INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

INSA de Lyon
Département Télécommunications,
Services et Usages

5TC-SRS
TP PostgreSQL / RBAC

Durée : 4h

Ce TP présente et applique les notions de contrôle d'accès à travers le modèle RBAC et la découverte des possibilités intégrées aux SGBD. Même si le SGBD ne fournira pas toujours l'expressivité adaptée à l'application visée, l'utilisation de ses primitives permet de confiner les brèches éventuelles, y compris pour des applications bien développées qui ne seront néanmoins pas parfaites.

Ce TP sera réalisé dans la VM "Debian-jessie-amd64", dont le script de lancement est à télécharger sur Moodle. Les comptes sont `debian/debian` et `root/root`.

1 Objectif

Vous êtes DBA (*Database Admin*). Une application est en cours de déploiement sur le SI et vous devez gérer la partie BD. Vous pouvez obtenir une copie de l'application *webapp.tar.gz* (sur Moodle) pour l'analyser et proposer des modifications à la marge. L'application est déjà installée dans la VM, dans le dossier `/var/www/html/webapp` et est accessible (depuis la VM) à l'URL `http://localhost/webapp/`. Un fichier de description `README.txt` est disponible à la racine.

Étant donnée la taille de l'application et la répartition des missions dans l'entreprise, il n'est pas concevable de modifier toutes les requêtes. La recette a déjà été faite, les modifications ne sont plus possibles qu'à la marge, le déploiement aura lieu dans la semaine.

2 Contrôle d'accès avec RBAC

L'application étant très imparfaite, commencez par y exploiter une vulnérabilité.

Ceci étant fait, vous proposez de raffiner l'authentification entre l'application et la BD afin de limiter les dégâts potentiels :

— Un utilisateur de l'application web = un utilisateur de la BD

— L'authentification sera réalisée directement avec la BD (stockages des authentifiants dans la session PHP)
— Des rôles (hiérarchiques) pour factoriser la gestion des droits
— Une sécurité à grain fin au niveau des lignes de tables (*Row security*)

Proposez (sur papier) et faîtes valider un modèle RBAC adapté. Déployez ce modèle et mettez à jour le code PHP en fonction (vous pouvez utiliser `pgadmin3` comme interface graphique à postgresql). Vérifiez que votre exploitation initiale ne fonctionne plus.

REMARQUE : En l'absence de *row-level security*, un résultat relativement similaire (mais plus complexe à maintenir) aurait pu être obtenu avec l'utilisation de vues.

3 Séparation des pouvoirs

Afin d'améliorer la qualité (l'intégrité) des données, proposez une mise en œuvre permettant la séparation des pouvoirs lors de l'ajout d'un nouveau client. Il faudra pour cela une table intermédiaire, stockant la transaction en cours de réalisation.

4 Documentations utiles

— Gestion des rôles : `https://www.postgresql.org/docs/current/static/user-manag.html`
— GRANT : `https://www.postgresql.org/docs/9.5/static/sql-grant.html`
— Row security policies : `https://www.postgresql.org/docs/9.5/static/ddl-rowsecurity.html`