Mini Case Study: Lightweight Web Hardening

Overview

Objective: run a focused blue team hardening sprint on a small e-commerce style website I help to maintain. The goal was to improve default protections, reduce attack surface, and build a simple runbook for ongoing reviews—without disrupting availability.

Scope & Tools

Scope	Public web surface (app layer + TLS edges), headers & cookies, obvious panel
Out of scope	Deep app pentest, credential testing, customer data, infrastructure internals.
Tools	Nikto 2.5, Nmap (default scripts), testssl.sh (fast profile).

Initial Observations

- Missing security headers: Strict-Transport-Security (HSTS), X-Frame-Options, X-Content-Type-Options.
- Session cookie lacked Secure/HttpOnly/SameSite flags.
- Admin/mail panels publicly reachable on standard paths.
- TLS at TLS 1.2 only with some CBC ciphers enabled; no TLS 1.3; OCSP stapling off.

Note: All hostnames, IPs, and cookies were redacted for this case study.

Targeted Hardening Actions

- Added security headers: HSTS (initial short max-age, no preload), X-Frame-Options: SAMEORIGIN, X-Content-Type-Options: nosniff; started a minimal Content-Security-Policy pilot.
- Secured session cookies: Secure, HttpOnly, SameSite=Lax; standardized cookie path & domain.• TLS tuning: trimmed legacy CBC ciphers; kept forward secrecy AEAD suites; prepared TLS 1.3 enablement.
- Reduced panel exposure: limited public discoverability for admin/mail endpoints; tightened auth &rate-limit.
- Response hygiene: removed X-Powered-By leak; normalized redirects; added cache controls for dynamic pages.

Results (High-Level)

- Stronger default protections and smaller attack surface.
- Cleaner TLS posture and improved external grade (A-class).
- Fewer noise findings in follow-up scans (headers + TLS).
- A lightweight runbook for periodic checks (monthly or post-deploy).

Small but Meaningful Extras

- Referrer-Policy and Permissions-Policy to reduce passive data leakage and limit browser features
- Access hygiene: panel 2FA and SSH key-only (no password auth)
- Light monitoring: scheduled monthly Nikto/Nmap+testssl checks to catch regressions

Next Steps

- Enable TLS 1.3 and OCSP stapling when hosting allows.
- Advance CSP from report-only to enforcing (iterative allow-listing).
- Introduce WAF rules or app-level rate limits for sensitive routes.
- Automate headers checks and TLS scans in CI or a monthly cron job.
