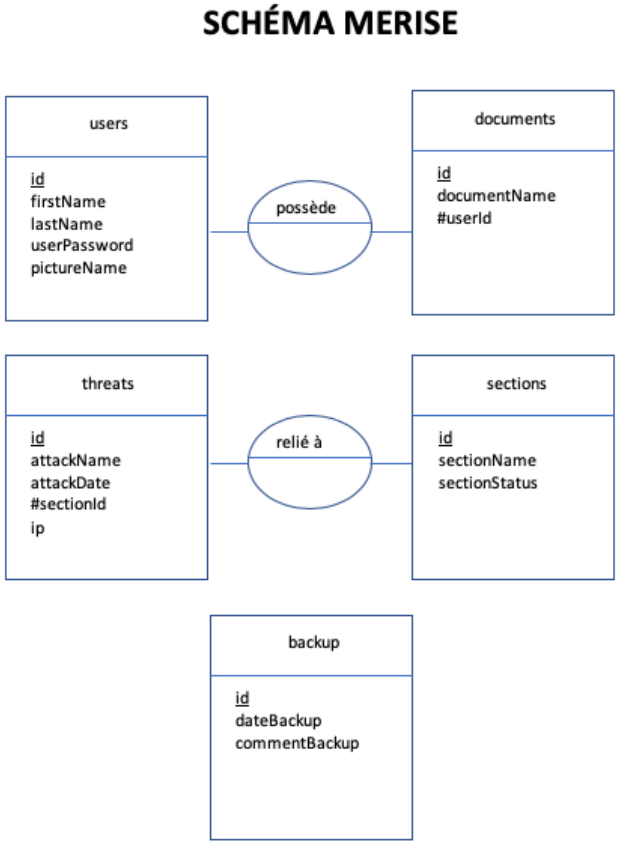
Dans ce projet, ce serveur est utile pour ce qui concerne les fichiers que les utilisateurs ajouteront dans leur plateforme web que nous verrons plus bas.

En effet, ce serveur va permettre une meilleure gestion des différents documents qui pourront être ajoutés par les utilisateurs.
Le serveur Fs de ce projet utilise le port 4000.

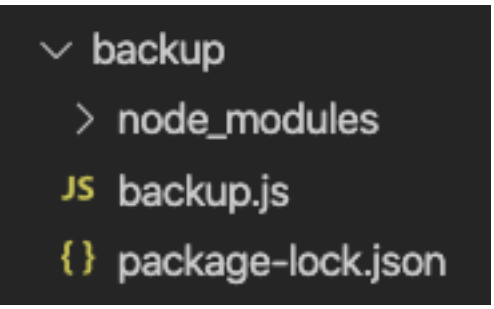
c. Base de données

La base de données utilisée pour ce projet contient les tables users, threats, sections, documents, backup.

Voici un diagramme merise illustrant cette base de données.



d. Backup



Les sauvegardes générées par ce projet sont de 2 types : les sauvegardes JSON qui comportent les attaques répertoriées et les sauvegardes SQL qui contiennent l'ensemble de la base de données.

i. Backup JSON

La sauvegarde Json s'effectue dans le fichier logs.json du serveur.

Ce fichier est actualisé à chaque nouvelle attaque détectée.

Ce fichier sera également utilisé par le script python intelligent qui permettra de filtrer et trier les données pertinentes de ce fichier de log.

ii. Backup SQL

La sauvegarde Sql s'effectue dans à l'aide du script backup.js se trouvant dans le dossier backup de ce projet.

Ce script permet de générer un fichier qui contient l'ensemble des données des tables Sql de la base de données 3proj.

e. Site web de l’entreprise

Le site web de l’entreprise est accessible sur le port 3000.

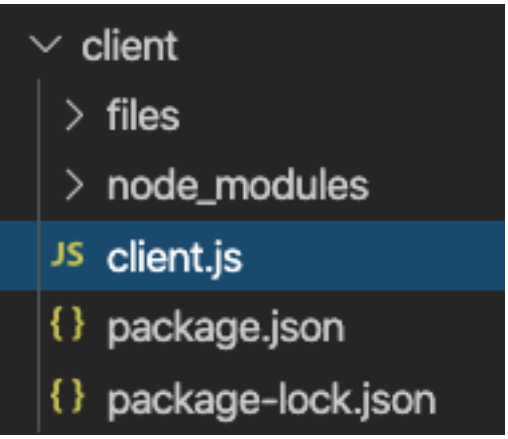
Ce site web contient différentes sections comme : un espace de connexion, d’inscription, de registre des documents, des utilisateurs, d’une page profile.

