MP

Marius Pareščius

KIBERNETINIS SAUGUMAS

Security scan report · generated 2026-08-14 19:34

Report date/time: **2026-08-14 19:34 (+02:00)** · GMT / UTC: **2026-08-14 17:34** · Template version: **1.1** · App version: **2.99.3** · Scan type: **simple**

D

52/100

zero.webappsecurity.com

Weak security level — fixes recommended.

This is a **free, simple online test** meant for the site owner — **not for an IT specialist, DevOps or CISO**. It shows only publicly visible, common issues. **A full security audit is a paid service and far more thorough** (deeper testing, internal analysis, manual review). For an audit: pagalba@parescius.lt

Summary

We ran an external security scan of **zero.webappsecurity.com** — assessing 11 technical areas. Overall security level: **52/100 (D)**.

Found **10** items to fix (1 critical, 3 medium, 6 low priority); 7 checks passed successfully.

Critical vulnerabilities found — immediate action is required.

What the score means: **90–100 (A)** excellent · **80–89 (B)** good · **70–79 (C)** average · **50–69 (D)** weak · **below 50 (E/F)** risky — action needed. Scored from 100, deducting for each issue (up to 10 per issue).

Priority actions

Where to start — the most important issues and what to do about them.

1. **Could not verify the SSL/TLS certificate**
   Where: Hosting / server (TLS certificate)
   Install a valid TLS certificate (e.g. free Let’s Encrypt via your hosting panel) and enable HTTPS for the whole domain.

2. **Missing HSTS (Strict-Transport-Security)**
   Where: Web server (nginx / Apache config or hosting provider)
   On the web server add the response header: Strict-Transport-Security: max-age=31536000; includeSubDomains; preload.

3. **Missing Content-Security-Policy**
   Where: Web server (nginx / Apache config or hosting provider)
   On the web server define a CSP, e.g.: Content-Security-Policy: default-src 'self'; then gradually allow the sources you need.

Overview

Key technical parameters of the site at scan time.

IP address	**54.82.22.214**
SSL issuer	
Server	**Apache-Coyote/1.1**
HSTS	**no**
CSP	**no**
Reputation	**clean**

Issues found (10)

Issues are sorted by severity — most critical first. Each one: what it means and the concrete fix.

× Could not verify the SSL/TLS certificate -10

A secure (encrypted) connection to your site could not be established, or the certificate is invalid. Data between visitors and the site may travel unprotected (technically — no valid HTTPS/SSL).

Where to look / who is responsible: Hosting / server (TLS certificate)

How to fix: Install a valid TLS certificate (e.g. free Let’s Encrypt via your hosting panel) and enable HTTPS for the whole domain.

! Missing HSTS (Strict-Transport-Security) -10

The browser is not forced to always connect to your site securely (encrypted), so the first visit can be intercepted and the visitor redirected to a fake site (technically — the HSTS header is missing).

Where to look / who is responsible: Web server (nginx / Apache config or hosting provider)

How to fix: On the web server add the response header: Strict-Transport-Security: max-age=31536000; includeSubDomains; preload.

! Missing Content-Security-Policy -8

The site does not restrict where code may be loaded from, making it easier to inject a malicious script that steals visitors’ data (technically — no Content-Security-Policy, the main XSS defence).

Where to look / who is responsible: Web server (nginx / Apache config or hosting provider)

How to fix: On the web server define a CSP, e.g.: Content-Security-Policy: default-src 'self'; then gradually allow the sources you need.

! Missing X-Frame-Options / frame-ancestors -6

Your site can be invisibly embedded into someone else’s page and trick visitors into clicking where they did not intend (technically — no clickjacking protection).

Where to look / who is responsible: Web server (nginx / Apache config or hosting provider)

How to fix: On the web server add X-Frame-Options: SAMEORIGIN or the CSP directive frame-ancestors 'self'.

• Server discloses software versions -5

Your server publicly shows which software and version it runs. This helps attackers find a vulnerability matching exactly that version (technically — versions disclosed in response headers).

Where to look / who is responsible: Web server (nginx / Apache config)

How to fix: On the web server hide versions (e.g. server_tokens off; on Nginx, ServerTokens Prod on Apache, remove X-Powered-By).

• Missing X-Content-Type-Options -4

The browser may mis-“guess” a file type and execute malicious content (technically — no X-Content-Type-Options).

Where to look / who is responsible: Web server (nginx / Apache config or hosting provider)

How to fix: On the web server add the header: X-Content-Type-Options: nosniff.

• Missing Referrer-Policy -3

When a visitor moves from your site to another, the exact addresses and parameters of your pages can leak (technically — no Referrer-Policy).

Where to look / who is responsible: Web server (nginx / Apache config or hosting provider)

How to fix: On the web server add: Referrer-Policy: strict-origin-when-cross-origin.

• No IPv6 (missing AAAA record) -2

The domain has no AAAA record — the site is not reachable over IPv6. IPv6 is now standard; some users and mobile networks are IPv6-only and may reach the site slowly or not at all.

Where to look / who is responsible: DNS / hosting (IPv6)

How to fix: Add an AAAA DNS record and enable IPv6 on your server or CDN.

• Components with a disclosed version detected (check known CVEs)

The site discloses the versions of the components it uses. Attackers can use these to look for publicly known vulnerabilities (CVEs).

Where to look / who is responsible: Website (component / plugin versions)

How to fix: Regularly update all components to the latest versions and, where possible, hide version information.

• HTTP/3 (QUIC) not supported

The site does not support HTTP/3 (QUIC) — a newer, faster protocol more resilient to network loss. This is a performance/modernity recommendation, not a security flaw.

Where to look / who is responsible: Web server / CDN (HTTP/3)

How to fix: Enable HTTP/3 on the server (e.g. nginx with QUIC) or via a CDN such as Cloudflare.

Passed checks (7)

Checks your site passed successfully.

✓ Cookies use Secure/HttpOnly

✓ DNSSEC enabled

✓ Not found in threat databases

✓ Not on any blocklists

✓ Site IP is clean on reputation lists

✓ Domain registration date

✓ Google Safe Browsing: no threats found

Next steps

1. Start with the critical and high-priority issues (see "Priority actions").
2. Update outdated software and set up regular maintenance.
3. After fixing, re-run the scan to confirm the result.

Need help getting this sorted? We audit, fix the gaps and maintain it — **pagalba@parescius.lt**.