**Missing Security Headers** – Security headers protect against common attacks like **XSS, Clickjacking, and MITM**.

* **Risk**: Without them, attackers can exploit vulnerabilities in browser security.
* **How to Avoid**: Use X-XSS-Protection, Content-Security-Policy (CSP), and X-Frame-Options.

**Unsecured Forms** – Forms without security measures are vulnerable to **SQL Injection (SQLi), XSS, and CSRF attacks**.

* **Risk**: Attackers can inject harmful scripts or manipulate requests.
* **How to Avoid**: Validate inputs, use CSRF tokens, and enforce HTTPS.

**SQL Injection (SQLi)** – Attackers insert malicious SQL queries to **manipulate database records**.

* **Risk**: Can lead to **data theft, modification, or database deletion**.
* **How to Avoid**: Use **prepared statements and parameterized queries**.

**Cross-Site Scripting (XSS)** – Injecting malicious JavaScript into webpages **to steal data or hijack sessions**.

* **Risk**: Can **compromise user accounts or modify site content**.
* **How to Avoid**: Sanitize input, use CSP, and set HttpOnly cookies.

**Clickjacking** – A hidden iframe tricks users into performing **unintended actions**.

* **Risk**: Attackers can **steal credentials or perform unauthorized actions**.
* **How to Avoid**: Use X-Frame-Options: DENY or frame-ancestors in CSP.

**Insecure Cookies** – Cookies without security attributes can be **stolen or manipulated**.

* **Risk**: Attackers can **hijack user sessions and impersonate users**.
* **How to Avoid**: Set Secure, HttpOnly, and SameSite=Strict attributes.

**Open Redirects** – The website allows redirects to **untrusted external URLs**.

* **Risk**: Attackers can **redirect users to phishing or malware sites**.
* **How to Avoid**: Restrict redirections to a **trusted list of URLs**.

**Directory Listing Enabled** – The server exposes **all files and directories** to the public.

* **Risk**: Attackers can **access sensitive files like backups or config files**.
* **How to Avoid**: Disable directory listing using .htaccess (Options -Indexes).

**Using HTTP Instead of HTTPS** – Data is transferred **unencrypted**, making it vulnerable to interception.

* **Risk**: Hackers can **steal or modify sensitive information** through MITM attacks.
* **How to Avoid**: Install **SSL/TLS**, enforce **HTTPS**, and enable **HSTS**.