Mecanismos de seguridad básicos en un SGBD

Mecanismo para la verificación de la identidad de los usuarios

- Identidad: Forma de identificarse un usuario (decir quién soy). Una identidad puede describir a un usuario,a un grupo de usuarios (rol), puede tener una validez limitada en el tiempo o en número de usos o puede depender de otros criterios. Es una información en principio pública.
- Autenticación: Forma de verificar un usuario su identidad. Es una información privada.

Mecanismo para el control del acceso

- Permiten controlar qué usuarios pueden acceder a qué datos y qué pueden hacer con ellos. Es decir, controla la interacción entre sujetos (usuarios y procesos) y los objetos a los que acceden (tablas, esquemas, funciones, etc.).
- Está formado fundamentalmente por dos componentes:
  - Políticas de acceso: criterios para otorgar o denegar permisos de acceso a un objeto del sistema de información.
  - Mecanismos de seguridad: todos aquellos procedimientos que velarán por el cumplimiento de las políticas de acceso.

Gestión de usuarios

- La interfaz en MySql para la gestión de usuarios está formada por las instrucciones: CREATE USER, GRANT, REVOKE.
- Internamente el servidor almacena la información en el esquema *mysql*.
- La identidad viene determinada por el equipo desde el que se accede y por el nombre de usuario que se especifique.
- Para averiguar los privilegios de una identidad se puede utilizar:

```
show grants for 'usuario'@'equipo'
```

- El control de acceso de MySql funciona en dos pasos:
  - Paso 1: Verificación de identidad mediante autenticación antes de conceder una conexión
  - Paso 2: Una vez iniciada la sesión, ante cada sentencia que desee ejecutar la identidad el sistema verifica que dispone de los privilegios pertinentes.

Ejemplo-práctico

- Mirar permisos del usuario ‘phpmyadmin’@’localhost’
- Consultar los mismos permisos pero utilizando la base de datos *mysql*
- Gestión de permisos en el manual: ¿dónde?

Creación de usuarios y eliminación

- Sintaxis:

```
CREATE USER user_specification [, user_specification] …
user_specification:
user [IDENTIFIED BY [PASSWORD] 'password']
```

-
- Las reglas básicas para la creación de nombres son:
  - Sintaxis para una identidad: 'nombre_usuario'@'nombre_equipo'
  - Las comillas son necesarias en el nombre de usuario o de equipo cuando contiene algún carácter especial como '-' o %.
  - Una identidad que sólo dispone de un nombre de usuario equivale a 'nombre_de_usuario'@'%'.
  - Los metacaracteres % y _ colocados como parte del nombre_equipo tienen el mismo significado que en una comparación de literales utilizando LIKE.
  - Se puede crear usuarios anónimos dejando el nombre_usuario como un espacio en blanco: ' '@'localhost'
  - El nombre_de_equipo se puede indicar con el nombre del equipo o con la IP. Es decir, 'localhost' y '127.0.0.1' son equivalentes.

- Algunos ejemplos:

```
CREATE USER pablo@’a23.iesjoandaustria.org’ IDENTIFIED BY ‘1234’;
CREATE USER pepe@’192.168.0.%’ IDENTIFIED BY '1234'
```

- Eliminación de usuarios:

```
DROP USER user [,user]...
```

Concesión de privilegios

- Para la gestión de privilegios tenemos la sentencia GRANT. Veámosla:

GRANT **priv_type** [(column_list)][, **priv_type** [(column_list)]] ...ON [**object_type**] **priv_level** TO **user_specification** [, **user_specification**] ...[WITH **with_option** ...]

object_type: TABLE | FUNCTION | PROCEDURE

priv_level: *.* | db_name.* | db_name.tbl_name | tbl_name | db_name.routine_name

user_specification: user [IDENTIFIED BY [PASSWORD] 'password']

with_option: GRANT OPTION | MAX_QUERIES_PER_HOUR count | MAX_UPDATES_PER_HOUR count | MAX_CONNECTIONS_PER_HOUR count | MAX_USER_CONNECTIONS count

- Para poder usar GRANT se debe disponer del privilegio GRANT OPTION.
- Hay que acudir al manual para averiguar qué privilegios soporta nuestra versión de MySql.
- La cláusula WITH se utiliza para:
    - Permitir a un usuario dar privilegios a otro (siempre y cuando el primero los tenga, claro). Para ello utilizaremos WITH GRANT OPTION
    - Limitar el uso de los recursos del servidor: limitando número de consultas, modificaciones, conexiones,etc.

Revocación de privilegios

REVOKE **priv_type** [(column_list)] [, **priv_type** [(column_list)]] ...ON [**object_type**] **priv_level** FROM **user** [, **user**] ...

O también:

REVOKE ALL PRIVILEGES, GRANT OPTION FROM **user** [, **user**] ...

Problemas: DCL-1

- Crea los siguientes usuarios y asígnales los permisos indicados en la bbdd 'ConsultasBásicas':
  - María@CualquierEquipo: permisos para consultar cualquier vista de la bbdd y modificarla(Alter view).
  - Pep@CualquierEquipoDe192.168.0.*: permisos para consultar la tabla “emple” y “depart”. Puede otorgar los mismos permisos a otros usuarios aunque no puede crear usuarios. El número máximo de consultas que puede realizar es de 10 consultas por hora.
  - Administrador@localhost: será el administrador de esta bbdd. Tendrá, por tanto, permisos para crear tablas, vistas, rutinas, modificarlas, eliminarlas, bloquearlas, gestionar usuarios de esa bbdd.
  - Invitado@CualquierEquipo: permisos de consulta en la columna “dni” de la tabla “personas” y “apellido” en “emple”.
  - Revoca la capacidad de modificar vistas a María.

Problemas: DCL-2

Ejercicio por parejas:

- Cada elemento de la pareja, creará un usuario y le otorgará unos permisos muy concretos. Para ello se utilizará la interfaz phpmyadmin.
- A continuación, el compañero se conectará al sistema del otro y tratará de averiguar qué permisos le han sido concedidos.
- Para realizar esta conexión habrá de utilizar primero el terminal y, en caso de no haber problemas, utilizará a continuación, la interfaz phpmyadmin.