OWASP Top 10 – PetStore Application Vulnerability Mapping

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk ID | Risk Name | Description | Example in PetStore Application | Mitigation |
| A01:2021 | Broken Access Control | Users access features or data without appropriate permissions. | Unauthenticated users can access cart or order summary URLs by directly entering them in the browser. | Implement strict access control checks on both client and server sides; deny access by default and validate roles per request. |
| A02:2021 | Cryptographic Failures | Failure to secure data in transit or at rest. | Login or registration forms may submit passwords without HTTPS; no mention of password encryption. | Enforce HTTPS for all endpoints; store passwords using salted hashing (e.g., bcrypt). |
| A03:2021 | Injection | User input is improperly handled and used in backend queries or scripts. | Product search box may be vulnerable to SQL or script injection if input is not sanitized. | Use parameterized queries and strong input validation; apply proper output encoding. |
| A04:2021 | Insecure Design | Lack of security controls in the application logic or architecture. | Quantity fields in the cart accept negative values or arbitrary numbers. | Apply secure design principles like data validation at every layer and abuse-case modeling. |
| A05:2021 | Security Misconfiguration | Incorrect server or application settings leave the system exposed. | Absence of security headers like CSP, HSTS, or X-Frame-Options; stack traces revealed on errors. | Enable all necessary HTTP security headers, turn off verbose error messages, and review default configurations. |
| A06:2021 | Vulnerable and Outdated Components | Use of outdated libraries, plugins, or platforms with known issues. | JavaScript libraries or dependencies may be outdated and unpatched. | Maintain a component inventory; use tools like `npm audit` and keep libraries up-to-date. |
| A07:2021 | Identification and Authentication Failures | Weak login mechanisms or insecure session handling. | No CAPTCHA on login form; sessions persist after logout; weak password rules. | Implement CAPTCHA, strong password policies, secure session handling, and enforce session expiration. |
| A08:2021 | Software and Data Integrity Failures | Lack of validation for critical updates or external content. | External JavaScript is loaded without Subresource Integrity (SRI); no CI/CD validation mentioned. | Use SRI tags, signed updates, secure deployment pipelines, and verify source integrity. |
| A09:2021 | Security Logging and Monitoring Failures | Application fails to log or alert on suspicious behavior. | Login failures or suspicious checkout attempts are not logged or monitored. | Enable audit logs, centralize monitoring, and set up alerts for critical events. |
| A10:2021 | Server-Side Request Forgery (SSRF) | Server makes requests on behalf of user without validation. | If any APIs accept URLs (e.g., profile images or feeds), attacker may attempt internal server access. | Whitelist approved domains, sanitize input, and restrict internal network access from external inputs. |