SAGIM

SGBD I
MySql Server

Optimisation & Sécurité

MySQL est l'un des systèmes de gestion de bases de données relationnelles (SGBDR) les plus populaires au monde. Cependant, comme tout système informatique, il est vulnérable à des attaques si des mesures de sécurité appropriées ne sont pas mises en place.

Supprimez les comptes anonymes : Les comptes anonymes permettent l'accès sans authentification, ce qui est un risque de sécurité.

Désactivez la connexion root à distance : Limitez l'accès root à la machine locale pour éviter les attaques à distance.

Supprimez la base de données de test : La base de données de test est souvent inutile et peut être une cible pour les attaquants.

mysql_secure_installation

SAGIM

ouvrez l'invite de commandes :

Appuyez sur Win + R, tapez cmd et appuyez sur Entrée.

Exécutez le script :

Tapez la commande suivante :

mysql_secure_installation

Répondez aux questions du script :

Supprimer les comptes anonymes : Répondez Y (Yes).

Remove anonymous users? [Y/n] Y

Désactiver la connexion root à distance : Répondez Y (Yes).

Disallow root login remotely? [Y/n] Y

Supprimer la base de données de test : Répondez Y (Yes).

Remove test database and access to it? [Y/n] Y

Sécuriser l'installation avec des requêtes :

Supprimer les utilisateurs anonymes : Exécutez la commande suivante dans MySQL :

```
DELETE FROM mysql.user WHERE User='';
```

Désactiver les connexions root à distance : Exécutez la commande suivante :

```
DELETE FROM mysql.user WHERE User='root' AND Host != 'localhost';
```

Supprimer la base de données de test : Exécutez la commande suivante :

```
DROP DATABASE IF EXISTS test;
```

Recharger les tables de privilèges : Exécutez la commande suivante :

```
FLUSH PRIVILEGES;
```

Création d'Utilisateurs :

Créez des utilisateurs spécifiques pour chaque application ou service :

```
CREATE USER 'nom_utilisateur'@'localhost' IDENTIFIED BY 'mot_de_passe';
```

Accordez uniquement les permissions nécessaires :

```
GRANT SELECT, INSERT, UPDATE ON base_de_donnees.*
TO 'nom_utilisateur'@'localhost’;
FLUSH PRIVILEGES;
```

2.3. Révoquer les Permissions

```
REVOKE ALL PRIVILEGES ON base_de_donnees.*
FROM 'nom_utilisateur'@'localhost';
```

Gestion des Utilisateurs et des Permissions :

Pour vous connecter à **MySQL** en utilisant un utilisateur spécifique, vous devez utiliser la commande **MySQL** dans votre terminal ou ligne de commande.

En spécifiant l'utilisateur avec l'option **-u** et en fournissant le mot de passe avec l'option **-p**.

Voici la syntaxe générale :

```
mysql -u nom_utilisateur –p
```

Spécifier l'hôte :

Si MySQL est hébergé sur un serveur distant, vous pouvez spécifier l'hôte avec l'option -h :

```
mysql -h nom_du_serveur -u nom_utilisateur -p
```

spécifier la base de données avec l'option -D :

```
mysql -u nom_utilisateur -p -D nom_de_la_base
```

Sécurisation des Données :

SAGIM

Chiffrez les connexions avec SSL/TLS :

```
GRANT USAGE ON *.* TO 'nom_utilisateur'@'localhost' REQUIRE SSL;
```

Effectuez des sauvegardes régulières : Utilisez **mysqldump** pour créer des sauvegardes de vos bases de données.

```
mysqldump -u root -p --all-databases > backup.sql
```

Activez les logs d'audit : Configurez MySQL pour enregistrer les activités suspectes.

```
SET GLOBAL log_output = 'FILE';
SET GLOBAL general_log_file = 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Data\\mysql.log';
SET GLOBAL general_log = 'ON';
```

Limitez le nombre de tentatives de connexion : Pour prévenir les attaques par force brute.

```
CREATE USER 'nom_utilisateur'@'localhost'
IDENTIFIED BY 'mot_de_passe'
WITH MAX_CONNECTIONS_PER_HOUR 10;
```

SAGIM 2024/2025

TP de synthèse

TP de synthèse

Exercice 1 : Création d'un Utilisateur avec des Permissions Limitées:

Énoncé :

1. Créez un utilisateur **app_user** avec le mot de passe **SecurePass123**!.
2. Accordez-lui uniquement les permissions SELECT et INSERT sur la base de données **app_db**.
3. Vérifiez que l'utilisateur ne peut pas exécuter de commandes **DELETE** ou **UPDATE**.

TP de synthèse

```
-- Étape 1 : Création de l'utilisateur
    CREATE USER 'app_user'@'localhost' IDENTIFIED BY 'SecurePass123!';
-- Étape 2 : Attribution des permissions
    GRANT SELECT, INSERT ON app_db.* TO 'app_user'@'localhost’;
    FLUSH PRIVILEGES;
-- Étape 3 : Vérification des permissions
    SHOW GRANTS FOR 'app_user'@'localhost';
```

Exercice 2 : Configuration de l'Accès SSL pour un Utilisateur

Énoncé :

1. Créez un utilisateur secure_user avec le mot de passe SslPass456!.
2. Configurez cet utilisateur pour qu'il ne puisse se connecter que via une connexion SSL.
3. Vérifiez que la connexion sans SSL est refusée.

TP de synthèse

```
-- Étape 1 : Création de l'utilisateur
    CREATE USER 'secure_user'@'localhost' IDENTIFIED BY 'SslPass456!';
-- Étape 2 : Imposition de la connexion SSL
    GRANT USAGE ON *.* TO 'secure_user'@'localhost' REQUIRE SSL;
    FLUSH PRIVILEGES;
-- Étape 3 : Vérification des permissions
    SHOW GRANTS FOR 'secure_user'@'localhost';
```