Projet 4NPM

Hillal AIMENE

ECOLE-IT

Table des matières

1. Introduction :

Le but de ce projet est de créer une application backend en utilisant Node.js, Express et Mongoose. Ce projet mettra en pratique les différentes parties et notions abordées durant le cours. Certains choix technologiques vous seront imposés pour encadrer au mieux le projet et rester dans le contexte du cours. Il est essentiel d'avoir bien suivi le cours et assimilé les différentes notions clé de Node.js, notamment son approche asynchrone et les différentes façons de l'aborder.

2. Contexte :

Vous êtes développeur(se) dans une entreprise qui travaille sur une solution un projet innovant dans la conception de batteries pour les voitures électriques. L'an dernier, votre entreprise a été victime d'un vol de données sensibles et confidentielles. Heureusement pour l'entreprise, la fuite a été identifié rapidement et il s'agit d'un employé qui a utilisé un ordinateur externe pour se connecter au réseau de l'entreprise et dérober les données. Pour éviter tout autre incident similaire, l'entreprise a décidé de vous confier la mission de mettre en place un système qui permet de détecter les intrusions dans le réseau. Vous êtes lead développeur et responsable de la mise en place de la solution. L'entreprise vous donne carte blanche sur le choix des outils à utiliser mais vous impose une contrainte de temps ; 3 semaines pour délivrer une première version fonctionnelle. Après une mure réflexion, vous avez décidé d'opter pour un server Node.js qui implémente une API qui fournit un monitoring sur les accès réseau en temps réel.
Pour vous aider à identifier les appareils externes à l'entreprise, celle-ci met à votre disposition un fichier dans lequel sont renseignées toutes les adresse MAC des matériels.

3. Énoncé :

Votre solution consiste à mettre en place un server Node.js capable de surveiller le fichier des logs du DHCP dans le but de détecter les potentielles intrusions dans le réseau. Lors de détection d'intrusion, votre système doit être capable de remonter l'information en temps réel vers l'interface de monitoring. Notez que dans ce projet, il n'est pas question de développer l'interface mais uniquement la partie backend.
En plus de remonter l'information en temps réel, votre server doit être capable de stocker l'information pour pouvoir la retrouver ultérieurement.

En ce qui concerne les accès, votre server doit être capable de créer un nouvel utilisateur, pourvoir l'authentifier et maintenir une session active.

Toutes les ressources de votre server, à l'exception de celle responsable de l'authentification, doivent être protégée de manière à nécessiter une authentification.

Autrement dit, un utilisateur demandant l'accès à une ressource et n'ayant pas été authentifié au préalable, ne doit pas pouvoir accéder à cette ressource.

4. Contraintes techniques :

Comme annoncé précédemment, vous êtes dans l'obligation de respecter certaines consignes, dont :

- Langage de programmation : Node.js
- Framework http : Express
- Validateur de données : express-validator
- Système de base de données : MongoDB
- OBD (Object Data Modeling) : Mongoose
- Protocole de communication en temps réel : web socket

5. Barème :

a) Ressources **(4points)** :

- Implémentation de la ressource pour ajouter un utilisateur : **1 point**
- Implémentation de la ressource pour authentifier un utilisateur : **1 point**
- Implémentation de la ressource pour supprimer la session d'un utilisateur (logout) : **1 point**
- Implémentation de la ressource pour récupérer toutes les intrusions dans le réseau : **1 point**

b) Fonctionnalités **(6 points)** :

- Votre server peut se lancer : **1 point**
- Votre server peut se connecter à une base de données : **1 point**
- Votre server peut maintenir une session : **1 point**
- Votre server est capable de délivrer l'information d'intrusion en temps réel : **2 points**
- Votre server est capable d'enregistrer une information d'intrusion : **1 point**

c) Sécurité **(3 points)** :

- Vous valider les informations reçues par l'utilisateur : **1 point**
- Vous gérez l'authentification d'une manière fiable : **1 point**
- Les mots de passes sont hashés et salés avant stockage : **1 point**

d) Bonne pratiques et propreté du code **(2 points)** :

- Vous avez utilisé les middlewares avec Express : **1 point**
- Vous avez bien divisé votre projet et organisé le code, écrit et produit un code lisible et facilement maintenable : **1 point**

e) Documentation **(1 point)** :

