BUT 3
Parcours
Informatique

Rapport TP4

IUT de Paris - Rives de Seine
Université Paris Cité

Node JS

Tony GUAN

1. Pour la première étape, lorsqu'on run le dev et qu'on essaye d'accéder au requête http://localhost:3000/secu et http://localhost:3000/dmz , on a : " "replique": "Tu ne sais rien, John Snow.."" et ""replique": "Ca pourrait être mieux protégé..."" . On encode ensuite Tyrion et wine qui sont l'username et le password puis on retest avec postman. l'encodage donne "VHlyaW9uOndpbmU=" j'ai eu à ce moment là un petit problème car j'avais rajouter un espace à la fin de ce que je voulais encoder. Ce qui m'a donné un encodage différent avant que je me rende compte que les espaces étaient aussi comptés. Une fois ce problème réglé, j'ai décoché ce que j'avais rajouté dans le header pour le mettre dans Auth avec les id en clairs. lorsqu'on regarde dans le code spinnet, on retrouve la ligne --header 'Authorization: Basic VHlyaW9uOndpbmU=' qui fait la même chose à notre place. La fonction after() va donc servir à vérifier une fois que les services sont lancés, que la route est bien mise en place avec les bons identifiants. Pour créer une route 'autre' avec l'accessibilité à tous, il suffit de copier le code et d'enlever la ligne du onRequest pour faire sauter l'authentification.

2. Pour cette deuxième étape on crée tout d'abord une clé avec la commande openssl genrsa -out server.key 2048. On crée ensuite un fichier Certificate Signing Request avec la commande openssl req -new -key server.key -out server.csr puis on le signe avec openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt qui est valide 365 jours. On le teste ensuite avec openssl s_server -accept 4567 -cert server.crt -key server.key -www -state et on voit que ça nous retourne ACCEPT. On teste ensuite sur postman et on obtient une page web avec des TLS. On rajoute dans le logger des paramètres en https pour sécuriser l'accès.

3. Pour générer les clés de chiffrement, on utilise la commande openssl ecparam -genkey -name prime256v1 -noout -out .ssl/ec_private.pem dans le repertoire ssl pour la clé privée. Ensuite, à partir de la clé privée, on génère la clé publique avec openssl ec -in ec_private.pem -pubout -out ec_public.pem . Ensuite il faut compléter dans les différents fichiers fournis.(Merci à chat GPT)