6.FUNCTIONAL AND PERFORMANCE TESTING

6.1 Vulnerability Report

1) **Intrusion detection system using snort**

    A) Identified Vulnerabilities and Assessment

    - ICMP Ping Flood → Severity: Medium
    - TCP SYN Flood → Severity : High
    - Open privileged ports → Severity : High

    B) Impact Analysis

    - The ICMP flood and TCP SYN flood attacks can significantly affect the system's **availability**, making it unresponsive to legitimate users. Continuous attacks could result in **resource exhaustion**, where CPU and memory are overloaded, leading to performance degradation or a complete system failure.
    - If privileged ports are left open without proper security controls, attackers could attempt to **exploit misconfigured services**, leading to **unauthorized access**. Additionally, without active **firewall filtering and automated countermeasures**, the system remains exposed to **repeated attacks** without effective mitigation.

2) **SQL Injection**

A) Identified Vulnerabilities and Assessment

- **Vulnerability Type:** SQL Injection (Authentication Bypass)
- **Affected Component:** Login Authentication System
- **Attack Vector:** Input fields for Username and Password
- **Payload Used:** `admin' OR '1'='1' --`

B) Impact Analysis

- The login form of the application does not properly sanitize user inputs before embedding them into SQL queries. This allows an attacker to manipulate the SQL query logic to always return true, bypassing authentication mechanisms.
- Due to the ease of exploitation and the severity of the impact, this vulnerability poses a high security risk.

3) Cross-Site Scripting :

Identified Vulnerabilities and Assessment

- Vulnerability Type: Cross-Site Scripting (XSS)

- Affected Component: User Input Fields (Search, Comment Section)
- Attack Vector: Injection of Malicious JavaScript Code
- Payload Used: <script>alert('XSS Attack')</script>

Impact Analysis:

- The web application does not properly validate or sanitize user inputs before rendering them in the browser. This allows an attacker to inject and execute arbitrary JavaScript code on the client-side.
- Attackers can steal session cookies, leading to account hijacking.
- Defacement or misleading content injection, tricking users into entering sensitive information.
- If combined with phishing techniques, can be used for credential theft.
- Persistent XSS can store malicious scripts in the database, affecting multiple users.
- Lack of input sanitization and output encoding leaves the system vulnerable to both stored and reflected XSS attacks.

4) Security Misconfigurations:

A) Identified Vulnerabilities and Assessment
- Vulnerability Type: Security Misconfiguration
- Affected Component: Server and Application Settings
- Attack Vector: Default Credentials, Improper Security Headers, Exposed Configuration Files
- Observed Issues:
- Exposed .env files, revealing database credentials and API keys.
- Missing HTTP Security Headers, such as X-Frame-Options and Content-Security-Policy.
- Default Admin Credentials, making it easy for attackers to gain unauthorized access.

B) Impact Analysis :
- Misconfigured security settings expose sensitive data and create multiple attack vectors.
- Exposed .env files allow attackers to extract credentials, leading to unauthorized database access.
- Lack of security headers makes the application vulnerable to clickjacking, XSS, and data injection attacks.
- Using default credentials increases the risk of unauthorized access and privilege escalation.

- Attackers can exploit misconfigurations to bypass security controls and compromise the entire system.

5) Cross-Site Request Forgery :

A) Identified Vulnerabilities and Assessment

- Vulnerability Type: Cross-Site Request Forgery (CSRF)
- Affected Component: Money Transfer Functionality (`/transfer` endpoint)
- Attack Vector: Unauthorized execution of sensitive actions via a forged request
- Payload Used:

```
 <form action="http://127.0.0.1:5000/transfer" method="POST">

<input type="hidden" name="recipient" value="attacker">

 <input type="hidden" name="amount" value="500">

<input type="submit" value="Claim Your Reward">

</form>
```

Impact Analysis :

- Lack of CSRF protection allows attackers to trick authenticated users into performing unauthorized money transfers without their knowledge.
- Session cookies are automatically included in cross-site requests, allowing the attack to execute without requiring user credentials.
- Attackers can embed malicious forms in phishing emails or external websites, exploiting user trust to steal funds.
- No CSRF token implementation, making it possible for attackers to replay forged requests repeatedly.
- If an admin panel lacks CSRF protection, attackers can manipulate account settings, change user roles, or delete accounts.