- Fournir une documentation technique de votre application qui doit expliquer les fonctionnalités de votre application, les technologies utilisées etc. : **0.5 point**
- Fournir un fichier readme.md qui explique comment faire les installations, lancer le server et obtenir les différentes ressources, par exemple en utilisant Postman : **0.5 point**

f) Bonus **(5points) :**

Dans cette partie vous êtes libre de proposer des fonctionnalités supplémentaires qui vont dans le sens de l'amélioration du projet. Ne seront pas pris en comptes les ajouts à des fonctionnalités déjà existantes, par exemple l'ajout de nouvelles routes ou nouvelles ressources. Toutefois, si vous êtes en manque d'inspiration, voici quelques pistes :

- Utilisation du https au lieu du http
- Protection contre les attaques CSRF
- Capacité de relance automatique en cas de crash
- Ajouter un système de rôles pour vos utilisateurs et restreindre la ressource qui récupère tous les utilisateurs pour le rôle « admin »

6. Astuces et techniques :

Pour pouvoir détecter les nouvelles entrées dans un fichier, vous pouvez utiliser la fonction wath du module fs
Pour vous faciliter la communication en protocole web socket, vous pouvez utiliser la librairie socket.IO

Pour pouvoir identifier tous les matériels externes au réseau vous pouvez répertorier les ceux de l'entreprise dans votre base de données grâce au fichier des adresse MAC de l'entreprise.
Le fichier de log DHCP est modifié dès qu'un matériel se connecte au réseau. Vous retrouverez des lignes suivants cette syntaxe :

```
March 14 14:00:15 medspx dhcpd: DHCPREQUEST for 192.168.34.69 from 0001.632C.299C via eth0b
March 14 14:00:13 medspx dhcpd: DHCPREQUEST for 192.168.74.1 from 0u:nc:l0:zq:79:0s via eth0b
March 14 13:58:17 medspx dhcpd: DHCPREQUEST for 192.168.52.45 from 7x:46:7e:g7:z4:4c via eth0b
March 14 13:56:45 medspx dhcpd: DHCPREQUEST for 192.168.5.64 from 0001.632C.299C via eth0b
March 14 13:40:03 medspx dhcpd: DHCPREQUEST for 192.168.50.36 from rx:mq:y0:8r:yb:w4 via eth0b
March 14 13:40:02 medspx dhcpd: DHCPREQUEST for 192.168.6.91 from 0001.632C.299C via eth0b
March 14 13:40:00 medspx dhcpd: DHCPREQUEST for 192.168.36.86 from lr:sm:04:2x:2s:9o via eth0b
March 14 13:34:48 medspx dhcpd: DHCPREQUEST for 192.168.63.81 from 3j:7b:9b:m1:75:33 via eth0b
March 14 13:33:26 medspx dhcpd: DHCPREQUEST for 192.168.62.41 from 0001.632C.299C via eth0b
```

Les informations qu'on retrouve dans ce fichier correspondent à :

- **Mach 14 13:33:26 :** Cela représente la date et l'heure du message. Dans cet exemple, il s'agit du 14 mai à 13h33 et 26 secondes.
- **medspx :** C'est le nom de l'hôte ou du serveur où le message a été généré.
- **dhcpd :** C'est le nom du service ou du programme qui a généré le message, dans ce cas, le serveur DHCP.
- **DHCPREQUEST :** C'est le type de message DHCP. Dans ce cas, il s'agit d'une demande DHCP émise par un client.
- **for 192.168.62.41 :** C'est l'adresse IP demandée par le client DHCP. Dans ce cas, le client demande l'adresse IP 192.168.62.41.
- **from 0001.632C.299C :** C'est l'adresse MAC du client DHCP qui a émis la demande.
- **via eth0b :** Cela indique par quelle interface réseau la demande DHCP a été reçue. Dans ce cas, la demande a été reçue via l'interface réseau eth0b.

7. Rappels des consignes :

Ce projet a pour but de faire pratiquer les différentes connaissances acquises durant le cours. Il est alors important d'essayer de le réaliser par ses propres moyens. Tout code qui ne provient pas de vous sera considéré comme une tentative de triche et vous serez sanctionné.
Considérez qu'il y aura des entretiens techniques dans lesquels il vous sera demandé d'expliquer votre code d'une manière technique. Si vous n'êtes pas capable de l'expliquer, vous serez considérés comme ayant recours à la triche.
Vous devez effectuer le rendu dans les temps impartis. Autrement vous recevrez une note nulle. Aucun rendu envoyé autrement qu'un dépôt sur la plateforme ne sera pris en compte.