**Web Application Security Testing Report**

**1. Introduction** This report documents the results of a web application security assessment conducted using OWASP ZAP. The purpose of this assessment is to identify common web vulnerabilities and evaluate the security posture of the target application following OWASP guidelines. The testing was conducted within the defined scope and boundaries.

---

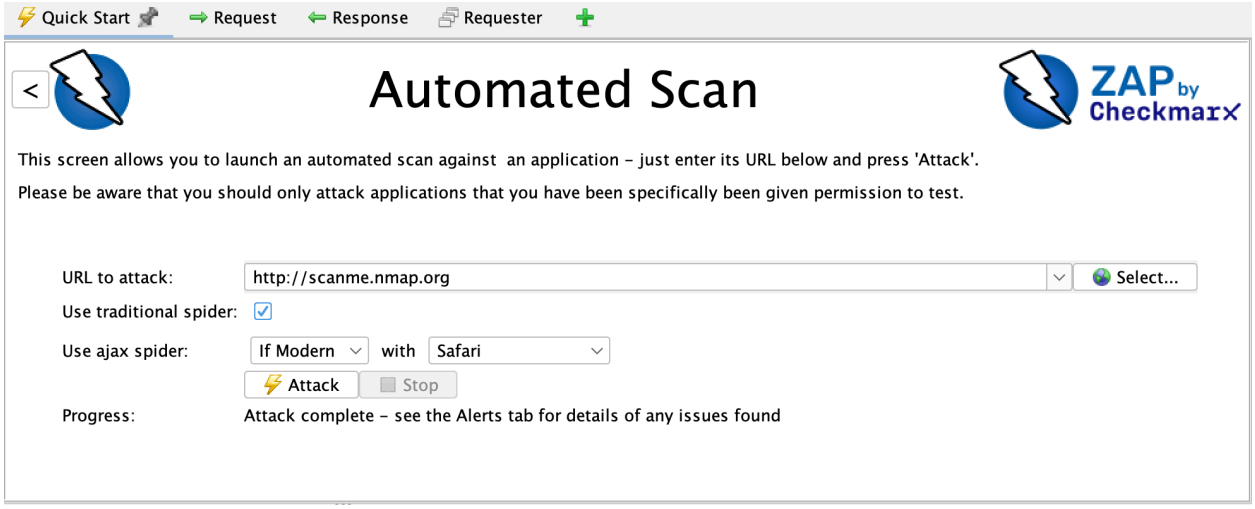
## 2. Testing Methodology

### 2.1 Scope Definition

- Target URL: `http://scanme.nmap.org`
- Allowed Tests: Passive scanning, active scanning (with permission)
- Exclusions: Authentication-based testing, sensitive transactions

### 2.2 Tool Configuration

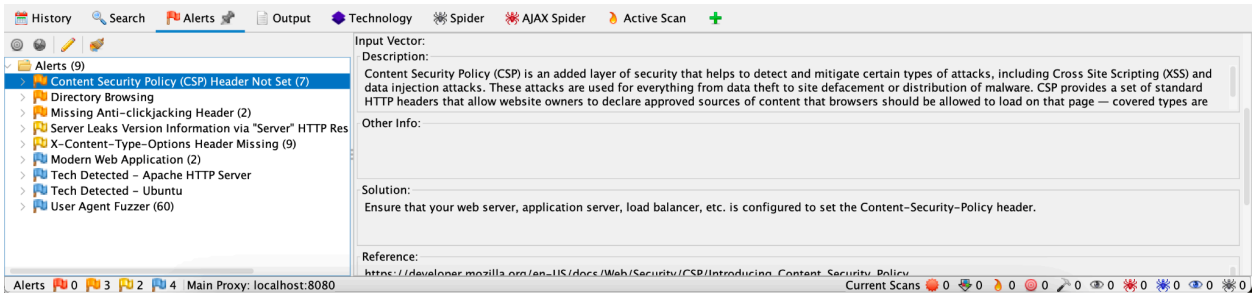
- **Tool Used:** OWASP ZAP
- **Traditional Spider:** Enabled
- **AJAX Spider:** Enabled (Modern browsers: Safari)
- **Automated Scan:** Initiated with default ZAP settings

