Prevención de Fuga de Credenciales en Integración Continua Local

Cuadernillo | Vol. 1

mercedev.es — 2026-05-01 | Fase 8 (Expansión de Contenido)

El Desafío (Síntoma)

Durante el desarrollo del motor de automatización social (LinkedIn), el script generó un archivo local con el token de acceso (`.linkedin_token.json`). Dado que el orquestador del proyecto (`merci-commit.py`) ejecuta un `git add .` automático para empaquetar los cambios, existía un riesgo crítico e inminente de Fuga de Datos (Data Leak): el token iba a ser inyectado en el control de versiones antes de poder ser excluido manualmente.

La Maniobra (Lógica)

Se detuvo el empaquetado y se aplicó una doble capa de contención (Shift-Left Security): 1. **Defensa Pasiva:** Se añadió el archivo `.linkedin_token.json` al `.gitignore` . 2. **Defensa Activa (Linter):** Se parcheó el orquestador de auditoría (`merci-audit.py`) añadiendo la llave maestra a la regla estricta `BANNED_TRACKED_FILE` . Si en el futuro Git intenta rastrear este archivo, el auditor abortará el commit automáticamente, emitiendo un error fatal.

El Aprendizaje / Deuda Técnica

En entornos con CI (Continuous Integration - Integración Continua) o automatización de commits, la confianza en el desarrollador no escala. Un archivo `.gitignore` es una medida pasiva que puede fallar si el archivo ya fue indexado o si se fuerza su inclusión. Implementar escudos activos (Linters) que auditen el área de preparación (*Stage*) de Git buscando credenciales antes de confirmar el empaquetado es la única garantía real de "Zero Trust" (Confianza Cero).