1. SQL Injection (SQLi)

Goal: Log in as the Administrator without knowing their password.

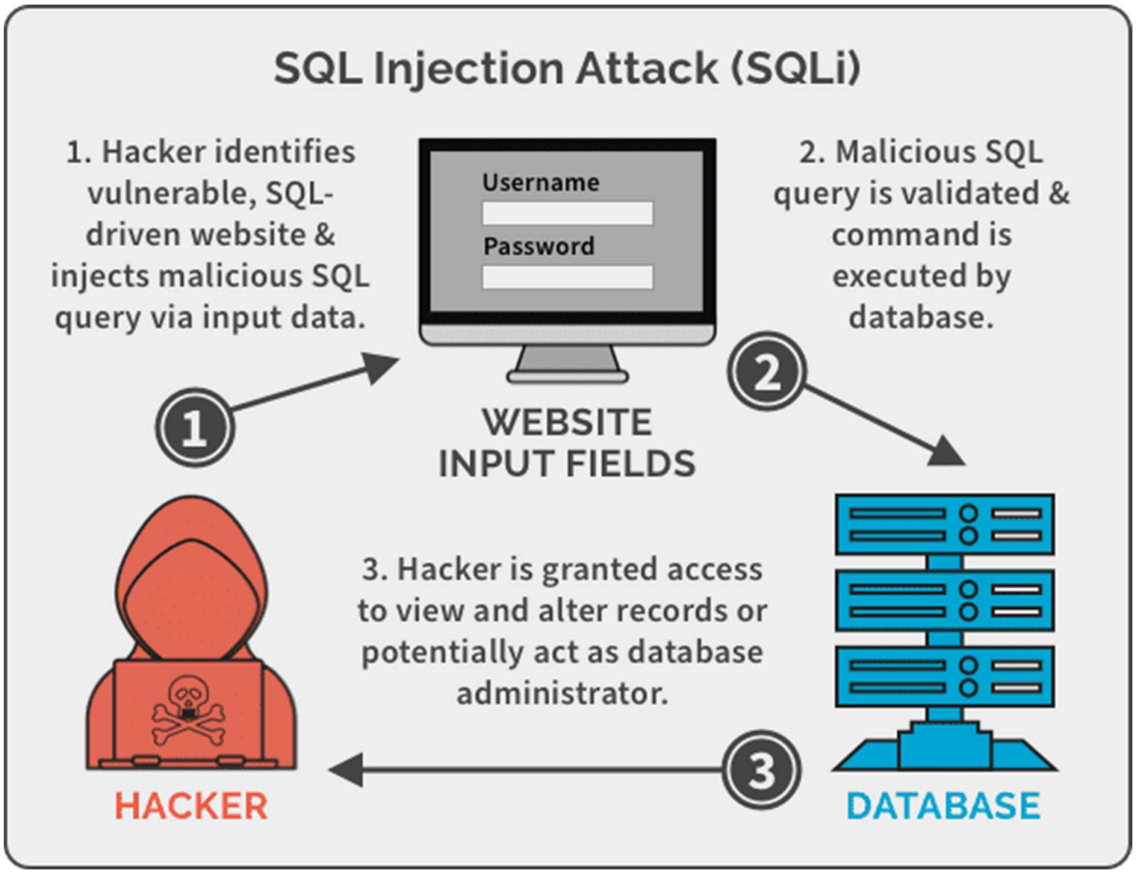
Step-by-Step:

1. **Navigate to Login:** Go to the login page of the application.
2. **Enter the Payload:** In the Email field, type: ' OR 1=1 --
3. **Password:** Enter anything (e.g., 123).
4. **How it works:** * The single quote ' closes the email string in the database query.
    - OR 1=1 is a logic statement that is **always true**.
    - -- is a comment in SQL, which tells the database to ignore the rest of the query (like the password check).
5. **Result:** The server sees the "True" statement and logs you in as the first user in the database—usually the **Admin**.

Visual Concept:

Input: ' OR 1=1 --

Backend Query becomes: SELECT * FROM Users WHERE email = '' OR 1=1 --' AND password = '...'

