> ./greyhack-llm --local --human-in-the-loop

GreyHack LLM

IA de seguridad informática que corre
en la máquina del cliente.
Copiloto de pentest y fábrica de informes.

Astillero · Fuerteventura · Noviembre 2026

Los auditores escriben más informes que pentests.

- Las herramientas de IA en la nube son rápidas, pero mandan el alcance, los hallazgos y los tokens capturados fuera de la red del cliente. Inaceptable para banca, defensa, sanidad o cualquier NDA seria.
- Los modelos locales, por su parte, se inventan CVEs, cabeceras HTTP y referencias WSTG cuando se les deja solos. No valen para un informe entregable.

> la respuesta

100 % local
+ motor determinista.

- El modelo trabaja en la Mac del operador. Nada sale a la red salvo que el admin lo encienda.
- El operador aprueba cada paso. Humano en el bucle. Esa aprobación es también la raya ética.
- La plataforma calcula CVSS, mapea cumplimiento y verifica lo que el modelo escribe. El informe sale con CWE, WSTG, ISO, ENS, PCI, RGPD, NIS2 y DORA — automático, consistente, sin inventos.

> dos casos de uso · un software

Mismo chasis, dos maneras de vender.

A · Copiloto de pentest

Para auditores, MSPs, equipos de seguridad internos.

Recon asistido → explotación guiada → informe vendible → push a Jira.

B · Plataforma de estudio

Para academias, formadores, pentesters preparándose para una certificación.

Rutas (OSCP, CEH, web, Blue Team…) + profesor + examen + flashcards.

Mismo motor, misma instalación, mismos modelos. Un cliente academia puede desactivar el laboratorio guiado; un auditor puede ignorar las rutas. Pero el producto es uno.

> A · copiloto · reconocimiento

Cablea tu nmap, curl, whatweb al contexto del modelo.

- **Detector determinista de tecnologías**: Node, WordPress, Tomcat, Jenkins, SMB, FTP, JWT, GraphQL y 15 más.
- **Fichas de ataque auto-cargadas** por tecnología. Si el objetivo es Juice Shop, la ficha de Juice Shop entra en el contexto sola.
- **Tablero de hallazgos estructurado**: tecnologías, puertos, endpoints con su HTTP, parámetros.
- **Filtro de catch-all/SPA** que no envenena el mapa con rutas falsas.

Node/Express · Juice Shop · WordPress · Jenkins · SMB · GraphQL · JWT · + 15 más

> A · copiloto · explotación

El modelo propone.
El humano aprueba.
La plataforma pega los secretos.

- Laboratorio guiado (ejecuta): cada comando pasa por la tarjeta de aprobación del operador. Nunca se lanza nada sin visto bueno.
- Arsenal de comandos verificados del operador, con huecos auto-rellenados: {OBJETIVO}, {PUERTO}, {URL}.
- Captura de botín: JWT, cookie o hash se almacenan como {TOKEN} fuera de la vista del modelo. Los siguientes comandos los pegan al ejecutar — el modelo nunca ve el secreto en claro.
- Verificador determinista de éxito/fallo por paso: ✓ ÉXITO, ⚠ PARCIAL, × FALLO.

> Este diseño resuelve el problema ético y el práctico a la vez: el humano es responsable del ataque, y el modelo no se puede llevar los secretos capturados a su contexto.

El informe sale con CVSS, CWE, WSTG y cumplimiento.

- CVSS lo calcula la plataforma, no el modelo. Cinco rutinas de rescate para vectores incompletos.
- Cumplimiento mapeado por CWE a ISO 27001:2022, ENS, PCI DSS v4.0.1, OWASP, RGPD, NIS2 y DORA. ~60 CWE cubiertos.
- ~19 detectores QA sobre la salida del modelo: WSTG inventado, sub-técnica MITRE falsa, cabecera HTTP como <meta>, cobertura CVSS/CWE/WSTG por hallazgo, evidencia sin respaldo.

