FREE REFERENCE GUIDE

Laravel CSP

Quick Reference

& Pre-Enforcement Checklist

WHAT'S INSIDE

- Nonce explained in plain English
- All 13 CSP directives with descriptions
- 12-item pre-enforcement checklist
- 5 common mistakes to avoid
- Quick-start Blade & Vite snippets

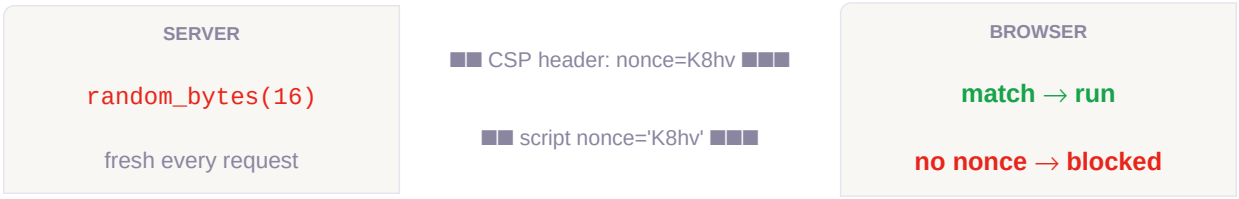
4 pages · Free · csp-generator.shakiltech.com

Shakil

WHAT IS A NONCE?

The key to making CSP work without breaking your app

A **nonce** (number used once) is a random string your server generates fresh on every page load. It goes into the CSP header and onto every script or style block you trust. The browser checks they match — a match means the script runs, no match means it's blocked. An attacker's injected script has no nonce, so it's always stopped.



The nonce changes on every request — an attacker who sees today's can't reuse it.

What a nonce is not

Not input sanitisation	It stops injected scripts running. It doesn't stop them being stored in your database.
Not encryption	The nonce is visible in HTML — that's fine. It expires when the request ends.
Not a CSRF token	Similar concept, completely different problem. You need both independently.

DIRECTIVE REFERENCE

All 13 directives and what they control

Directive	Value	What it controls
`default-src`	`'self'`	Fallback for any type not explicitly listed.
`script-src`	`'self' + nonce`	JavaScript. Scripts without the nonce are blocked.
`style-src`	`'self' + nonce`	CSS base fallback. Also governs @import in stylesheets.

style-src-elem	'self' + nonce	link rel=stylesheet tags and style blocks. Blocks can carry a nonce.
style-src-attr	'none'	Inline style= attributes. Cannot carry a nonce. Always block.
font-src	'self' data:	Font files. Add fonts.gstatic.com for Google Fonts.
img-src	'self' data:	Images. data: allows base64 inline images.
frame-src	'self'	iframe sources. Add video hosts as needed.
connect-src	'self'	fetch(), XHR, WebSockets, SSE. Add ws://localhost in dev only.
object-src	'none'	Browser plugins. Always block.
base-uri	'self'	Prevents base href injection attacks.
frame-ancestors	'none'	Who can iframe this page. Stops clickjacking.
upgrade-insecure-requests	(flag)	Auto-upgrades HTTP to HTTPS. Skip in local dev.

PRE-ENFORCEMENT CHECKLIST

Run through this before switching to enforcement mode

Start with **Content-Security-Policy-Report-Only**. Watch logs for 1–2 weeks. Tick every item below, then enforce.

- ✓ Vite::useCspNonce($nonce) called before $next($request)
  Without this, @vite() bootstrapper has no nonce and JS silently fails
- ✓ View::share('cspNonce', $nonce) in the middleware
  Makes $cspNonce available in every Blade template automatically
- ✓ All inline script blocks have nonce={{ $cspNonce }}
- ✓ All inline style blocks have nonce={{ $cspNonce }}
- ! Livewire — @livewireScripts(['nonce' => $cspNonce])
  Or @livewireScriptConfig(['nonce' => $cspNonce]) for Livewire v3
- ✓ onclick, onsubmit and event handler attributes removed from HTML
  Move all event handling into .js files
- ✓ style= attributes replaced with CSS classes
  style-src-attr is none — inline style attributes cannot carry a nonce
- ✓ All CDN domains are in the appropriate allowlist
- ✓ connect-src includes your APIs, WebSocket hosts, analytics
- ! connect-src does NOT include localhost in production
  Vite's ws://localhost:5173 must be gated behind app()->environment('local')
- ✓ Violation report route is live and throttled — throttle:5
  Without rate limiting an attacker can flood your logs
- → Verified headers at securityheaders.com — target A+

COMMON MISTAKES

What trips people up every time

Adding 'unsafe-inline' to script-src	Defeats the entire purpose — any inline script can run, including an attacker's.
Skipping Vite::useCspNonce()	@vite() injects an inline bootstrapper. Without the nonce it's blocked and your JS silently doesn't load.
Shipping localhost in connect-src	Gate ws://localhost:5173 behind app()->environment('local'). Don't rely on removing it manually.
Enforcing before report-only	Always deploy with Report-Only first. Watch logs. Enforce only when violations go quiet.
Confusing style-src-attr with style-src	style= attributes cannot carry a nonce. Set style-src-attr to 'none' and use CSS classes.

QUICK START

The three lines that matter most

1. Generate the nonce — before $next()

```
$nonce = base64_encode(random_bytes(16));
View::share('cspNonce', $nonce);
Vite::useCspNonce($nonce);
```

2. Set the header — after $next()

```
$response->headers->set(
'Content-Security-Policy',
implode('; ', $policy)
);
```

3. Blade templates

```
@vite(['resources/js/app.js'])
<script nonce="{{ $cspNonce }}">...</script>
<style nonce="{{ $cspNonce }}">...</style>
@livewireScriptConfig(['nonce' => $cspNonce])
```

RESOURCES

Go deeper

Free CSP Generator	csp-generator.shakiltech.com	Configure your policy visually and get the PHP middleware code
Full blog post	blog.shakiltech.com	Complete implementation — nonce, Vite, violation reporting, report-only mode
Grade your headers	securityheaders.com	Run after deploying. Target A+ before calling it done
Check your policy	csp-evaluator.withgoogle.com	Google's tool flags common weaknesses

Hosted violation reporting	`report-uri.com`	If you don't want to build your own reporting pipeline
Spatie laravel-csp	`github.com/spatie/laravel-csp`	Structured, per-policy-class approach with first-class test support
MDN CSP Reference	`developer.mozilla.org`	The definitive reference for every directive

Free to share. If this helped, the full blog post goes deeper on every topic.