# Manual de Usuario SeaGuardian🛡️

Security Event Analyzer + GUI Matrix + Dashboard FastAPI

## Tabla de contenidos

1. [Introducción](#introducción)
2. [Arquitectura general](#arquitectura-general)
3. [Requisitos previos](#requisitos-previos)
4. [Instalación](#instalación)
5. [Variables de entorno](#variables-de-entorno)
6. [Uso de la GUI](#uso-de-la-gui)
7. [Uso del Dashboard Web](#uso-del-dashboard-web)
8. [Enriquecimiento OSINT](#enriquecimiento-osint)
9. [Reglas de detección](#reglas-de-detección)
10. [Exportación de resultados](#exportación-de-resultados)
11. [Integración con Palo Alto & Panorama](#integración-con-palo-alto--panorama)
12. [Flujo completo de trabajo](#flujo-completo-de-trabajo)
13. [Solución de problemas](#solución-de-problemas)
14. [Mantenimiento y actualizaciones](#mantenimiento-y-actualizaciones)
15. [Licencia](#licencia)

## Introducción

SeaGuardian es una herramienta de análisis de eventos de seguridad que combina:

* \*\*Interfaz gráfica estilo “Matrix”\*\* escrita en Tkinter.
* \*\*Dashboard web\*\* en tiempo real basado en FastAPI + Chart.js.
* \*\*Enriquecimiento OSINT\*\* a partir de AbuseIPDB, Cisco Talos, GreyNoise y OTX.
* \*\*Integración nativa\*\* con Palo Alto Networks (firewall) y Panorama.

Su objetivo es simplificar el análisis, correlación y respuesta ante incidentes de forma local, sin Deepender de plataformas SaaS, que Peden ser vulnerates y posterior mente, comprometidas y dibulgadas las heuristicas de analysis de las entidades.

## Arquitectura general

┌────────────┐ Logs CEF/LEEF ┌──────────────┐

│ GUI │ ◀─────────────── ▶│ FastAPI │

│ Tkinter │ │ Dashboard API│

└────────────┘ └──────┬───────┘

▲ Enriquecimiento OSINT │SQLite (events.db)

│ ▼

└───── AbuseIPDB / Talos / GreyNoise / OTX

* \*\*GUI\*\* permite cargar archivos o pegar líneas individuales, visualizar hallazgos y exportar.
* \*\*FastAPI\*\* expone endpoints REST que alimentan el dashboard.
* \*\*SQLite\*\* almacena un histórico mínimo para las gráficas.

## Requisitos previos

Componente Versión mínima

Python 3.9

FastAPI 0.110

Uvicorn 0.29

Pandas 2.2

ReportLab 4.0

Opcional YARA 4.4

Sistema operativo Windows Linux macOS

## Instalación

# Clonar el repositorio

git clone https://github.com/tuUsuario/SeaGuardian.git

cd SeaGuardian

# Crear entorno virtual

python -m venv venv

source venv/bin/activate # Windows: .\venv\Scripts\activate

# Instalar dependencias

pip install -r requirements.txt

# (Opcional) Habilitar YARA

pip install yara-python

## Variables de entorno

Copia `.env.example` a `.env` y rellena:

ABUSE\_TOKEN=\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GN\_TOKEN=\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OTX\_TOKEN=\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PAN\_TOKEN=\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Uso de la GUI

1. \*\*Abrir log\*\*: selecciona un archivo `.log` o `.txt` con eventos CEF/LEEF.
2. \*\*Pegar líneas\*\*: también puedes copiar-pegar eventos individuales en el cuadro de texto.
3. \*\*Des-ofuscar IOC\*\*: marca la casilla si tus IOCs vienen como `hxxp://` o `1[.]2[.]3[.]4`.
4. \*\*Analizar\*\*: corre el pipeline ➜ enriquecimiento OSINT ➜ detección ➜ acciones sugeridas.
5. \*\*Exportar\*\*: pulsa \*\*Excel\*\* o \*\*PDF\*\* para informes rápidos.
6. \*\*Palo Alto / Panorama\*\*: ingresa IP + API Key y recupera los últimos 1440 minutos de registros.

