

Ceguera de Proxy y Bypass en Headless CMS

Cuadernillo | Vol. 1

mercedev.es — 2026-05-02 | Fase 9 (Inteligencia y Autonomía)

El Desafío (Síntoma)

Al intentar publicar artículos en WordPress desde la terminal local hacia el servidor de producción mediante la API REST, el servidor devolvía constantemente un error `401 Unauthorized`. Sin embargo, las credenciales eran correctas y el mismo script funcionaba a la perfección en el entorno `localhost`. Simultáneamente, WordPress en producción se negaba a mostrar la opción para generar Contraseñas de Aplicación por considerar el entorno inseguro, a pesar de tener un certificado SSL/TLS válido.

La Maniobra (Lógica)

El problema fue diagnosticado como **Ceguera de Proxy**. Servidores de alto rendimiento como CloudPanel utilizan Nginx y Varnish Cache por delante de PHP. Varnish, por seguridad, purga la cabecera `Authorization` antes de pasar la petición a PHP, dejando al CMS sin credenciales. Además, la conexión interna de Nginx a PHP se realiza por HTTP (terminación SSL en frontera), provocando la "Ceguera de HTTPS" en el núcleo de WordPress.

Soluciones aplicadas:

- Doble envío de credenciales:** El script en Python envía la cabecera estándar y una cabecera de contrabando `X-Authorization`, la cual los proxies ignoran y dejan pasar intacta. Un filtro en `functions.php` lee esta cabecera y restaura la memoria global de PHP justo antes de que WP valide al usuario.
- Extracción por Terminal (WP-CLI):** Ante la agresividad de OPcache sobrescribiendo variables de entorno, la contraseña maestra se generó descendiendo directamente al sistema operativo del servidor vía SSH usando `wp user application-password create`, saltándose al servidor web por completo.

El Aprendizaje / Deuda Técnica

Las configuraciones locales rara vez replican la agresividad de las barreras (WAF, Proxies, Cachés) de los entornos de producción en la nube. Las soluciones de enrutamiento temprano (*Shift-Left Routing*) inyectadas en el código de la aplicación son preferibles a alterar la infraestructura del servidor web, ya que garantizan que el ecosistema Headless siga siendo portable a cualquier otro proveedor de *hosting* en el futuro.