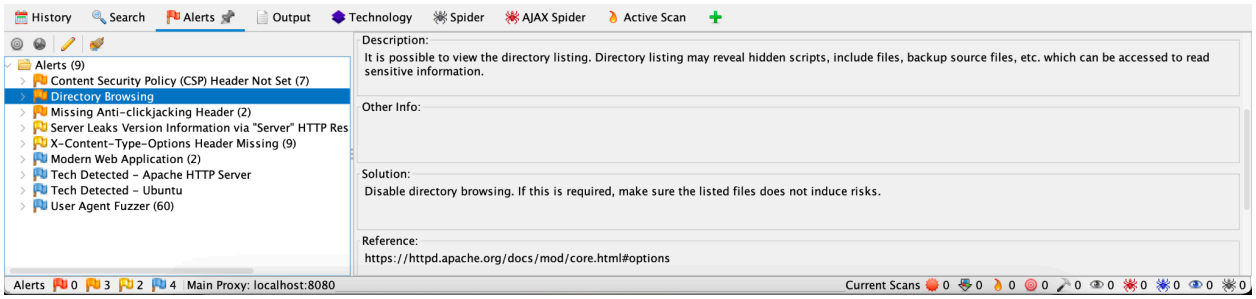



---

## 3. Identified Vulnerabilities and OWASP Mapping

| Vulnerability | OWASP Category | Risk Level | Description & Impact |
|---|---|---|---|
| Content Security Policy (CSP) Header Not Set | A6: Security Misconfiguration | Medium | Missing CSP allows XSS and data injection attacks. |
| Directory Browsing Enabled | A5: Security Misconfiguration | Medium | Unauthorized users can access sensitive files and scripts. |

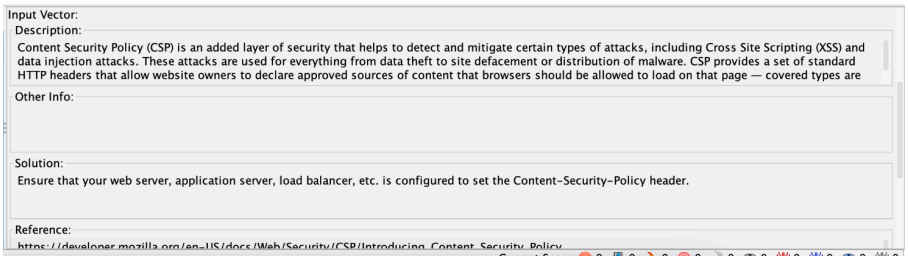


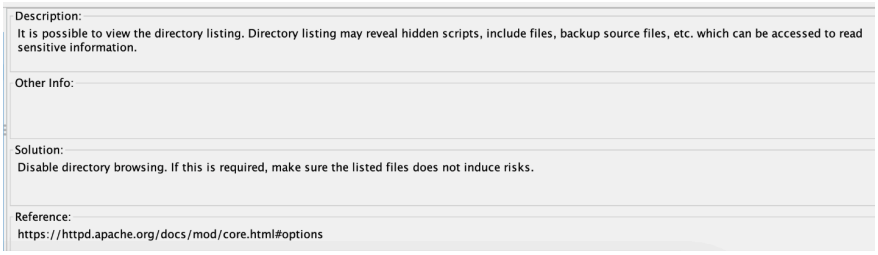




---

## 4. Evidence of Findings

1. **Content Security Policy (CSP) Header Not Set**
   - **Description:** CSP helps protect against XSS and injection attacks.
   - **Impact:** Increases risk of cross-site scripting (XSS).
   - **Evidence:** Missing CSP detected in HTTP headers.

- 
Input Vector:
Description:
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are
Other Info:
Solution:
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
Reference:
https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy

2. **Directory Browsing Enabled**
   - **Description:** Allows attackers to view internal files.
   - **Impact:** Exposure of sensitive files.
   - **Evidence:** Directory listing was accessible via a browser.
   - 
Description:
It is possible to view the directory listing. Directory listing may reveal hidden scripts, include files, backup source files, etc. which can be accessed to read sensitive information.
Other Info:
Solution:
Disable directory browsing. If this is required, make sure the listed files does not induce risks.
Reference:
https://httpd.apache.org/docs/mod/core.html#options

---

## 5. Remediation Recommendations

1. **Content Security Policy (CSP) Header Not Set**
   - **Solution:** Configure the web server to implement a strict CSP header.
   - **Reference:** CSP Guide
2. **Directory Browsing Enabled**
   - **Solution:** Disable directory listing in the server configuration.
   - **Reference:** Apache Directory Listing Guide

---

**6. Conclusion** This assessment revealed security misconfigurations that could be exploited by attackers. Implementing the recommended mitigations will enhance the security of the web application by reducing exposure to common vulnerabilities.

**Next Steps:**

- Apply the suggested security headers.
- Restrict unnecessary file access.
- Perform a follow-up scan after remediation.