Vulnerability Assessment Report

Read-Only Security Assessment

Target website : https://uec.ac.in/
Project : 01
Date : February 2026
Assessment type : Passive security scan

Executive Summary

Every organization relies on its website to communicate with users and represent its brand.This vulnerability assessment was conducted on https://uec.ac.in to identify common security weaknesses that could impact security, reliability, and user trust.

The assessment followed a strictly non-intrusive, read-only approach, focusing on publicly accessible information such as HTTP headers, visible configurations, and exposed technologies. No exploitation, authentication testing, or aggressive scanning was performed.

The review identified a combination of low to medium risk security configuration issues, along with positive security practices already in place. Addressing the identified findings will help improve the website's overall security posture.

Scope of Assessment

Inscope

- **Public-facing pages**
- **HTTP response headers**
- **Passive configuration analysis**

Outscope

- **Login/authentication testing**
- **Exploitation or attacks**
- **Brute force / DoS**

Methodology

This vulnerability assessment was conducted using a manual, read-only approach to ensure ethical and non-intrusive testing.

The methodology focused on identifying common security misconfigurations and information disclosure issues typically reviewed during an initial security audit.

The following steps were followed:

- Reviewed publicly accessible pages of the website
- Inspected HTTP request and response headers using browser developer tools

- Identified exposed technologies, server configurations, and software versions
- Checked for the presence and effectiveness of common security headers
- Observed visible cookies and basic client-side security attributes
- Classified findings based on potential business impact and risk severity

No exploitation, authentication testing, or intrusive scanning techniques were performed.

Tools Used

The following tools were used during the assessment:

- Google Chrome (Developer Tools – Network & Headers)
- Manual HTTP header inspection
- Canva (professional report design)
- GitHub (documentation and evidence storage)

Only passive and non-intrusive tools were used to maintain ethical assessment boundaries.

TARGET OVERVIEW

- Website: https://uec.ac.in
- Website Type: Educational Institution
- Content Management System: WordPress (observed)
- Web Server: LiteSpeed
- Hosting Platform: Hostinger
- Backend Technology: PHP

This information was identified through publicly visible HTTP response headers.

Findings :

Finding 1: High Risk

Outdated PHP Version Disclosure

Description : The website discloses the backend PHP version through HTTP response headers.
Evidence : X-Powered-By: PHP/7.2.34

Why this matters : PHP 7.2 has reached end-of-life and no longer receives security updates. Outdated software versions are commonly targeted using known vulnerabilities.
Recommendation : Upgrade PHP to a currently supported version (PHP 8.x or later) and disable version disclosure in HTTP headers.

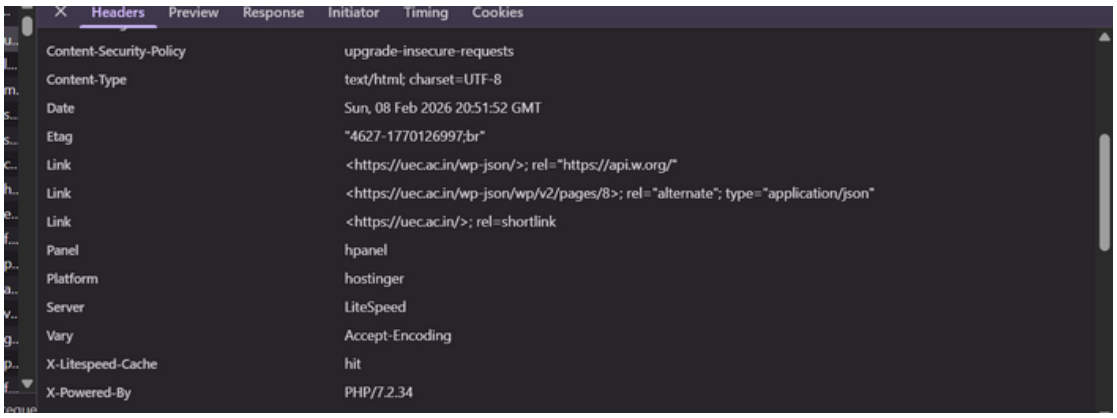
Finding 2: Medium Risk

Missing Standard Security Headers

Description : Several recommended HTTP security headers were not observed, including:

- X-Frame-Options
- X-Content-Type-Options
- Strict-Transport-Security

Evidence :



Why this matters : Missing security headers increase the risk of browser-based attacks such as clickjacking, MIME sniffing, and protocol downgrade attacks.

Recommendation : Enable standard HTTP security headers following OWASP security best practices.

Finding 3: Medium Risk

Incomplete Content Security Policy (CSP)

Description : A Content Security Policy is present but only enforces upgrade-insecure-requests without defining allowed resource sources.

Evidence :

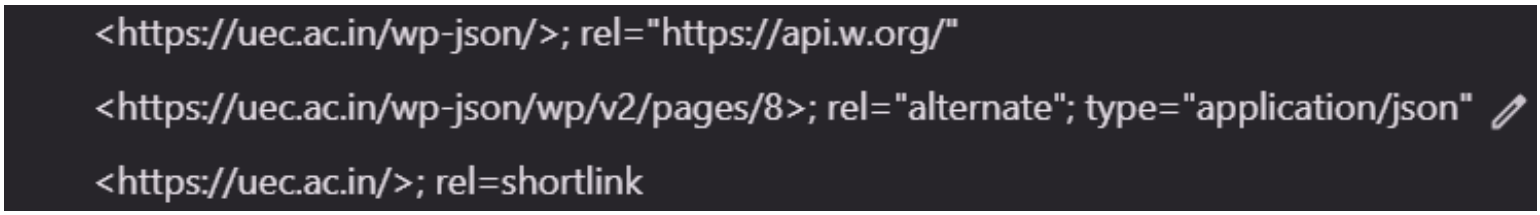
Why this matters : An incomplete CSP provides limited protection against content injection and script-based attacks.
Recommendation : Implement a comprehensive Content Security Policy defining trusted sources for scripts, styles, images, and frames.

Finding 4: Low Risk

WordPress REST API Exposure

Description : The WordPress REST API endpoint is publicly accessible.

Evidence :



Why this matters : While not a vulnerability by itself, exposed REST APIs may reveal site structure and metadata useful for reconnaissance.
Recommendation : Restrict REST API access where not required publicly and ensure no sensitive data is exposed.

Positive Security Observations

The following positive security practices were observed:

- HTTPS is properly enabled
- Secure TLS communication is enforced
- Modern compression techniques are in use
- HTTP/3 (QUIC) support observed

These measures contribute positively to overall website security.

Risk Summary

Risk Level	Number of Findings
High	1
Medium	2
Low	1

Conclusion & Disclaimer

Conclusion : This assessment identified common configuration and information disclosure issues that are fixable and preventable. Addressing the recommended actions will reduce potential security risks and improve the website’s overall security posture.

Disclaimer : This assessment was limited to publicly accessible information and passive analysis only.
No exploitation or intrusive testing was performed.