Infraestructura segura y autohospedada: cómo la tengo montada (y por qué)

Introducción

Este artículo documenta cómo está planteada una infraestructura autohospedada con foco en seguridad, control y reducción deliberada de superficie de ataque. El enfoque se integra en un stack basado en reverse proxy, autenticación centralizada y monitorización activa, donde cada componente cumple una función clara y no existe confianza implícita entre capas.

No se trata de paranoia ni de complejidad gratuita, sino de asumir que cualquier servicio expuesto será tocado, escaneado o forzado en algún momento, y diseñar el sistema para resistir, avisar y desgastar al atacante.

Infraestructura básica

Authentik: el portero de la fiesta

- Punto central de autenticación para subdominios y servicios internos.
- Nada accede directamente a las aplicaciones sin pasar por identidad previa.
- Uso sistemático de 2FA y autenticación fuerte.
- Integración con llaves físicas (YubiKey) para eliminar ataques basados únicamente en credenciales.

La autenticación no se delega en cada servicio, se centraliza y se endurece.

Cloudflare: la muralla exterior

- Oculta la IP real del servidor.
- Filtrado automático de DDoS, bots y tráfico genérico no deseado.

- Reglas personalizadas por país, ASN y patrones habituales de abuso.

La mayor parte del ruido muere antes de tocar infraestructura propia.

CrowdSec: la segunda línea

- Bloqueo activo de escaneos, fuzzing y rutas comunes de ataque.
- Protección de tráfico que no pasa por Cloudflare o llega por rutas laterales.
- Integración directa con el reverse proxy.
- Volumen real de bloqueo mensual del orden de millones de eventos.

No actúa como parche, sino como parte del flujo normal de tráfico.

Gestión de datos sensibles

Contraseñas a nivel absurdo con KeePass

- Gestor de contraseñas principal.
- Contraseñas largas, únicas y sin reutilización.
- Base de datos cifrada y sincronizada entre dispositivos propios.

La idea no es memorizar, sino hacer inviable la adivinación.

Códigos 2FA con Aegis

- Aplicación open-source y local.
- Base de datos cifrada.
- Sin dependencia de servicios externos.

El segundo factor no vive en la nube de terceros.

Syncthing: sincronización sin nube

- Sincronización cifrada punto a punto.
- Control total de nodos y versiones.
- Sin proveedores intermedios.

No existe el concepto de “restaurar desde otro”.

Monitorización y buenas prácticas

Wazuh: ojos en todas partes

- Servidor central autohospedado.
- Agentes en máquinas propias y entornos controlados.
- Alertas en tiempo real ante eventos sospechosos.
- Correlación y visibilidad sin depender de SaaS.

No se confía en que no pase nada: se vigila para saber cuándo pasa.

Revisión de logs

- Authentik, Cloudflare, Caddy y Wazuh revisados con regularidad.
- Las anomalías no se buscan manualmente: saltan solas.

Actualizaciones constantes

- Servicios siempre al día.
- Sin convivir con versiones obsoletas.
- Lo que no se mantiene, se elimina.

Políticas mínimas de seguridad en el reverse proxy (Caddy)

El reverse proxy no actúa únicamente como terminador TLS, sino como una capa defensiva activa:

- **Geobloqueo local por allowlist**, aplicado antes de servir contenido y sin depender exclusivamente del CDN.
- **TLS estricto**, sin negociación con versiones antiguas.
- **Headers de seguridad coherentes y agresivos** incluso en dominios inexistentes (HSTS, CSP, Permissions-Policy, anti-clickjacking).
- **Catch-all hostil por defecto** para subdominios no definidos, evitando enumeración y fingerprinting.
- **Métodos HTTP limitados** y respuestas explícitas para tráfico no esperado.
- **Integración nativa con CrowdSec** y logging estructurado orientado a análisis, no a volumen.
- **Endpoints de monitorización protegidos**, incluso para health checks internos.

El proxy frontal está diseñado para frustrar reconocimiento, reducir superficie de ataque y filtrar antes de delegar.

Ejemplo práctico: acceso a Gitea

1. **Cloudflare** filtra tráfico global y aplica reglas perimetrales.
2. **CrowdSec** bloquea escaneos y comportamiento anómalo que llega al servidor.
3. **Caddy** aplica geobloqueo, políticas HTTP y validaciones mínimas.
4. **Authentik** exige identidad, 2FA y llave física.
5. **Gitea** añade su propia autenticación y segundo factor.

El resultado es una cadena de obstáculos diseñada para frustrar y desgastar.

Resumen breve

Infraestructura basada en capas claras, sin confianza implícita y con fricción intencionada. La seguridad no es reactiva ni decorativa: es parte del diseño, está integrada y se mantiene activa en todo el flujo de acceso.

Referencias o enlaces de interés

- Instalación de Authentik
- Instalación de Syncthing
- Instalación de Wazuh
- Capítulo de CrowdSec
- Aegis (GitHub)
- KeePassXC
- KeePassDX

Revision #6
Created 2025-01-11 00:55:59 UTC by Juan Francisco
Updated 2026-01-17 09:41:49 UTC by Juan Francisco