![GUI](docs/screenshot\_gui.png)

## Uso del Dashboard Web

* Inicia automáticamente en `http://localhost:8000/dashboard` al lanzar la GUI.
* Se actualiza cada \*\*15 s\*\* mostrando:

\* \*\*Top 10 IPs\*\* (bar chart).

\* \*\*Distribución de reglas\*\* (doughnut chart).

\* \*\*Dispositivos\*\* detectados (pie chart).

![Dashboard](docs/screenshot\_dashboard.png)

## Enriquecimiento OSINT

| Fuente | Métrica | Categoría derivada |

|--------|---------|--------------------|

| AbuseIPDB | `abuseConfidenceScore` | `malicious / suspicious / benign` |

| Talos | `reputation` | idem |

| GreyNoise | `classification` | idem |

| OTX | Nº de pulses | `hit / none` |

La columna \*\*`ip\_category`\*\* toma el peor caso entre todas.

## Reglas de detección

| Nombre | Condición | Acción sugerida |

|--------|-----------|-----------------|

| \*\*InfoLeak+Illegal\*\* | `cs4 == Information Leakage` y HTTP status ≠ 200 | Bloquear IP, revisar info sensible |

| \*\*Illegal OPTIONS\*\* | Método `OPTIONS` + status ≠ 200 | Deshabilitar OPTIONS, endurecer CORS |

| \*\*Rating ≥ 3\*\* | `cn2 >= 3` | Escalar a SOC, capturar evidencias |

| \*\*OSINT mismatch\*\* | IP maliciosa pero `cs6Label` ≠ \*Malicious IP\* | Añadir a blacklist |

| \*\*YARA match\*\* | Coincidencia de reglas YARA | Aislar host, análisis de malware |

| \*\*Malicious-AnySource\*\* | `ip\_category == malicious` | Bloquear inmediatamente |

| \*\*PaloAlto Threat\*\* | Severidad \*high/critical\* | Verificar política de firewall |

## Exportación de resultados

* \*\*Excel\*\* (`.xlsx`): hoja tabular con todas las columnas + plan de acción al final.
* \*\*PDF\*\*: informe compacto (~1000 líneas) con estilo monocromo-verde, ideal para adjuntar.

## Integración con Palo Alto & Panorama

1. En Configuración ➜ Panorama IP / Key.
2. Botón \*\*Palo Alto\*\*: obtiene logs de la API REST del firewall.
3. Botón \*\*Panorama\*\*: igual pero vía Panorama.

\*Se recomienda crear claves solo-lectura.\*

## Flujo completo de trabajo

graph LR

A[Recopilar eventos] --> B[Enriquecer OSINT + YARA]

B --> C[Aplicar reglas]

C --> D[Acciones sugeridas]

C --> E[Persistir en SQLite]

D --> F[Exportar Excel / PDF]

E --> G[Dashboard FastAPI]

## Solución de problemas

| Síntoma | Posible causa | Solución |

|---------|---------------|----------|

| \*\*ModuleNotFoundError\*\* | Dependencia faltante | `pip install -r requirements.txt` |

| \*\*GUI no abre\*\* | constructor `\_init\_` mal escrito | cambiar a `\_\_init\_\_` |

| \*\*Dashboard 404\*\* | Uvicorn no arrancó | revisa consola; puerto 8000 ocupado |

| \*\*Timeout OSINT\*\* | sin Internet o API Key inválida | revisa `.env` |

## Mantenimiento y actualizaciones

* Actualiza dependencias con `pip-compile` o `pip install -U ...`.
* Limpia la base SQLite periódicamente (`DELETE FROM events WHERE ts < ...`).
* Añade nuevas reglas YARA en `yara\_rules/\*.yar`.

## Licencia

Publicado bajo licencia MIT. Consulta [LICENSE](../LICENSE) para más detalles.