# PLAN.md — HoneypotLite

\*\*Proyecto:\*\* HoneypotLite

\*\*Autor:\*\* Victor Suarez Cruz

\*\*Periodo objetivo:\*\* 2025-10-03 → 2025-10-08

\*\*Regla:\*\* trabajar solo en red aislada (host-only / internal). Lee `SAFETY.md` antes de empezar.

---

## Cómo usar este PLAN

- Marca las tareas completadas cambiando `- [ ]` por `- [x]`.

- Haz commits claros en Git con mensajes tipo: `feat: milestone1.1 - VM honeypot creada`.

---

## Estado general

- Inicio: `2025-10-03`

- Fecha objetivo: `2025-10-18`

- Estado: [] Planificación [X]En progreso ☐ Pausapindo ☐ Completado

---

## Milestone 0 — Seguridad y preparación

- [X] \*\*0.1\*\* Redactar `SAFETY.md` (reglas éticas / alcance / firma).

\*\*Validación:\*\* archivo con al menos 5 reglas y firma (nombre + fecha).

- [X] \*\*0.2\*\* Crear repositorio (GitHub privado o local) y añadir `README.md`, `SAFETY.md`, `PLAN.md`.

\*\*Validación:\*\* repo creado y archivos presentes.

- [X] \*\*0.3\*\* Documentar stack elegido para el dashboard (PHP/HTML/CSS u otra).

\*\*Validación:\*\* línea en README indicando stack.

- [X ] \*\*0.4\*\* Plan de VMs (nombres, IPs tentativas, recursos, snapshots).

\*\*Validación:\*\* documento con esquema de red y recursos.

Nombre VM: Ubuntu Server Nombre VM: Kali

OS / Versión: ubuntu-22.04.3 OS / Versión: kali-linux-2024.4

IP (estática): 10.40.78.200 IP (estática): 10.40.78.201

Netmask: 255.255.240.0 Netmask: 255.255.240.0

Texto

El contenido generado por IA puede ser incorrecto. Texto

El contenido generado por IA puede ser incorrecto.

## Milestone 1 — Entorno / VMs (básico)

- [ ] \*\*1.1\*\* Crear VM \*Honeypot\* (Ubuntu Server) en red host-only.

\*\*Validación:\*\* IP privada anotada.

- [ ] \*\*1.2\*\* Crear VM \*Atacante\* (Kali) en la misma red host-only.

\*\*Validación:\*\* IP privada anotada.

- [ ] \*\*1.3\*\* Verificar conectividad entre VMs (ping desde Kali a Honeypot).

\*\*Validación:\*\* salida de `ping` con respuestas.

- [ ] \*\*1.4\*\* Crear snapshots / backups iniciales y documentar el procedimiento.

\*\*Validación:\*\* snapshot creado y nombre registrado.

---

## Milestone 2 — Captura básica de eventos (MVP)

- [X ] \*\*2.1\*\* Definir formato de log (campos mínimos: timestamp, src\_ip, src\_port, dst\_port, payload\_snippet).

\*\*Validación:\*\* ejemplo de 3 líneas de log (pueden ser artificiales).

- [X] \*\*2.2\*\* Proceso manual para registrar una conexión (pasos descritos paso a paso).

\*\*Validación:\*\* documento con pasos y resultado esperado.

- [X/2 ] \*\*2.3\*\* Simular conexión desde Kali y verificar que se ve la petición en Honeypot.

\*\*Validación:\*\* captura de pantalla o copia del output en Honeypot confirmando la petición.

- [X] \*\*2.4\*\* Crear estructura de carpetas en Honeypot (`honeypot/logs`, `honeypot/payloads`) y comprobar permisos.

\*\*Validación:\*\* paths creados y permisos anotados (owner/group/perm).

---

## Milestone 3 — Ingesta automática (sin publicar código)

- [X] \*\*3.1\*\* Diseñar flujo de ingest (escucha → captura N bytes → sanitizar → append al log).

\*\*Validación:\*\* diagrama simple o lista de pasos (4–7 pasos).

- [X] \*\*3.2\*\* Elegir formato de persistencia (CSV vs JSON vs DB) y justificar elección.

\*\*Validación:\*\* breve nota con pros/cons y elección final.

- [X] \*\*3.3\*\* Definir política de retención y rotación de logs (tamaño, tiempo, backups).

\*\*Validación:\*\* reglas claras (ej.: rotación semanal, max 100MB, compress después de X).

- [ ] \*\*3.4\*\* Probar ingest automática (producción de al menos 3 entradas automáticas reproducibles).

\*\*Validación:\*\* 3 entradas en logs con timestamps y campos correctos.