ISO 27001 · ENS · PCI DSS v4 · OWASP Top 10 · RGPD · NIS2 · DORA

Cinco formatos de exportación. Push inteligente a tickets.

Exportación

- PDF desde el markdown fuente.
- SARIF 2.1.0 estándar: GitHub Code Scanning, Azure DevOps.
- CSV: Excel (BOM + preámbulo) y simple (pandas/Power BI).
- Dradis plain-text con tags.
- HTML autocontenido: un archivo, imprimible desde Ctrl+P.

Push a tickets

- Jira, GitHub Issues, DefectDojo.
- Credenciales cifradas en local.
- Re-empuje inteligente: cada ticket lleva el hash SHA-256 del hallazgo. Si cambió, se añade un comentario — no se duplica.
- Destino DEMO mock para enseñar sin crear ruido.

> B · suite educativa

El mismo motor enseña lo que acaba de hacer.

- **9 rutas de serie** en markdown estático (lecciones pre-escritas, nunca inventadas por el modelo): preparación OSCP, preparación CEH, pentest web, Blue Team, introducción a seguridad, Active Directory, PrivEsc Linux, OSINT, Bug Bounty.
- **Editor UI de rutas personalizadas** para que la academia cree las suyas.
- 🎯 **Simulador de examen** con SSE en tiempo real y revisión por categoría.
- 🃏 **Flashcards con SM-2** (repetición espaciada clásica).
- **Profesor pedagógico** y **CTF guiado** con pistas progresivas.

> Una academia instala el mismo .deb que un auditor. Desactiva el laboratorio guiado si no quiere que los alumnos ejecuten. El informe se convierte en «ejercicio resuelto» para la clase.

Local por defecto.
12 proveedores externos opcionales.

Local (recomendado)

Ollama + gemma4:12b-mlx para el día a día.

gemma4:26b (MoE, 3.8B activos) para el laboratorio guiado si la máquina llega.

La plataforma no llama a internet. Punto.

Proveedores externos

OpenAI · Anthropic · Gemini · DeepSeek · Groq · Mistral · Together · Qwen · GitHub Models · OpenRouter · Cerebras · openai_compat

Apagados de serie. Claves cifradas en local. El admin decide.

> cuatro clientes · un software

El mismo .deb para cuatro configuraciones.

Paranoico

Banca, defensa, sanidad, admin pública.

Todo local. Cero proveedores externos.

Pragmático

MSP, consultora.

Local + API externa selectiva para informes largos.

Con infra propia

GPU interna, vLLM/TGI.

openai_compat apunta a su endpoint.

Academia / formador

Formadores, centros, bootcamps.

Rutas + examen + flashcards. Sin laboratorio si no quieren que ejecuten.

> lo que GreyHack no hace

Por diseño.

- **No automatiza ataques a discreción**. El operador aprueba cada comando.
- **No genera prompts de anti-rechazo** ni jailbreaks contra otros modelos.
- **No crea páginas de phishing** ni material de ingeniería social.
- **No persigue el pentester autónomo** — inalcanzable hoy. Vender eso es prometer lo que no existe.

> El operador aprueba cada paso. Esa aprobación **es también la raya ética**: la responsabilidad del ataque siempre es del humano, no del software. Es la diferencia entre un copiloto que ayuda y una herramienta que se puede usar mal.

> validación en vivo · noviembre 2026

Cadena completa Juice Shop con el 26b.

1. nmap + curl → detecta Node/Express/Juice Shop.
2. ffuf → endpoints reales (filtra catch-all SPA).
3. SQLi en /rest/user/login con payload del arsenal.
4. Verificador determinista: ✓ ÉXITO automático.
5. Captura de botín: JWT de admin@juice-sh.op (717 car.) sin verlo.
6. whoami + cesta admin → cadena entera demostrada.
7. Informe PDF con 6 hallazgos, CVSS, CWE, WSTG, cumplimiento.

> ./contact --now

Hablemos.

LAB Astillero · Fuerteventura

TEL +34 652 947 524

GIT github.com/ericmugnoz