CSRF Report

localhost_csrf_2025-09-09_09-58-13

Executive Summary

This report details the results of a Cross-Site Request Forgery (CSRF) vulnerability assessment conducted on http://localhost:8082 on September 9, 2025. The assessment identified multiple exploitable vectors, indicating a significant risk of CSRF attacks. A total of 45 vectors were tested, with a high number of them found to be exploitable. Remediation is crucial to protect against unauthorized actions being performed on behalf of legitimate users.

The primary vulnerability lies in the lack of CSRF protection mechanisms, such as CSRF tokens, proper SameSite cookie attributes, and strict Origin/Referer checks. Without these protections, an attacker can potentially trick a user into performing actions they did not intend to, such as changing their password, making purchases, or modifying their account settings.

Report Summary

Generated Tue Sep 9 09:58:13 2025 | Base: http://localhost:8082 | Actions: 1 | Vectors: 45

■ *Exploited Vectors*

form_1 → img_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → script_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → iframe_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → meta_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → link_click (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → noreferrer_link (200) → Accepted no Referer Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → form_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → fetch_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → xhr_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → multipart_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → duplicate_token (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → samesite_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → referer_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → subdomain_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → method_override (200) → Accepted method override Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → img_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → script_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → iframe_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → meta_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → link_click (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → noreferrer_link (200) → Accepted no Referer Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → form_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → fetch_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → xhr_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → multipart_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → duplicate_token (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → samesite_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → referer_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → subdomain_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → method_override (200) → Accepted method override Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → img_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → script_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → iframe_get (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → meta_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → link_click (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → noreferrer_link (200) → Accepted no Referer Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → form_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → fetch_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → xhr_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → multipart_post (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → duplicate_token (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → samesite_refresh (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → referer_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → subdomain_bypass (200) → Accepted (heuristic) Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

form_1 → method_override (200) → Accepted method override Mitigation: Use CSRF tokens + SameSite=strict and strict Origin/Referer checks.

Detailed Results