---

## Milestone 4 — Dashboard (visualización)

- [X] \*\*4.1\*\* Definir páginas del dashboard (lista eventos, detalle payload, estadísticas).

\*\*Validación:\*\* mockup simple o lista de páginas con descripción.

- [X] \*\*4.2\*\* Definir filtros y búsquedas (por IP, puerto, fecha).

\*\*Validación:\*\* lista de filtros y ejemplos de uso.

- [X] \*\*4.3\*\* Elegir estilo (paleta, tipografía) y documentarlo.

\*\*Validación:\*\* paleta + ejemplo de cómo se verá la tabla.

- [X] \*\*4.4\*\* Probar lectura de logs desde la web (mostrar entradas en el dashboard).

\*\*Validación:\*\* captura de pantalla del dashboard con datos de prueba.

---

## Milestone 5 — Detección básica y alertas

- [X] \*\*5.1\*\* Definir reglas de alerta (ej.: >5 intentos desde misma IP en 10 min).

\*\*Validación:\*\* documento con 3 reglas y sus umbrales.

- [ ] \*\*5.2\*\* Seleccionar canal de notificación (Telegram, email) y justificar.

\*\*Validación:\*\* nota con ventajas/desventajas.

- [ ] \*\*5.3\*\* Plan de prueba que dispare alerta (pasos y resultado esperado).

\*\*Validación:\*\* plan escrito; al menos una alerta simulada (evidencia).

---

## Milestone 6 — Hardening y limpieza

- [ ] \*\*6.1\*\* Verificar que el honeypot \*\*no ejecuta\*\* comandos recibidos (lista de comprobaciones manuales).

\*\*Validación:\*\* checklist completada con resultados.

- [ ] \*\*6.2\*\* Documentar procedimiento de limpieza / reset (snapshots revert, borrar logs).

\*\*Validación:\*\* instrucciones claras para reset del lab.

- [ ] \*\*6.3\*\* Definir controles de acceso al dashboard (usuario/contraseña) y documentar.

\*\*Validación:\*\* mecanismo descrito y prueba manual.

---

## Milestone 7 — Documentación y publicación

- [ ] \*\*7.1\*\* Redactar `README.md` final (ya completado).

\*\*Validación:\*\* README con secciones: Introducción, Requisitos, Setup, Uso, SAFETY.

- [ ] \*\*7.2\*\* Preparar `report/` con reportes sanitizados (al menos 2 pruebas).

\*\*Validación:\*\* 2 archivos `report\_YYYY-MM-DD.md` con datos sanitizados.

- [ ] \*\*7.3\*\* Crear demo (GIF o vídeo corto) y texto para LinkedIn/GitHub release.

\*\*Validación:\*\* demo en `report/demo.mp4` o `report/demo.gif`.

- [ ] \*\*7.4\*\* Subir repo a GitHub (privado o público según sanitización) y crear release.

\*\*Validación:\*\* repo en GitHub con release/tag.

---

## Checklist rápido (control)

- [ ] VMs creadas y comunicantes

- [ ] Logs con formato definido y >= 3 entradas

- [X] Dashboard conceptual listo

- [ ] Reglas de alerta definidas

- [X] SAFETY.md y README inicial creados

---

## Plantilla de Evidencia (pega esto cuando completes una tarea)

Rellena y pega en el chat para que valide y te dé la siguiente sub-tarea.

\*\*Tarea completada:\*\* (ej. `1.3 Verificar conectividad`)

\*\*Fecha / Hora:\*\* YYYY-MM-DD HH:MM

\*\*Evidencia:\*\*

- IP Honeypot: `192.168.56.101`

- IP Kali: `192.168.56.102`

- Comando de verificación ejecutado (texto): `ping -c 3 192.168.56.101`

- Resultado (pega la salida relevante o una línea): `64 bytes from 192.168.56.101: icmp\_seq=1 ttl=64 time=0.23 ms`

- Nota / Problema encontrado (si hay): `...`

> Yo revisaré la evidencia, te daré feedback y la siguiente sub-tarea priorizada.

---

## Convenciones de commits / ramas (recomendado)

- Rama principal: `main`

- Crea ramas por milestone/tarea: `milestone1/vm-setup`, `milestone3/ingest-design`

- Mensajes de commit: `feat(milestone1): crear VM honeypot` / `docs: SAFETY.md añadido`

---

## Notas finales

- Trabaja por pequeñas tareas y reporta evidencia.

- Si te atasca algo, pega la salida de comandos/error y la revisamos.

- Mantén el repo privado hasta sanitizar datos reales antes de publicar.