

FUTURE INTERNS

2026

**VULNERABILITY
ASSESSMENT
REPORT**

**PREPARED BY
GIRIDHARAN S**

Target: demo.testfire.net
Prepared by: Giridharan.S
Date: January 2026



Executive Summary

A vulnerability assessment was conducted on demo.testfire.net using Nmap and OWASP ZAP. The objective was to identify exposed services, insecure configurations, and web application flaws. The findings highlight multiple risks ranging from outdated server software to missing security headers. This report provides clear remediation steps in business-friendly language.

Identified Vulnerabilities

1. Open Ports & Services

- 80/tcp (HTTP) → Apache Tomcat/Coyote JSP 1.1
- 443/tcp (HTTPS) → Apache Tomcat/Coyote JSP 1.1
- 8080/tcp (HTTP) → Apache Tomcat/Coyote JSP 1.1 Risk: High (outdated Tomcat version known for exploits)

2. Missing Security Headers

- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Missing Anti-clickjacking Header
- Cookie without SameSite Attribute
- X-Content-Type-Options Header Missing

3. Information Disclosure



- Debug error messages visible
- Server leaks version info via HTTP headers
- Suspicious comments in source code Risk: Medium (provides attackers with system details)

4. Session Management Issues

- Multiple GET/POST requests with session identifiers
- Weak session handling detected (cookies without secure attributes) Risk: High (session hijacking possible)

5. Site Map Analysis

- Exposed endpoints: /feedback.jsp, /sendFeedback, /swagger, /util
- Swagger endpoints may expose API documentation and testing interfaces Risk: High (API endpoints can be exploited if not secured)

Risk Classification

- High: Outdated Apache Tomcat, exposed port 8080, insecure session management, exposed API endpoints
- Medium: Missing security headers, information disclosure
- Low: Minor comments in source code

Remediation Steps



- Restrict or disable port 8080 if not required
- Update Apache Tomcat to the latest patched version
- Implement security headers (CSP, Anti-CSRF, Anti-clickjacking, SameSite cookies)
- Remove debug error messages and suspicious comments from code
- Harden session management (secure cookies, regenerate session IDs)
- Secure API endpoints and restrict Swagger access to developers only

Conclusion

The assessment revealed multiple vulnerabilities that could be exploited to compromise the application. Immediate remediation of high-risk issues (Tomcat upgrade, session hardening, API restrictions) is recommended. Addressing medium-risk issues (headers, information disclosure) will further strengthen the security posture.

SCREENSHOTS:



