Task Findings

Abdullah Akram
ID: DHC-3606

July 8, 2025

1 Vulnerabilities Found

The ZAP by Checkmarx Scanning Report, generated on July 8, 2025, identified several vulnerabilities in the scanned site (`http://localhost`). Below is a detailed list of the vulnerabilities categorized by risk level:

- **High Risk (2 Alerts, Medium Confidence)**:
  - *Cross Site Scripting (Reflected)*: Detected in a POST request to `http://localhost/h` This vulnerability (CWE-79, WASC-8) could allow attackers to inject malicious scripts into web pages viewed by users, potentially stealing sensitive data or performing unauthorized actions.
  - *SQL Injection - MySQL*: Identified in a POST request to `http://localhost/home-re` This vulnerability (CWE-89, WASC-19) could enable attackers to manipulate database queries, potentially accessing or modifying sensitive data.
- **Medium Risk (5 Alerts, Varying Confidence)**:
  - *Content Security Policy (CSP) Header Not Set* (High Confidence): Found in a GET request to `http://localhost/sitemap.xml`. The absence of a CSP header (CWE-693, WASC-15) increases the risk of XSS attacks by not restricting resource loading.
  - *Application Error Disclosure* (Medium Confidence): Detected in a GET request to `http://localhost/home-rental/app/register.php`. This issue (CWE-550, WASC-13) may expose sensitive system information, aiding attackers in exploiting other vulnerabilities.
  - *Missing Anti-clickjacking Header* (Medium Confidence): Identified in a GET request to `http://localhost/home-rental/`. Lack of X-Frame-Options header (CWE-1021, WASC-15) makes the site vulnerable to clickjacking attacks.
  - *Vulnerable JS Library* (Medium Confidence): Found in a GET request to `http://localhost/home-rental/assets/plugins/bootstrap/js/bootstra` Outdated or vulnerable JavaScript libraries (CWE-1395) can be exploited to compromise the application.

  - *Absence of Anti-CSRF Tokens* (Low Confidence): Detected in a GET request to `http://localhost/home-rental/auth/register.php`. Missing CSRF tokens (CWE-352, WASC-9) could allow attackers to perform unauthorized actions on behalf of authenticated users.

- **Low Risk (6 Alerts, Varying Confidence)**:
  - *Server Leaks Version Information* (High Confidence): Found in a GET request to `http://localhost/home-rental/assets/css/rent.css`. Exposing server version details (CWE-497, WASC-13) can help attackers identify exploitable vulnerabilities.
  - *Big Redirect Detected* (Medium Confidence): Identified in a GET request to `http://localhost/home-rental/app/register.php`. Large redirects (CWE-201, WASC-13) may leak sensitive information.
  - *Cookie No HttpOnly Flag* (Medium Confidence): Detected in a GET request to `http://localhost/home-rental/`. Cookies without the HttpOnly flag (CWE-1004, WASC-13) are accessible to client-side scripts, increasing XSS risks.
  - *Cookie without SameSite Attribute* (Medium Confidence): Found in a GET request to `http://localhost/home-rental/`. Missing SameSite attributes (CWE-1275, WASC-13) can expose cookies to CSRF attacks.
  - *Server Leaks Information via "X-Powered-By"* (Medium Confidence): Identified in a GET request to `http://localhost/home-rental/auth/register.php` Exposing framework details (CWE-497, WASC-13) can aid attackers.
  - *X-Content-Type-Options Header Missing* (Medium Confidence): Detected in a GET request to `http://localhost/home-rental/assets/css/style.css`. Absence of this header (CWE-693, WASC-15) may allow MIME-type sniffing attacks.

- **Informational (6 Alerts, Varying Confidence)**:
  - *Authentication Request Identified* (High Confidence): Found in a POST request to `http://localhost/home-rental/auth/login.php`. This indicates an authentication endpoint that may require further security review.
  - *Information Disclosure - Suspicious Comments* (Medium Confidence): Identified in the site, potentially exposing sensitive information in code comments (CWE-615, WASC-13).
  - *Modern Web Application* (Medium Confidence): Indicates the use of modern web technologies, which may have specific security considerations.
  - *Session Management Response Identified* (Medium Confidence): Detected in the site, suggesting session management mechanisms that need secure configuration.
  - *User Controllable HTML Element Attribute* (Medium Confidence): Found in the site, indicating potential XSS risks due to user-controlled attributes (CWE-20, WASC-20).

2 Areas of Improvement

Based on the vulnerabilities identified, the following areas require immediate attention to enhance the security of the `http://localhost` site:

- **Implement Input Validation and Sanitization**:
  - Address Cross Site Scripting (Reflected) and SQL Injection vulnerabilities by implementing robust input validation and parameterized queries. Use libraries like OWASP ESAPI for input sanitization and prepared statements for database queries.
  - Reference: OWASP XSS and OWASP SQL Injection Cheat Sheet.
- **Enhance HTTP Security Headers**:
  - Implement Content Security Policy (CSP) to mitigate XSS risks by restricting resource loading. Configure `X-Frame-Options` to prevent clickjacking and `X-Content-Type-Options` to prevent MIME-type sniffing.
  - Set `HttpOnly` and `SameSite` attributes on cookies to protect against XSS and CSRF attacks.
  - Reference: MDN CSP and OWASP Security Headers.
- **Update and Monitor Dependencies**:
  - Replace or update vulnerable JavaScript libraries (e.g., Bootstrap) to their latest secure versions. Use tools like Retire.js to monitor for outdated dependencies.
  - Reference: OWASP Top 10: Vulnerable Components.
- **Implement CSRF Protection**:
  - Add anti-CSRF tokens to all state-changing requests to prevent unauthorized actions. Use frameworks that provide built-in CSRF protection, such as Laravel or Django.
  - Reference: OWASP CSRF Cheat Sheet.
- **Reduce Information Disclosure**:
  - Configure the server to suppress version information in `Server` and `X-Powered-By` headers. Disable detailed error messages to prevent application error disclosure.
  - Remove or sanitize suspicious comments in source code to avoid leaking sensitive information.
  - Reference: Troy Hunt: Response Headers.
- **Strengthen Session and Authentication Mechanisms**:

- Securely configure session management to prevent session hijacking. Review authentication endpoints to ensure they use secure protocols (e.g., HTTPS) and strong password policies.
- Reference: ZAP Authentication Helper.