Teoria e Descrizione dell'Attacco

SQL Injection (SQLi):

- **SQL Injection** è una vulnerabilità che permette a un utente malintenzionato di inserire codice SQL arbitrario all'interno di query SQL, manipolando le interazioni con il database. Se l'input dell'utente non viene correttamente filtrato o sanitizzato, un attaccante può eseguire query SQL maligne.
- **Descrizione del Problema**:
    - In questo caso, la query SQL viene costruita dinamicamente con l'input dell'utente senza sanificazione:

*query = f"SELECT * FROM users WHERE username = '{username}' AND password = '{password}'"*
Questo approccio permette l'iniezione di codice SQL. Inserendo `1' OR '1'='1` come nome utente, l'attaccante manipola la logica della query, trasformandola in una condizione sempre vera (`OR '1'='1'`).

La vulnerabilità permette l'autenticazione senza conoscere la vera password o il nome utente.

Come Prevenire:

- Utilizzare **query parametriche/preparate** per separare la logica della query dagli input dell'utente, ad esempio:

*cursor.execute("SELECT * FROM users WHERE username = %s AND password = %s", (username, password))*

CWE Involti

1. **CWE-89: SQL Injection**:
    - Questa vulnerabilità è l'esempio classico di SQL Injection, dove l'applicazione non sanitizza correttamente gli input forniti dall'utente, permettendo l'esecuzione di comandi SQL arbitrari.