Què és SQL?

SQL és un llenguatge de programació estandarditzat (ANSI el 1986 i ISO el 1987) que s'utilitza per gestionar bases de dades relacionals i realitzar operacions diverses sobre les dades que contenen.

Una base de dades és una col·lecció de dades organitzades en files, columnes i taules, i indexada per facilitar la cerca d'informació rellevant.

Exemple de taula SQL amb dades d'empleats:

Nom de la taula: employees

userid	first_name	last_name	department	salary	auth_tan
32147	Paulina	Travers	Comptabilitat	$46.000	P45JSI
89762	Tobi	Barnett	Desenvolupament	$77.000	TA9LL1
96134	Bob	Franco	Màrqueting	$83.700	LO9S2V
34477	Abraham	Holman	Desenvolupament	$50.000	UU2ALK
37648	John	Smith	Màrqueting	$64.350	3SL99A

Una empresa guarda la informació següent sobre els seus empleats a les bases de dades: un número únic d'empleat (userid), el cognom, el nom, el departament, el salari i un número d'autenticació per transaccions (auth_tan). Cada columna representa un tipus de dades, i cada fila correspon a un empleat.

Les consultes SQL es poden utilitzar per modificar una taula, afegir o eliminar files de dades i canviar la seva estructura.

Tipus principals d'ordres SQL

1. Llenguatge de Manipulació de Dades (DML)

Com el seu nom indica, el llenguatge de manipulació de dades (DML) tracta amb la manipulació de dades. Les instruccions més comunes de SQL, com ara **SELECT, INSERT, UPDATE** i **DELETE**, es classifiquen com a ordres DML.

Les ordres DML poden ser utilitzades per:

- Sol·licitar registres (**SELECT**).
- Afegir registres (**INSERT**).
- Eliminar registres (**DELETE**).

- Modificar registres existents (**UPDATE**).

Exemple d'instrucció SELECT:

```
SELECT phone
FROM employees
WHERE userid = 96134;
```

Aquest exemple recupera el número de telèfon de l'empleat amb el userid 96134.

2. Llenguatge de Definició de Dades (DDL)

El llenguatge de definició de dades (DDL) inclou ordres per definir estructures de dades, com ara l'esquema d'una base de dades, que inclou objectes com taules, índexs, relacions, vistes, i més.

Les ordres DDL més comuns són:

- **CREATE:** Crear objectes de base de dades com taules o vistes.
- **ALTER:** Modificar l'estructura d'una base de dades existent.
- **DROP:** Eliminar objectes de la base de dades.

Exemple de CREATE:

```
CREATE TABLE employees(
   userid varchar(6) not null primary key,
   first_name varchar(20),
   last_name varchar(20),
   department varchar(20),
   salary varchar(10),
   auth_tan varchar(6)
);
```

Aquesta instrucció crea la taula d'empleats de l'exemple anterior.

3. Llenguatge de Control de Dades (DCL)

El llenguatge de control de dades (DCL) s'utilitza per implementar lògica de control d'accés a la base de dades, com ara concedir o revocar privilegis.

- **GRANT:** Concedeix permisos d'accés a objectes de la base de dades.
- **REVOKE:** Revoca permisos d'accés prèviament concedits.

Exemple d'instrucció GRANT:

GRANT SELECT ON employees TO user1;

Aquest exemple permet que user1 pugui consultar la taula employees.

Què és una injecció SQL?

La injecció SQL (o SQLi) és una de les tècniques d'atac web més comunes. Consisteix en la inserció de codi maliciós en una consulta SQL a través de les dades d'entrada proporcionades per l'usuari.

Exemple d'injecció SQL

Considera una aplicació web que permet als usuaris recuperar informació simplement introduint un nom d'usuari en un formulari. Les dades introduïdes per l'usuari s'envien al servidor i s'insereixen en una consulta SQL processada per un intèrpret SQL.

La consulta SQL per recuperar informació de l'usuari seria:

SELECT * FROM users WHERE name = '" + userName + "';

La variable userName conté l'entrada del client i l'"inserta" dins la consulta.

- Si l'entrada fos Smith, la consulta resultaria:
- SELECT * FROM users WHERE name = 'Smith';

Això recuperaria totes les dades de l'usuari amb el nom Smith.

- **Si un atacant introdueix dades malicioses** (exemple: caràcters com ', ;, o --) i aquestes no són filtrades, podria modificar el comportament previst de la consulta SQL per realitzar accions malicioses.

Conseqüències d'una injecció SQL

Un atac d'injecció SQL reeixit pot permetre a un atacant:

- Llegir i modificar dades sensibles de la base de dades.
- Executar operacions administratives.
- Desactivar registres o auditories del sistema de gestió de bases de dades (DBMS).
- Eliminar taules o registres.
- Afegir nous usuaris.
- Recuperar contingut d'arxius del sistema de fitxers del DBMS.

- Executar comandes al sistema operatiu.

Altres possibles efectes:

- Suplantació d'identitat.
- Manipulació de dades existents.
- Anul·lació de transaccions o modificació de saldos.
- Revelació completa de totes les dades del sistema.
- Destrucció o inaccessibilitat de dades.
- Obtenir privilegis d'administrador al servidor de bases de dades.

Injecció SQL amb cadenes de text

Si una aplicació construeix consultes SQL simplement concatenant cadenes introduïdes per l'usuari sense sanititzar, aquesta és altament vulnerable a injeccions SQL amb cadenes.

Exemple:

- L'aplicació permet l'entrada d'un nom per obtenir dades d'un usuari.
- Si no es valida correctament, una entrada com aquesta:
- ' OR '1'='1'

pot transformar la consulta inicial per obtenir dades de tots els usuaris.

Què és Query Chaining?

El chaining (encadenament de consultes) consisteix a afegir una o més consultes al final de la consulta original. Això es fa utilitzant el metacaràcter ;, que marca el final d'una instrucció SQL i permet començar una altra.

Comprometre la disponibilitat

Un atacant podria:

- Eliminar comptes d'usuaris o canviar les seves contrasenyes, impedint-los accedir-hi.
- Eliminar parts de la base de dades o fins i tot tota la base de dades, fent les dades inaccessibles.
- Revocar els drets d'accés d'administradors, bloquejant-los totalment.
- El teu objectiu és eliminar completament aquesta taula abans que ningú ho descobreixi.