SMART OS BRAIN – AI-POWERED SECURITY MONITORING SYSTEM

Smart OS Brain is a modular, AI-assisted operating system security engine designed to **monitor, detect, analyze, and respond to threats in real time**.
It combines **rule-based security**, **AI anomaly detection**, **file system monitoring**, **ransomware behavior analysis**, and a **live GUI dashboard**.

This project demonstrates **advanced Python engineering**, **cybersecurity principles**, **multithreading**, and **backend–frontend integration**.

KEY FEATURES

SECURITY MONITORING ENGINE

- Real-time process scanning
- Suspicious behavior detection (CPU abuse, malicious naming patterns)
- Automatic response via quarantine mechanism
- Continuous threat logging

AI ANOMALY DETECTION

- Detects abnormal CPU and memory usage
- Adaptive thresholds to reduce false positives
- Whitelisting of trusted system processes
- Runs continuously in the background

REAL-TIME FILE WATCHER

- Monitors selected directories (e.g. Downloads)
- Detects suspicious file activity
- Triggers alerts on abnormal file behavior

RANSOMWARE BEHAVIOR DETECTION

- Detects mass file changes in short time intervals
- Flags encryption-like behavior
- Designed to simulate real ransomware detection logic

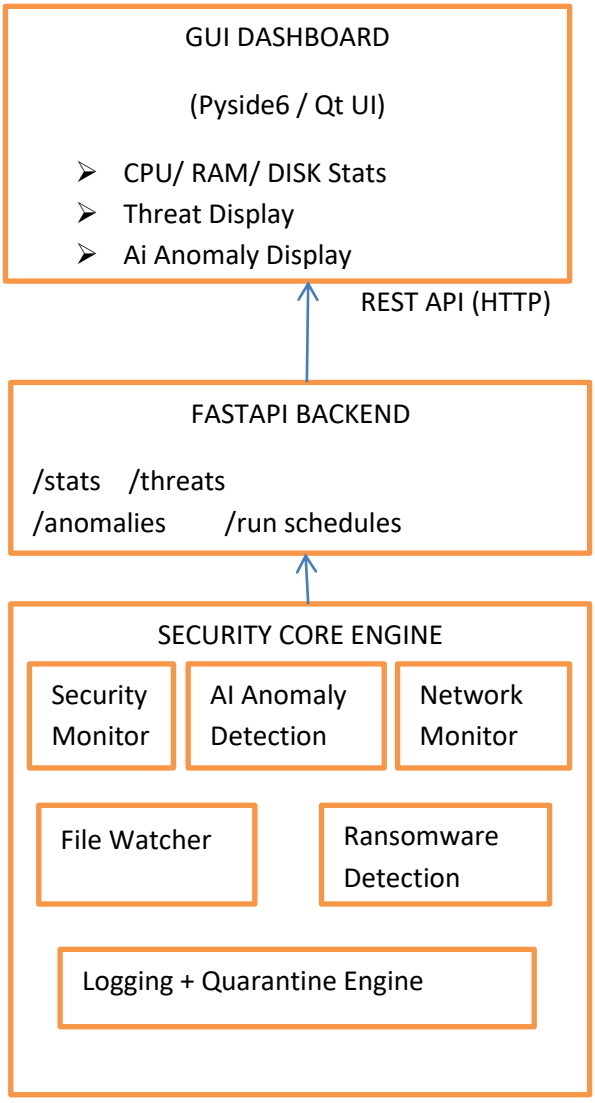
NETWORK TRAFFIC MONITOR

- Tracks network activity
- Flags abnormal traffic patterns
- Extensible for IDS/IPS integrations

GUI SECURITY DASHBOARD

- Live CPU, RAM, and Disk statistics
- Displays detected threats and anomalies
- Communicates with backend through REST API
- Built using **PySide6 (Qt for Python)**

SYSTEM ARCHITECTURE



TECHNOLOGY STACK

- **Language:** Python 3.10+
- **Backend API:** FastAPI + Uvicorn
- **System Monitoring:** psutil

- **File Watching:** watchdog
- **GUI:** PySide6 (Qt)
- **Concurrency:** threading
- **OS:** Windows (portable to Linux with minor changes)

HOW TO RUN:

Download file from github

Open folder directory on command prompt or terminal

1 Install Dependencies

```
pip install psutil watchdog fastapi uvicorn PySide6 requests
```

2 Start Backend

```
python main.py
```

Backend will run at:

```
http://127.0.0.1:8010
```

3 Launch GUI Dashboard

```
python -m ui.app
```

What You'll See

- Live CPU, RAM, Disk usage
- Detected threats in real time
- AI anomaly alerts
- Continuous monitoring without freezing the UI

Security & Safety Notes

- System-critical processes are whitelisted
- Quarantine avoids terminating core OS services
- Designed for **educational and defensive security research**

Skills Demonstrated

- Advanced Python programming
- Multithreading & concurrency
- OS process management
- AI-inspired anomaly detection
- Backend–frontend integration
- Secure system design

Future Enhancements

- Machine learning model (Isolation Forest / Autoencoder)
- Database-backed threat history
- Cross-platform support
- Web-based dashboard
- Exportable security reports

Author

Okwuiwe Alphonsus Jonas
Owner – **JOK TECH GLOBAL ENTERPRISE**
Cybersecurity & Systems Engineer