Reporte Proyecto Final
“Du-ia”

Autores:

- Malcolm Hiram Navarro Lopez
- Manuel Alejandro Heredia Nogales
- Jesus Ernesto Carrasco Teran

1. Resumen

El proyecto consiste en el desarrollo de un Agente de Ciberseguridad impulsado por Inteligencia Artificial. Este sistema actúa como un analista virtual capaz de recibir instrucciones en lenguaje natural (español), ejecutar herramientas de auditoría de red complejas de forma autónoma y, lo más importante, interpretar los resultados técnicos para ofrecer explicaciones claras y recomendaciones de seguridad.

2. Porque

Este proyecto surge como una aplicación práctica de los conocimientos adquiridos en la asignatura de Redes de Computadoras cursada el semestre anterior, con el fin de crear un agente que pueda hacer monitoreo de red y te de un reporte que resuma de manera que alguien ajeno a las redes de computadora pueda comprender.

2. Problema que Resuelve

El análisis de seguridad en redes enfrenta tradicionalmente tres barreras principales:

- **Complejidad Técnica:** Las herramientas de auditoría (como escáneres de puertos o analizadores de tráfico) requieren conocimientos avanzados de comandos y sintaxis, lo que limita su uso a expertos.
- **Saturación de Datos:** Estas herramientas generan miles de líneas de datos crudos (logs) que son difíciles de leer y correlacionar manualmente.
- **Riesgo Operativo:** Ejecutar comandos de seguridad sin el cuidado adecuado puede saturar una red o violar políticas de privacidad.

3. Solución y Funcionalidad

El sistema funciona como un intermediario inteligente entre el usuario y la infraestructura de red. Su funcionalidad se divide en tres etapas clave:

1. **Comprensión y Orquestación:** El usuario solicita una tarea y el agente entiende la intención y selecciona automáticamente la herramienta adecuada para el trabajo, ya sea un escaneo, una captura de tráfico o una búsqueda de información.
2. **Ejecución Segura:** Antes de realizar acciones sensibles que podrían afectar la red, el agente evalúa los riesgos y solicita confirmación al usuario, garantizando que siempre haya control humano sobre las operaciones críticas.
3. **Interpretación de Resultados:** En lugar de mostrar solo códigos técnicos, el sistema analiza los datos y genera un reporte en lenguaje sencillo.

4. Características Principales

- **Memoria y Continuidad:** Este agente "recuerda" conversaciones pasadas. Permite pausar una investigación y retomarla días después sin perder el contexto de lo que se estaba analizando.
- **Transparencia:** El sistema explica qué está haciendo en todo momento y por qué, educando al usuario durante el proceso.
- **Diversidad de Herramientas:** Integra múltiples capacidades en una sola interfaz: escaneo de vulnerabilidades, monitoreo de tráfico en tiempo real, análisis de registros (logs) del sistema y verificación de identidad de sitios web (DNS/WHOIS).

5. Estructura

```
topicos_Ia/
├── main.py                    # Punto de entrada
├── toolTest.py                # Versión original (referencia)
├── demo_sessions.py           # Demo de gestión de sesiones
├── requirements.txt           # Dependencias
├── .env                       # Configuración (API keys)
│
├── src/
│   ├── core/
│   │   ├── agent_controller.py    # Controlador principal
│   │   ├── tool_manager.py        # Gestor de herramientas
│   │   ├── interpreter.py         # Traductor de resultados
│   │   └── permissions.py         # Gestión de permisos
│   │
│   ├── tools/
│   │   ├── cai_tools_wrapper.py   # Herramientas CAI
│   │   ├── nmap_tool.py           # Escaneo de red
│   │   ├── whois_tool.py          # WHOIS y DNS
│   │   └── log_analyzer_tool.py   # Análisis de logs
│   │
│   ├── ui/
│   │   ├── cli_interface.py       # Interfaz de terminal
│   │   ├── custom_terminal.py     # Terminal personalizada (coordinador)
│   │   ├── terminal_display.py    # Funciones de visualización
│   │   ├── terminal_commands.py   # Manejador de comandos
│   │   ├── session_commands.py    # Comandos de gestión de sesiones
│   │   └── prompts.py             # Mensajes amigables
│   │
│   └── models/
│       ├── conversation_memory.py # Memoria de sesión
│       └── session_manager.py     # Gestión de sesiones persistentes
│
├── logs/                      # Logs de sesiones (JSONL)
├── reports/                   # Reportes generados (futuro)
├── memory/                    # Memoria persistente de conversaciones
└── docs/                      # Documentación adicional
    ├── GESTION_SESIONES.md    # Guía de gestión de sesiones
    ├── PERMISOS.md            # Documentación de permisos
    └── architecture.md        # Arquitectura del sistema
```