Université Euro Méditerranéenne Fès

EuroMed University of Fez

Ecole d'Ingénierie Digitale et d'Intelligence Artificielle (EIDIA)

Field of Study: 2nd Year Cyber Security

Semestre: 3

Cours: Smartphone and web security

Repport

Identified Vulnerabilities and Issues

Supervised by:

Pr. TMIMI mehdi

Prepared by:

ECHCHOURA Mohammed Amine

Année Universitaire : 2025/2024

Identified Vulnerabilities and Issues

1. Plaintext Password Storage

- **Issue**: Passwords are stored directly in the database without encryption.
- **Impact**: If the database is compromised, attackers gain access to all user passwords in plaintext.
- **Solution** :
  - Use a hashing algorithm such as « bcrypt » to hash passwords before storing them.
  - Verify passwords by comparing the hashed version during login.

2. Lack of Input Validation

- **Issue**: User inputs (e.g., username and password) are not validated or sanitized.
- **Impact**: This allows potential attacks like SQL Injection, Cross-Site Scripting (XSS), or other malicious payloads.
- **Solution** :
  - Use input validation libraries such as validator or express-validator.
  - Sanitize inputs to remove any harmful characters or code.

3. Unencrypted Session Storage

- **Issue**: The session key (mysecretkey) is hardcoded and weak.
- **Impact**: If the session key is exposed, attackers can forge session tokens.
- **Solution** :
  - Store secrets and sensitive information in environment variables using tools like dotenv.
  - Use a strong, randomly generated session secret.

4. Lack of Secure Cookie Flags

- **Issue**: Cookies do not have the Secure and HttpOnly flags set.
- **Impact**: Cookies can be intercepted or accessed via client-side JavaScript, leading to session hijacking.
- **Solution** :
  - Configure cookies with the Secure, HttpOnly, and SameSite flags to protect against theft and unauthorized access.

5. Fixing Lack of Rate Limiting:

- **Issue**: No protection against brute-force attacks on endpoints like /login and /register.
- **Impact**: Attackers can repeatedly attempt login or registration, leading to account compromise or denial-of-service (DoS).
- **Solution:**
  - Implement rate-limiting middleware, such as express-rate-limit, to limit repeated requests.

6. Missing HTTPS Enforcement

- **Issue**: The application does not enforce HTTPS connections.
- **Impact**: Data, including credentials, could be intercepted in transit (man-in-the-middle attacks).
- **Solution**:
    - Use HTTPS for all communication.
    - Enforce secure headers with libraries like helmet

7. Exposure of Sensitive Information

Issue: Errors are logged with full stack traces and internal details, which may expose database structure, file paths, or other sensitive information.

```
console.error('Error reading data:', err); => Exposes full error details
```

Solution: Log only the error message without exposing stack traces:

```
console.error(`Error: ${err.message}`);
```