Week 2: SQL Injection (SQLi) & Insecure Direct Object References (IDOR)

🔹 SQL Injection (SQLi)

What is it?

- SQLi occurs when user input is improperly sanitized, allowing attackers to manipulate SQL queries.
- This can lead to unauthorized data access, data manipulation, or even complete database compromise.

How to find it?

- Test input fields (login, search, etc.) by injecting SQL payloads like:

  ' OR 1=1 --

  ' UNION SELECT null, null, username, password FROM users --

- Look for errors in responses or unexpected behavior (e.g., bypassing login).

How to prevent it?

- Use **prepared statements** or **parameterized queries** to prevent SQL manipulation.
- Validate and sanitize all user inputs (e.g., remove special characters).
- Implement **least privilege access** to limit database permissions.

🔹 Insecure Direct Object References (IDOR)

What is it?

- IDOR occurs when an attacker is able to access or modify objects (files, records, etc.) they shouldn't be authorized to access by manipulating input parameters (e.g., URL or POST data).

How to find it?

- Modify URL parameters (e.g., ?id=1, ?file=doc1) and check if unauthorized objects are accessible:

  https://example.com/profile?id=2

  https://example.com/file?filename=secretfile.pdf

- Test for unprotected resources that don't check user authorization.

How to prevent it?

- Always check if the user is authorized to access the resource by validating their session and role.
- Use **access control mechanisms** and avoid exposing sensitive data in URLs.
- Implement **randomized or obfuscated** identifiers instead of sequential numbers (e.g., ?id=xyz123).