Ce site web va permettre de démontrer les attaques que peut subir l’entreprise de santé Health.

En effet, les différentes attaques que le client effectuera sur le serveur se feront par des failles présentes sur le site web de l’entreprise.

Les illustrations de cette plateforme sont disponibles dans le manuel utilisateur.

3) Client



Le client qui va vous être présenté ci-dessous et démontré dans le manuel utilisateur a été entièrement programmé en NodeJS également.

a. Client Websocket

La partie websocket du client permet d'envoyer des requêtes concernant différentes attaques au serveur.

Ce client websocket dispose aussi d'un menu et d'un système d'input pour choisir et lancer les différentes attaques.

b. Client Fs

Le client websocket est aussi un client fs qui va permettre de se connecter au serveur fs afin d'effectuer des attaques dessus, notamment l'attaque du backdoor qui consiste à injecter un fichier malicieux dans le serveur.

c. Attaques disponibles

i. Attaque de données

Dance ce client, l'attaque de données correspond au numéro 1 du menu.

Cette attaque permet de récupérer et de lire des documents utilisateurs qui ne sont pas sécurisés par le serveur et donc accessible par le client.

ii. Attaque DoS

L'attaque DoS comme expliqué dans la documentation générale de ce projet, est une attaque permettant de mettre hors service un serveur.

Cette attaque est assignée au numéro 2 du menu.

Le client va donc envoyer des milliers de requêtes inutiles au serveur afin de le surcharger et de le faire planter.

iii. Attaque Backdoor

L'attaque backdoor de ce client va permettre d'injection un fichier malicieux lors de la soumission du formulaire d'inscription du site web du serveur car ce dernier n'est pas sécurisé.

Le fichier malicieux pourra ensuite être appelé directement depuis le site web dans la rubrique documents.

Cette attaque est assignée au numéro 3 du menu.

iv. Attaque Flooding

L'attaque flooding du client va demander d'entrer un nombre qui sera utilisé pour faire des request http post sur le formulaire d'inscription du site web x fois et surchargera donc le serveur d'utilisateurs et de requêtes.

Cette attaque correspond au numéro 4 du menu.

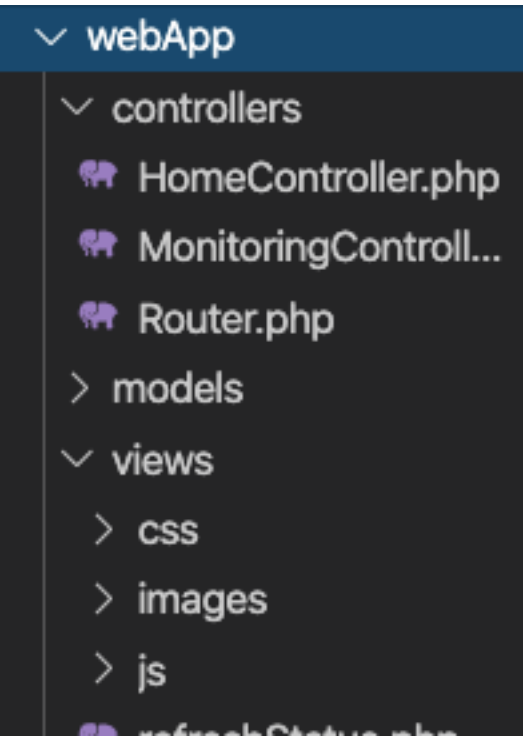
v. Attaque Sql injection

L'attaque d'injection Sql de ce client consiste à envoyer une requête sur le formulaire de connexion du site web non sécurisé.

Cette requête fait planter le serveur et rend le site inaccessible lors du lancement de cette attaque.

