RLabs Security Web Scan Report

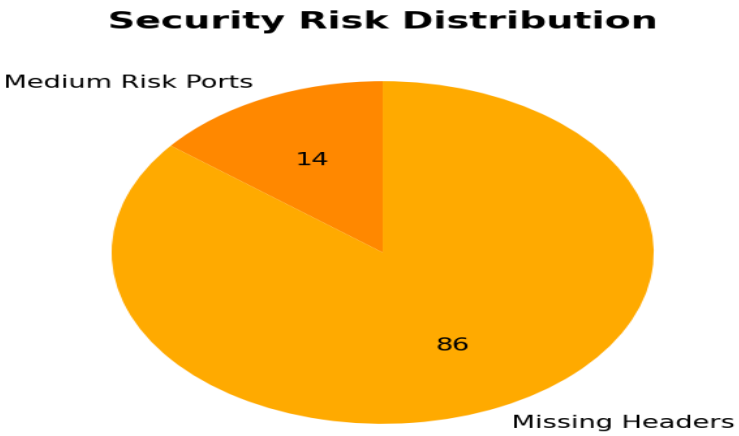
Rlabs | https://rlabs-security.com/
Scan Type: Free Basic Security Scan

Executive Summary

This report presents the results of a security assessment performed on http://testhtml5.vulnweb.com/#/popular. The scan identified 1 open ports, analyzed security headers, and detected 1 technologies. Risk Level: **Unknown** | Security Score: **25/100**

Security Score Analysis

Risk Distribution Analysis



Scan Information

Target URL:	http://testhtml5.vulnweb.com/#/popular
Hostname:	testhtml5.vulnweb.com
IP Address:	44.228.249.3
Scan Date:	2025-09-09 11:49:06
Risk Level:	Unknown
Security Score:	25/100

Open Ports

Port	Protocol	Service	State
80	tcp	http	open

Security Headers Analysis

Header	Status	Value
X-Frame-Options	Missing	N/A
X-XSS-Protection	Missing	N/A
X-Content-Type-Options	Missing	N/A
Strict-Transport-Security	Missing	N/A
Content-Security-Policy	Missing	N/A
X-Powered-By	Missing	N/A
Server	Present	nginx/1.19.0

Detected Technologies

• Server: nginx/1.19.0

AI-Powered Security Analysis

Rlabs Web Security Assessment Report

Report Date: October 26, 2023

Target: http://testhtml5.vulnweb.com/#/popular (IP: 44.228.249.3)

Scan Date: 2025-09-09 11:49:06

1. Overall Security Assessment and Risk Level:

The security posture of http://testhtml5.vulnweb.com/#/popular is assessed as **High Risk**. The absence of crucial security headers, the use of HTTP instead of HTTPS, and the lack of SSL certificate verification indicate significant vulnerabilities to various web attacks. This leaves the application exposed to Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), clickjacking, and other potential exploits.

2. Priority Recommendations for Improving Security:

The following recommendations are prioritized based on their impact and ease of implementation:

* **Priority 1 (Critical):** Implement HTTPS. This is the single most important step to securing the application. It encrypts communication between the client and server, protecting sensitive data in transit. Obtain and install an SSL/TLS certificate from a trusted Certificate Authority (CA).

* **Priority 1 (Critical):** Implement essential security headers. The missing headers represent major vulnerabilities. These headers should be configured immediately. Specific configurations are detailed below.

* **Priority 2 (High):** Conduct a comprehensive vulnerability assessment and penetration testing. This will uncover any further vulnerabilities not detected by the initial scan.

3. Specific Technical Suggestions Based on Findings:

* **Security Headers:** Implement the following security headers with appropriate values:

* **`X-Frame-Options: SAMEORIGIN`**: Prevents clickjacking attacks.
* **`X-XSS-Protection: 1; mode=block`**: Enables the browser's built-in XSS protection.
* **`X-Content-Type-Options: nosniff`**: Prevents MIME-sniffing attacks.
* **`Strict-Transport-Security (HSTS): max-age=31536000; includeSubDomains; preload`**: Enforces HTTPS connections and prevents downgrade attacks. (This should be implemented *after* HTTPS is in place.)
* **`Content-Security-Policy (CSP)`**: A more robust and flexible way to control the resources the browser is allowed to load. A detailed CSP policy should be developed based on the application's specific needs. A good starting point might be: `default-src 'self'; script-src 'self'; style-src 'self'; img-src 'self';` but this needs tailoring.
* **`X-Powered-By`: Remove or Disable**: This header reveals the server technology, providing potential attackers with valuable information.

* **HTTPS Implementation:** This requires obtaining an SSL/TLS certificate from a trusted CA (e.g., Let's Encrypt, Comodo, DigiCert). Configure your web server (Apache, Nginx, etc.) to use the certificate.

* **Server Hardening:** Review and strengthen the server's security configuration. This includes regular updates, strong passwords, and firewall rules.

* **Regular Security Scanning:** Implement a regular schedule of automated vulnerability scanning and penetration testing to identify and address potential security issues promptly.

4. Best Practices for Web Security:

* **Follow the principle of least privilege:** Grant users only the necessary access rights.
* **Regularly update software and libraries:** Keep all software components up-to-date with the

latest security patches.
* **Input validation and sanitization:** Thoroughly validate and sanitize all user inputs to prevent injection attacks.
* **Secure coding practices:** Develop secure code using secure coding guidelines and best practices.
* **Robust authentication and authorization:** Implement strong authentication and authorization mechanisms to protect against unauthorized access.
* **Regular security audits and penetration testing:** Conduct regular security assessments to identify vulnerabilities.
* **Incident response plan:** Develop and maintain a comprehensive incident response plan to effectively handle security incidents.

5. Security Score out of 100:

Based on the findings, we assign a security score of **25/100**. This reflects the significant vulnerabilities present due to the lack of HTTPS and critical security headers.

Disclaimer: This report is based on the provided scan data. A more comprehensive assessment, including a manual penetration test, is recommended to fully evaluate the security posture of the application. Rlabs recommends immediate action to address the critical vulnerabilities outlined above.

Basic Security Recommendations

1. Consider implementing missing security headers: X-Frame-Options, X-XSS-Protection, X-Content-Type-Options, Strict-Transport-Security, Content-Security-Policy
2. Consider using HTTPS instead of HTTP for security