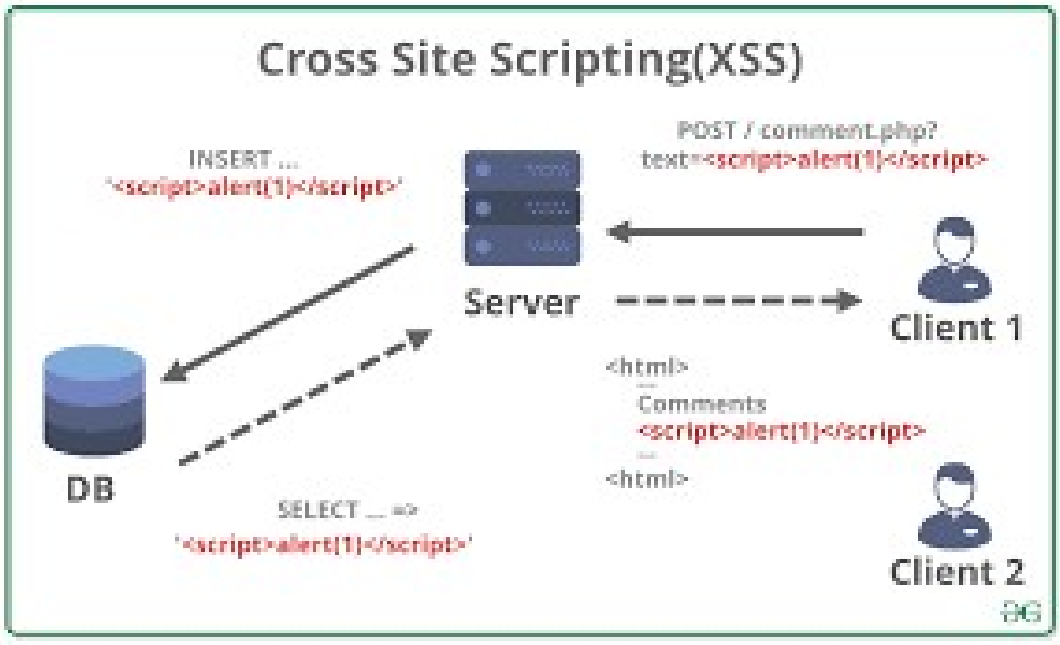


2. Cross-Site Scripting (XSS)

Goal: Execute a script in the browser to prove the site doesn't "sanitize" user input.

Step-by-Step:

1. **Find a Search Bar:** Use the search field at the top of the page.
2. **Enter the Payload:** Type the following: <iframe src="javascript:alert('XSS')">
3. **Submit:** Press enter.
4. **Result:** If a pop-up alert box appears saying "XSS," the vulnerability is confirmed.
5. **Risk:** An attacker could replace alert('XSS') with a script that steals your document.cookie and sends it to their server.

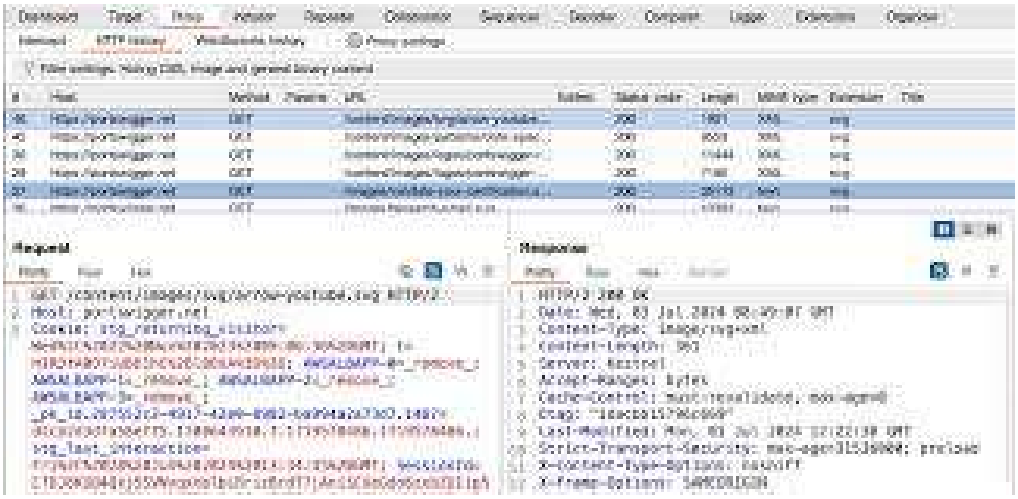


3. Intercepting & Modifying (Burp Suite)

Goal: Change the price of an item before the order reaches the server.

Step-by-Step:

1. **Setup:** Open Burp Suite, go to the **Proxy** tab, and click **Open Browser**.
2. **Intercept:** In Burp, toggle **Intercept is ON**.
3. **The Action:** In the browser, go to an item and click **Add to Basket**.
4. **The Capture:** The browser will "hang." Switch to Burp Suite. You will see the raw HTTP request.
5. **The Modification:**
    - Look for a parameter like price: 1000 or quantity: 1.
    - Change the price value to 1 or even a negative number like -500.
6. **Forward:** Click the **Forward** button in Burp.
7. **Result:** Check your cart in the browser. You will see the item added at the modified price you set in Burp.



Summary of Findings (Reporting)

Vulnerability	Attack Vector	Mitigation
SQL Injection	Login Form	Use **Parameterized Queries** (Prepared Statements).
XSS	Search Bar	Implement **Output Encoding** and Content Security Policy (CSP).
Price Tampering	Intercepted POST Request	Perform all validation (price/stock) on the **Server-side**.