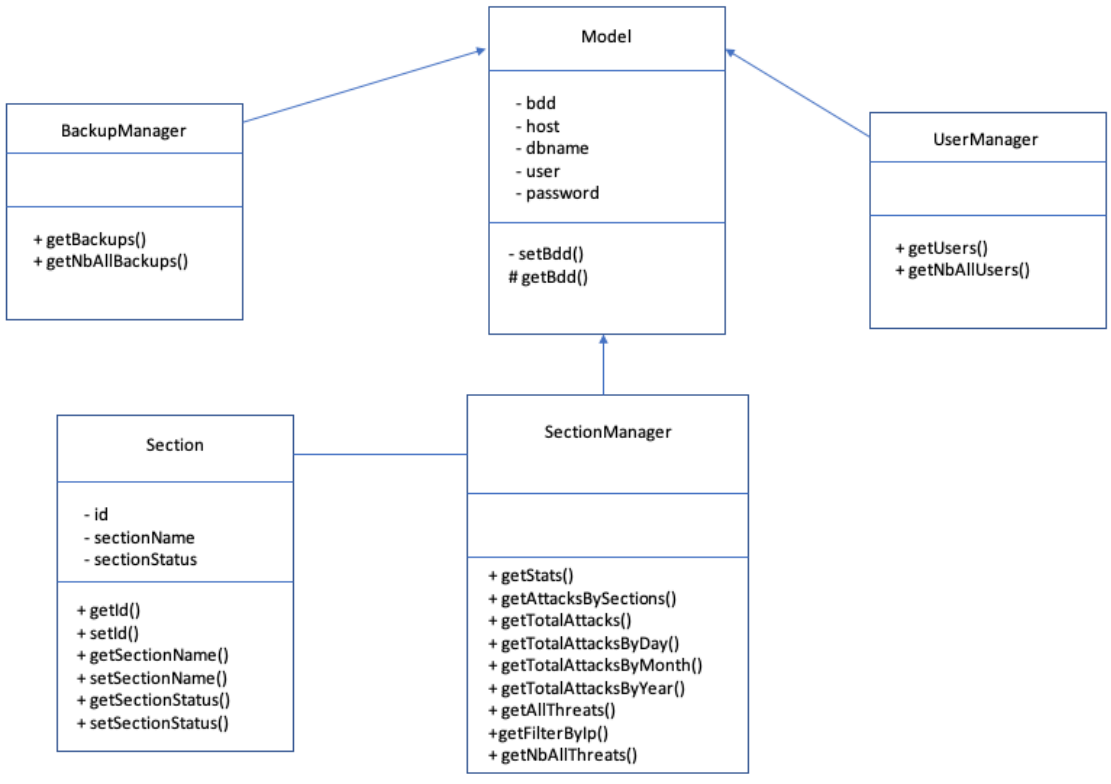
Cette attaque correspond au numéro 5 du menu.

4) Monitoring



Le monitoring de ce projet permet de visualiser graphiquement les différentes parties attaquées du système.
L'administration du monitoring permet également d'avoir une vue globale sur les différentes pages qui vous seront présentées ci-dessous.
Le monitoring de ce projet a entièrement été réalisé en orienté objet PHP avec MVC et PDO pour la connexion à la base de donnée.
Voici le diagramme de classe des Models de cette application de monitoring.

Diagramme de classes des fichiers du Models



a. Page d'accueil : vue globale des sections attaquées

La page d'accueil de ce monitoring permet de visualiser avec des animations l'état en temps réel des services du serveur, c'est-à-dire : l'état de la protection de la base de données, des utilisateurs, des fichiers malicieux, des attaques par déni de services.

Les données en temps réels sont affichées grâce à du Ajax.

b. Administration Monitoring

i. Statistiques

Les statistiques du monitoring permettent d'afficher les données d'attaques en utilisant des graphes, notamment des pie chart avec la libraire canvajs.

ii. Données utilisateurs

Les données utilisateurs sont récupérées à l'aide d'une jointure sql faisant une liaison avec la table users et la table documents afin d'afficher l'ensemble des utilisateurs et de leurs documents dans un tableau.

iii. Analyse des attaques

Les analyses d'attaques sont elles aussi réalisées avec des jointures et permettent d'afficher les données en fonctions des adresses IP.

iv. Historiques des sauvegardes

La partie backup de ce monitoring permet d'afficher l'historique des sauvegardes de base de données.

5) Script python intelligent

Ce script python intelligent permet d'analyser le fichier logs.json afin de l'afficher de manière visible et de pouvoir en tirer des informations réelles et statistiques.

Ce script possède aussi un menu qui permet soit de lister toutes les informations présentes dans le fichier log soit des informations spécifiques en entrant dans la console ce que l'on souhaite chercher.

Par exemple, avec ce script intelligent, il est possible de faire des recherches par ip, date, attaque, section.

6) Network Map