LLM-Powered SOC Assistant

A Natural Language Interface for Open-Source Security Monitoring

Progress Report 2

Umesh Kalupathirannehelage – 300389749

Applied Research Project

CSIS4495-071

1. Work Log Table

Student Name: Umesh Kalupathirannehelage

Date	Number of Hours	Description of Work Done
Sept 28, 2025	2	Installed and configured PostgreSQL extensions and verified ETL script loading alerts correctly after moving to AWS instance.
Sept 29, 2025	2.5	Deployed Wazuh Manager and re-registered existing Kali agent. Debugged UDP port 1514 connection issues using tcpdump and security group rules.
Sept 30, 2025	2	Configured new Ubuntu agent for additional host monitoring; tested log ingestion consistency and rule triggering.
Oct 1, 2025	3	Resolved Wazuh agent queue permission issue; ensured both agents send events successfully. Conducted SSH brute-force test to generate logs.
Oct 3, 2025	2	Updated FastAPI schema hints to generalize event parsing (auth, sudo, anomaly, FTP, etc.) and improved normalization logic for LLM output.
Oct 4, 2025	1.5	Fixed PostgreSQL password authentication issues in .env; verified FastAPI /healthz and /ask endpoints returning correct JSON responses.
Oct 6, 2025	2.5	Extended prompt schema to improve failed vs successful SSH classification using rules_desc ILIKE filters.
Oct 7, 2025	3	Designed and implemented a minimal front-end web interface (HTML + JS) for querying the SOC Assistant via REST API; not yet connected to backend.
Oct 9, 2025	1	Tested multi-agent visibility in PostgreSQL, confirmed new logs from both Ubuntu and Kali. Started research on chart-based visualization for the front-end.

2. Description of Work Done

Sept 28–30, 2025

During this week, I focused on migrating the entire application except Ollama to AWS EC2 instance and stabilizing the system after the move. I installed and configured PostgreSQL extensions to ensure the ETL pipeline could load alerts correctly from Wazuh into the database. The Wazuh Manager was deployed on the same instance, and I re-registered the existing Kali agent.

A major challenge was the UDP port (1514) connectivity between agents and the manager, I diagnosed the issue using tcpdump and adjusted AWS instance's security group inbound rules to allow traffic.

I tested consistent alert ingestion for both agents. This required verifying that Wazuh's event queue and the PostgreSQL schema remained in sync. Overall, the migration to AWS solved the previous resource limitations faced on VirtualBox and improved data flow stability across all components.

Oct 1-4, 2025

At the start of October, I focused on improving agent stability and FastAPI backend logic. A recurring "queue not accessible: permission denied" error from the Ubuntu agent required fixing Wazuh's internal socket permissions. Once resolved, both agents successfully reported events, and I performed SSH brute-force attempts, Nmap scans, File transfers using FTP between the two agents, to generate and validate real-time alerts. In parallel, I refined the FastAPI schema hints to expand beyond SSH, now covering authentication, sudo, anomaly, and FTP-related logs. This made the model's responses more generalized and realistic for real SOC queries. I also fixed PostgreSQL authentication issues in the .env configuration file that previously caused connection failures. Finally, I verified both /healthz and /ask endpoints, ensuring correct JSON responses from the FastAPI service. These updates enhanced the LLM pipeline's reliability and broadened its query understanding, addressing issues that caused mismatched SQL outputs in earlier builds.

Oct 6-9, 2025

In the most recent week, my focus shifted toward improving usability and expanding analytical features. I enhanced the prompt schema to improve accuracy in distinguishing successful vs. failed SSH events, leveraging refined keyword logic (rule_desc ILIKE '%fail%', %success%, etc.). This enhancement made natural language queries more accurate and context aware. Additionally, I designed and implemented a basic front-end web interface (basic HTML) to interact with the SOC Assistant's API. While it isn't connected to the backend yet, the interface demonstrates the structure for user queries and result visualization. I also began researching chart-based visualization to graph event trends such as failed SSH attempts or anomaly detections. Testing multi-agent visibility confirmed that logs from both the Ubuntu and Kali agents are stored and queryable in PostgreSQL, establishing a strong foundation for the analytical dashboard in the next stage of the project.

Since the last report, I have continued to maintain consistent updates to the GitHub repository, primarily focusing on a simple user interface, backend and data-collection improvements. Most of the effort during this phase went into adding agents, testing Wazuh log collection, and ensuring stable ETL ingestion into PostgreSQL.

The files/folders I have checked in the repo are as follows:

Under the implementation folder,

- **main.py (update)** – Updated twice with refined schema hints, better SQL normalization, and improved rule matching for SSH, authentication, and anomaly events.
- **ui/–** Newly added folder containing a index.html file that implements a simple front-end web interface for asking natural-language security queries.

Under the documents folder,

- Project_UKa749_Report2.pdf – This progress report document.