# Major Security Vulnerabilities & Countermeasures

## 1. SQL Injection (SQLi)

* **Vulnerability:** Attackers manipulate **SQL queries via input fields**, exposing or modifying database data.
* **Countermeasures:**
  + Use **prepared statements** (parameterized queries).
  + **Sanitize and validate** user inputs.
  + **Limit database permissions** (e.g., avoid using root accounts).

## 2. Cross-Site Scripting (XSS)

* **Vulnerability:** Injects **malicious** scripts into webpages, affecting users’ **browsers**.
* **Countermeasures:**
  + Escape and **sanitize user inputs**.
  + Implement **Content Security Policy (CSP)**.
  + Use **secure frameworks** that auto-escape inputs.

## 3. Cross-Site Request Forgery (CSRF)

* **Vulnerability:** Forces **authenticated** users to perform unwanted actions on
* **Countermeasures:**
  + Use **CSRF tokens** in forms and API requests.
  + Enforce **SameSite cookie attributes**.
  + Require **user re-authentication** for sensitive actions.

## 4. Broken Authentication

* **Vulnerability:** **Weak authentication** mechanisms allow attackers to hijack accounts.
* **Countermeasures:**
  + Enforce **strong password policies** (length, complexity).
  + Implement **multi-factor authentication (MFA)**.
  + Use **secure session management** (e.g., secure and HttpOnly cookies).

## 7. Unvalidated Redirects & Forwards

* **Vulnerability:** Attackers exploit redirects to send users to malicious sites.
* **Countermeasures:**
  + Validate **redirect URLs** and use an allowlist.
  + Avoid using user-controllable redirect parameters.

## 9. Server-Side Request Forgery (SSRF)

* **Vulnerability:** Attackers **force servers to make unauthorized requests** to internal/external services.
* **Countermeasures:**
  + Block **internal network access** from untrusted inputs.
  + Use **allowlists** for outgoing requests.
  + Validate and sanitize all **user-supplied URLs**.

## 10. Insecure APIs

* **Vulnerability:** **Poorly secured APIs** expose sensitive data to unauthorized access.
* **Countermeasures:**
  + Use **API authentication (OAuth, JWT, API keys)**.
  + Implement **rate limiting** to prevent abuse.
  + Use **encrypted connections (TLS/SSL)**.