**Vulnerability Assessment Report**

**Application Name :** PACE eCap Portal

**Assessment Date :** January 29, 2025

**Assessor :** Chanukya Keerthi

**Tools Used :** OWASP ZAP, Burp Suite, Nmap

**1. Executive Summary**

A security assessment of the PACE eCap Portal (ecap.pace.ac.in) identified 13 vulnerabilities, including 3 Medium-risk, 6 Low-risk, and 4 Informational issues. Key findings include missing security headers (CSP, HSTS), potential Cross-Site Request Forgery (CSRF) risks, and server information leaks. While no critical vulnerabilities were detected, addressing Medium and Low risks is critical to mitigate potential attacks such as clickjacking, session hijacking, and data leakage.

**2. Methodology**

**2.1 Tools & Techniques**

**OWASP ZAP:** Automated passive/active scanning for vulnerabilities (e.g., XSS, CSRF, insecure headers).

**Burp Suite:** Manual validation of findings (not explicitly detailed in this report).

**Nmap:** Network enumeration to identify open ports/services (not explicitly detailed in this report).

**2.2 Scope**

**Targets:** https://ecap.pace.ac.in, http://ecap.pace.ac.in

Included Risks: Medium, Low, Informational.

**3. Detailed Findings**

**3.1 Medium-Risk Vulnerabilities**

**1. Content Security Policy (CSP) Header Not Set**

**Risk:** Medium | **Confidence:** High

**Affected URL:** http://ecap.pace.ac.in

**Description:** Missing CSP headers allow potential Cross-Site Scripting (XSS) attacks by not restricting sources of executable scripts.

**Impact:** Unauthorized script execution, data theft.

**2. Missing Anti-Clickjacking Header**

**Risk:** Medium | **Confidence:** Medium

**Affected URL:** http://ecap.pace.ac.in

**Description:** Missing X-Frame-Options header exposes the site to clickjacking attacks.

**Impact:** UI redress attacks, user interaction hijacking.

**3. Absence of Anti-CSRF Tokens**

**Risk:** Medium | **Confidence:** Low

**Affected URL:** https://ecap.pace.ac.in/default.aspx

**Description:** No CSRF tokens detected in the login form (POST request).

**Impact:** Unauthorized actions (e.g., password changes) via forged requests.

**3.2 Low-Risk Vulnerabilities**

**1. Strict-Transport-Security (HSTS) Header Not Set**

**Risk:** Low | **Confidence:** High

**Affected URL:** https://ecap.pace.ac.in/robots.txt

**Description:** Missing HSTS header weakens HTTPS enforcement, exposing users to SSL-stripping attacks.

**2. Server Leaks Version Information**

**Risk:** Low | **Confidence:** High

**Affected URLs:**

http://ecap.pace.ac.in/robots.txt (Server version via Server header).

https://ecap.pace.ac.in/ (X-AspNet-Version header exposed).

**Impact:** Attackers can exploit known vulnerabilities in the disclosed software.

**3. Insecure Cookie Configuration**

**Risk:** Low | **Confidence:** Medium

**Affected URL:** https://ecap.pace.ac.in/authcheck.aspx

**Description:** Cookies lack the Secure flag, allowing transmission over unencrypted HTTP.

**3.3 Informational Findings**

**1. User Controllable HTML Element Attribute (Potential XSS)**

**Risk:** Informational | **Confidence:** Low

**Affected URL:** https://ecap.pace.ac.in/default.aspx

**Description:** User input reflected in HTML attributes without sanitization.

**2. Re-examine Cache-Control Directives**

**Risk:** Informational | **Confidence:** Low

**Affected URL:** https://ecap.pace.ac.in/contactus.html

**Description:** Misconfigured caching may expose sensitive data.

**4. Recommendations**

**Immediate Actions (Medium Risks)**

**Implement CSP Headers:**

Add Content-Security-Policy to restrict script sources. Example:

**“Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted.cdn.com; “**

**Enable Anti-Clickjacking:**

**Add X-Frame-Options:** DENY or SAMEORIGIN to HTTP responses.

**Add Anti-CSRF Tokens:**

Use tokens in forms (e.g., \_\_RequestVerificationToken in ASP.NET).

**Prioritized Fixes (Low Risks)**

**Enforce HSTS:**

**Add Strict-Transport-Security:** max-age=31536000; includeSubDomains.

**Remove Server Version Headers:**

Suppress X-Powered-By, Server, and X-AspNet-Version headers.

**Secure Cookies:**

Set Secure and HttpOnly flags for session cookies.

**Best Practices (Informational):**

Sanitize user input in HTML attributes to prevent potential XSS.

Audit caching policies for sensitive pages (e.g., Cache-Control: no-store).

**5. Conclusion**

The assessment highlights the need for stronger security headers, input validation, and server hardening. Addressing these issues will reduce exposure to XSS, CSRF, and information leakage. Regular scans and penetration testing are recommended to maintain compliance with OWASP Top 10 standards.

**Appendix**

**Alert Types & References**

**Vulnerability Type CWE ID Reference**

Missing Anti-CSRF Tokens 352 OWASP CSRF Guide

CSP Header Not Set 693 MDN CSP Guide

Server Information Leak 200 OWASP Info Leak Guide