

Domando a WooCommerce: Evasión de Hooks y Rastreo Fantasma

Estantería: DevSecOps e Infraestructura

Subtema: Gobernanza

mercedev.es — 2026-05-26 | Epic 6 - Fase 1

Domando a WooCommerce: Evasión de Hooks y Rastreo Fantasma

El Desafío

Durante la optimización extrema de una tienda híbrida (búsqueda del TBT 0ms), los análisis de rendimiento revelaron dos anomalías críticas: un Tiempo hasta el Primer Byte (TTFB) de casi 1000ms que estrangulaba el servidor, y la persistencia de un script de rastreo (`sourcebuster.min.js`) con un Tiempo de Bloqueo Total (TBT) de ~320ms. El script lograba evadir las órdenes estándar de purga (`wp_deregister_script` con prioridad 100) porque las versiones modernas de WooCommerce (8.5+) lo inyectan de forma condicional y tardía para su característica de "Order Attribution" (Rastreo de origen de pedidos).

La Maniobra

- 1. Amputación del cuello de botella DB:** Se extirpó una rutina de auto-sanación de bloques que colgaba del hook `init` . Escanear la base de datos en cada petición HTTP aniquilaba el TTFB.
- 2. El Exterminador de Scripts:** En lugar de intentar cazar el script en el *frontend*, se cortó la funcionalidad de raíz en el *backend*. Se inyectaron los filtros `woocommerce_order_attribution_tracking_enabled` (retornando `false`) y `pre_option_woocommerce_order_attribution_tracking_enabled` (retornando `'no'`). Además, se elevó la prioridad de purga general a `999` .

El Aprendizaje / Deuda Técnica

Combatir contra el código basura de los grandes CMS no siempre se gana en la capa de presentación (Frontend). Plugins pesados utilizan patrones escurridizos para forzar sus lógicas de telemetría y rastreo de usuarios.

La verdadera ingeniería de rendimiento exige conocer la arquitectura subyacente: apagar la característica (Feature Flag) directamente interceptando la consulta a la base de datos (`pre_option`) es la única garantía de que el código hostil jamás llegará a encolarse, devolviendo el rendimiento a un immaculado 100/100.