Comandos basicos SQL para Pentesting (WIP)

2020-09-07

En este articulo se van a hablar de los comandos mas relevantes para tratar Bases de Datos SQL. Se van a tratar los principales motores de BD. El objetivo es ser capaz de realizar las principales operaciones que se pueden realizar en un pentesting. Tanto desde el punto de vista de conectarse a una BD victima como de desplegar nuestra propia BD para realizar pruebas.

Los motores de BD que se van a tratar son los siguientes:

- Familia MySQL (MySQL, MariaDB, etc)
- PostgreSQL
- Oracle
- SQL Server

Familia MySQL (MySQL, MariaDB, etc)

El primer paso sera conectarnos a la base de datos, esto se puede realizar de multiples formas, pero la mas rápida es realizarlo desde con consola con el comando *mysql*.

```
mysql -h 127.0.0.1 -uroot -ppass [-D db_name]
```

Si no se quiere poner la contraseña en el propio comando simplemente hay que dejar el argumento *-p* vacio y el comando *mysql* te pedira la contraseña.

Se puede indicar la BD a la que nos conectamos de forma opcional con el argumento *-D*.

Visualizar bases de datos y tablas

Una vez que estamos conectados a la BD, podremos ejecutar los siguientes comandos:

- **show databases**: Para mostrar las BD disponibles.
- **use test_db**: Para seleccionar la BD test_db.
- **show tables**: Para mostrar las tablas disponibles en la BD test_db.
- **desc table1**: Para mostrar las columnas de la tabla table1.
- **show columns from table1**: Para mostrar las columnas de la tabla table1.

Aqui se muestran los comandos descritos anteriormente. Es importante destacar que todos los comandos tienen que terminar el punto y coma (**;**).

```
show databases;

use test_db;

show tables;

desc table1;

show columns from table1;  # es igual a desc
```

Gestión de bases de datos y tablas

```
create database test_db;
```

```
CREATE TABLE table3(
    id int(50) not null auto_increment primary key,
    user varchar(35),
    pass varchar(50),
    description varchar(50) default 'bato'
);
```

drop table [table name];

select 1;

Gestion de usuarios

```
CREATE USER 'user1'@'localhost' IDENTIFIED BY 'pass1';
DROP USER user1@‘localhost’;
```

cambiar contraseña de un usuario, que no sea root, ya que para este es necesario parar el serivicio y colocar este comando en un fichero de autoarranque de mysql.

```
ALTER USER 'user1'@'localhost' IDENTIFIED BY 'pass2';
```

Gestion de permisos

```
GRANT [permiso] ON [databse].[table] TO '[user1]'@'localhost';
GRANT ALL PRIVILEGES ON *.* TO 'user1'@'localhost';
```

Una vez que has finalizado con los permisos que deseas configurar para tus nuevos usuarios, hay que asegurarse siempre de refrescar todos los privilegios.

```
FLUSH PRIVILEGES;
```

Si necesitas remover un permiso, la estructura es casi idéntica a la que los asigna:

```
REVOKE [permiso] ON [databse].[table] FROM '[user1]'@'localhost';
```

Dump de la base de datos y restauración

```
/bin/mysqldump -c -u admin -ppassword databasename tablename > /tmp/databasename.tablename.sql

bin/mysql -u admin -ppassword databasename < /tmp/databasename.sql
```