CyberSentinel-AI Knowledge Base

CyberSentinel-AI is an AI-powered cybersecurity tool that leverages LLM agents to detect, analyze, and score cyber threats. This document provides all the essential knowledge for understanding, using, and extending the project.

---

1. Project Overview

CyberSentinel-AI is designed to automate threat detection and security analysis using intelligent agents. It enhances cybersecurity workflows by:

- Detecting suspicious code or behavior
- Performing vulnerability assessments
- Scoring threats based on severity
- Logging results in a structured and readable format

Core Technologies

- **Python 3.10+**
- **Large Language Models (LLMs)** for analysis
- **Flask** (optional) for dashboards
- **JSON & CSV** for logging

---

2. Project Structure

```
CyberSentinel-AI/
│── README.md
│── requirements.txt
│── run_agents.py
│── run_servers.py
│── run_notebooks.sh
│
├── ai_agents/          # All AI-based detection and coordination agents
├── security_tools/     # Helper scripts and scanning tools
├── core_utils/         # Logging, configs, and server utilities
├── docs/               # Documentation and media files
└── notebooks/          # Jupyter demos and experiments
```

Folder Details

- **ai_agents/**
  - `caldera_agents.py` – LLM agents for adversary simulations
  - `code_agents.py` – Agents to analyze suspicious scripts
  - `coordinator_agents.py` – Coordinates multi-agent actions

    - `text_agents.py` – Processes and interprets textual threats
- **security_tools/**
    - `caldera_tools.py` – Helper for simulated attacks
    - `code_tools.py` – Analyzes scripts for potential malware
    - `web_tools.py` – Scans and interacts with web-based targets
- **core_utils/**
    - `constants.py` – Global constants and settings
    - `logs.py` – Handles JSON logging for results
    - `web_server.py` – Optional lightweight Flask server
    - `shared_config.py` – Configurations for agents
    - `ftp_server.py` – Simulates file transfer scenarios

---

3. Installation

1. Clone the repository:

```
git clone <your-repo-link>
cd CyberSentinel-AI
```

2. Install dependencies:

```
pip install -r requirements.txt
```

3. (Optional) Create a `.env` file based on `.env_template` for API keys.

---

4. Usage

Run the AI agents for threat detection:

```
python run_agents.py
```

Start the server for visualization (if Flask is enabled):

```
python run_servers.py
```

Run demo notebooks:

```
bash run_notebooks.sh
```

---

5. Features

1. **AI Threat Detection** – Leverages LLMs for anomaly detection
2. **Threat Scoring** – Assigns severity levels to detected threats
3. **Structured Logging** – JSON and color-coded CLI logs
4. **Optional Dashboard** – Real-time monitoring of threats

6. How to Extend the Project

- **Add New Agents**: Create a new file in `ai_agents/` and register it in `run_agents.py`
- **New Tools**: Place in `security_tools/` and update imports
- **Advanced Dashboards**: Enhance `web_server.py` with Flask or Streamlit

7. Contributor

Developed by **Pritam**