NoSQL - TP REDIS

Groupe Cyber/Infra

Consigne

Ce TP consiste en la **mise en place d'un système d'authentification reposant sur cluster Redis**.

L'idée est de stocker les utilisateurs, les sessions, les OTP sur un **cluster Redis hardené** et **persisté**.

Chaque groupe sera composé d'étudiants cyber **ET** infra.

Chaque partie devra répondre sur le rendu aux question posées.

Partie cyber

La partie cyber s'occupera de **l'implémentation de la stratégie d'authentification** (langage recommandé NodeJS ou Python).

L'API ne proposera aucune vue, que du json.

- `/login`
  - La session est créée.
- `/register`
  - L'utilisateur est créé.
  - Le mot de passe est stocké en bcrypt. **Pourquoi ?**
- `/logout`

    - La session est supprimée.
- `/otp` permet de récupérer un OTP (One Time Password) afin de se connecter malgré l'oubli de son mot de passe.
    - L'OTP est envoyé par mail ou telegram à l'utilisateur.
    - Au bout de 3 essais infructueux l'OTP est invalidé, un nouveau est envoyé. **Pourquoi ?**
    - L'OTP n'est utilisable qu'une fois.
    - L'OTP a une durée de validité définie (il expire au bout d'un certain temps).

L'idée ici est de bien implémenter les communications avec Redis en utilisant les fonctions pertinentes.

Le code devra être réalisé en tenant compte du **Top10 OWASP** (Notamment sur la partie Injection NoSQL).

Un article intéressant sur une mauvaise pratique :

https://medium.com/@PatrickSpiegel/https-medium-com-patrickspiegel-nosql-injection-redis-25b332d09e58

Comment s'en protéger ?

Partie infra

Pour soutenir cette API la partie infra devra mettre à disposition une infrastructure Redis précise :

- cluster Redis avec **3 noeuds** -> si je supprime un ou deux noeuds le cluster est toujours fonctionnel.
- **Persistence des données** du cluster -> si je supprime tous les noeuds j'ai encore les données.
- **Hardening** du cluster Redis -> protection par mot de passe, isolation...

Quelle est votre choix de stratégie pour la mise en cluster ? Pourquoi ?

Quels sont les risques d'un cluster non hardené ?

Barème

La notation sera la suivante :

Cluster Redis

Chaque point doit être démontré par le test adéquat.

- Redis
    - UP -> 1 pts
    - en cluster -> 2 pts
    - persisté -> 2 pts
    - hardené -> 2 pts
    - hardené ++ -> 1 pts
- Questions -> 2 pts

API

- Code des routes
    - `/login` -> 2 pts

    - `/logout` -> 1 pts
    - `/register` -> 1 pts
    - `/opt` -> 2 pts
- API up et fonctionnelle -> 2 pts
- Questions -> 2 pts