3. REQUIREMENT ANALYSIS

3.1 List of Vulnerabilities

- SQL Injection : Exploited input fields to manipulate database queries and bypass authentication.
- Cross-Site Scripting : Found in input fields where JavaScript execution was possible.
- Cross-Site Request Forgery : Identified vulnerability allowing unauthorized actions using forged requests.
- Intrusion Detection using Snort : Used Snort to monitor network traffic and detect malicious activity.
- Security Misconfiguration - Discovered due to missing security headers and exposed admin configurations.

3.2 Solution Requirement

(Vulnerability assessment details)

1. Cross site scripting (XSS):
    - Input sanitization and validation were implemented to prevent malicious script execution.
    - Content Security Policy (CSP) was configured to restrict unauthorized script execution.
    - HttpOnly and Secure flags were applied to cookies to prevent session hijacking.

2. Security Misconfigurations:
    - Security headers (X-Frame-Options, X-Content-Type-Options) were added.
    - Directory listing was disabled to prevent unauthorized access to sensitive files.
    - Default credentials were removed and strong authentication policies were enforced.

3. SQL Injection :
    - To detect the vulnerability
    - To execute solution

4. Intrusion Detection System using Snort:
    - To detect the incoming traffic
    - To see where the traffic is coming from
    - To check the danger level of the attacks

5. Cross-Site Request Forgery:
    - CSRF tokens were implemented to validate legitimate user requests and prevent unauthorized actions.
    - SameSite cookie attribute was set to `Strict` or `Lax` to prevent cookies from being sent in cross-origin requests.
    - Referer and Origin header validation was enforced to ensure requests originate from trusted sources.

3.3 Technology Stack

- Burp Suite
- Snort
- Virtual Machines
- Languages used : JavaScript , Python