Clouds Dubai
Stay Secure !...

Case Study: Web Application Vulnerability Assessment and Penetration Testing (VAPT)

1. Overview

Client: Medical Company
Date of Test: October 6 – 18, 2024
Assessment Type: Grey Box Penetration Test

Tested Assets: 3 Web Applications

2. Objectives

The main goals of this assessment were:

- To identify vulnerabilities as per OWASP Top 10, SANS, NIST, and PTES standards.
- To assess the effectiveness of current security controls.
- To recommend actionable mitigations and improvements.

3. Methodology

Standards & Frameworks Used:

- OWASP Testing Guide v4
- NIST SP 800-115
- SANS CWE Top 25
- PTES

Tools Employed:

- Burp Suite Pro
- Nmap, Nikto
- SQLMap
- ZAP
- Hydra
- Metasploit
- Manual Testing

4. Summary of Findings

Given below are the summary of the findings.

Severity	Count	Vulnerabilities
High	2	Privilege Escalation, WordPress Username Disclosure
Medium	5	Directory Traversal, Session Timeout, Brute-force Attack
Low	10	Missing Headers, Version Disclosure, CORS Misconfiguration

Total Findings: 17

5. Key Vulnerabilities

High Risk findings

1. Privilege Escalation

- **Issue**: Practitioners can view user lists and appointment rules meant only for managers.
- **Impact**: Unauthorized access to sensitive data or elevated functions.
- **Fix**: Enforce role-based access checks before serving privileged data.

2. Sensitive Info Disclosure (WordPress usernames)

- **Issue**: Public enumeration of WordPress users via /wp-json/wp/v2/users.
- **Fix**: Restrict access to the WP API or implement authentication controls.

Medium Risk findings

1. Account Harvesting

- **Issue**: Login error messages reveal account existence.
- **Fix**: Use generic error messages and rate-limiting.

2. Directory Traversal

- **Issue**: Path like ../server/pages-manifest.json can leak files.
- **Fix**: Normalize paths and validate input strictly.

3. Brute-force Login (WP Login)

- **Issue**: No rate-limiting or CAPTCHA on login page.
- **Fix**: Implement throttling, CAPTCHA, and lockout mechanisms.

4. Improper Session Timeout

- **Issue**: Session tokens valid for up to 3 months.
- **Fix**: Set session timeouts to 30 mins of inactivity.

5. Sensitive Data in Cookies

- **Issue**: Cookies store PII without encryption.
- **Fix**: Store sensitive data server-side or encrypt it with secure flags.

Low Risk findings

- **Authentication Bypass via direct access to document URLs**
- **Missing HTTPOnly and Secure flags on cookies**
- **Outdated jQuery/React libraries**
- **Weak CORS Policy (allowing all origins)**
- **WordPress default files accessible (readme.html, license.txt)**
- **Components with Known Vulnerabilities**
- **Concurrent account login**
- **Misconfigured Content Security Policy**
- **Missing Security Headers**
- **Software version disclosure in Response header**

6. Recommendations

- Enforce RBAC and validate JWT/auth tokens at every endpoint.
- Implement CAPTCHA and account lockouts to block brute-force.
- Sanitize all user inputs and normalize paths to avoid traversal.
- Use secure, HTTPOnly, and SameSite cookie flags.
- Periodically update JS libraries and hide version information.
- Apply strong CORS policies to whitelist only trusted domains.

7. Positive Observations

- The applications blocked several public exploits.
- Input validation layers prevented SQLi and Command Injection.
- Proper session and token handling was implemented in parts.

8. Lessons Learned

- Even low-risk issues like outdated libraries can become entry points for an attack.
- Simple misconfigurations like verbose error messages enable enumeration.
- Regular testing and updates are critical for HIPAA-aligned platforms like medical apps.

9. Conclusion

This VAPT engagement for the medical company revealed **17 open vulnerabilities**, ranging from simple misconfigurations to high-impact flaws like **privilege escalations**. Mitigating these findings will significantly enhance the security posture of the application and help in achieving compliance with industry standards.