LLM-Powered SOC Assistant

A Natural Language Interface for Open-Source Security Monitoring

Progress Report 3

Umesh Kalupathirannehelage – 300389749

Applied Research Project

CSIS4495-071

1. Work Log Table

Student Name: Umesh Kalupathirannehelage

Date	Number of Hours	Description of Work Done
Oct 12, 2025	2	Debugged Wazuh agent communication errors between Ubuntu and Kali instances; fixed agent queue access and verified that both systems send alerts to the manager.
Oct 14, 2025	2.5	Enhanced ETL ingestion script to track last-ingested timestamps, preventing duplicate records in PostgreSQL.
Oct 15, 2025	2	Added /etl/run API endpoint and integrated it with the frontend “Fetch Latest Logs” button for on-demand updates.
Oct 17, 2025	2	Improved frontend layout and color scheme; adjusted component spacing and increased chart display area for better visualization.
Oct 18, 2025	2	Implemented chart logic for failed vs successful SSH logins; used Chart.js to generate dynamic percentage-based visualizations.
Oct 19, 2025	1.5	Verified multi-agent log ingestion consistency and improved SQL schema hinting for new categories (sudo, anomaly, FTP).
Oct 20, 2025	1.5	Researched Wazuh configuration to include more monitored sources (system audit, UFW, and kernel logs) into alerts.json.
Oct 21, 2025	2	Worked on documentation and began drafting midterm report sections for implemented features.
Oct 23, 2025	3	Organized the application code properly using separate folders for css and js. Added a dynamic loading spinner to visually indicate when queries are being processed by the backend
Oct 24, 2025	1	Completed integration verification and final polish before submitting progress report and midterm document.

2. Description of Work Done

Oct 12-17, 2025

During this week, my main focus was on stabilizing Wazuh agent communications and improving backend efficiency. I debugged issues between the Ubuntu and Kali agents, where the queue socket was inaccessible due to permission errors. After fixing the ownership and restarting Wazuh services, both agents were successfully sending alerts to the manager.

Next, I enhanced the ETL ingestion script to include a timestamp-tracking mechanism, which prevents duplicate records from being inserted into PostgreSQL during multiple ingestion runs. I also added the /etl/run API endpoint, enabling the frontend "Fetch Latest Logs" button to trigger ETL operations directly through the FastAPI backend. Finally, I refined the frontend interface, improving the overall color balance, adjusting layout spacing, and expanding the chart display section for better readability and usability.

Oct 18-24, 2025

This week was primarily focused on data visualization, schema improvements, and documentation work. I implemented Chart.js-based visualization that dynamically generates percentage comparisons between failed and successful SSH login attempts. This addition provided quick visual insights into authentication trends for SOC analysts. I also verified multi-agent ingestion consistency, ensuring both Ubuntu and Kali agents' alerts were recorded correctly in PostgreSQL. The SQL schema hints were further improved to handle broader categories like sudo activity, anomalies, and FTP logs. Additionally, I began researching ways to include additional log sources, such as /var/log/ufw.log, /var/log/syslog, and kernel logs, within Wazuh's configuration to expand alerts.json coverage.

By the end of this week, I started drafting midterm report documentation, including screenshots, descriptions of implemented features, and testing notes. Final integration testing was performed to confirm the synchronization of all components (ETL, FastAPI, Ollama, and PostgreSQL).

Under the implementations folder,

- **main.py (updated)** – Added chart logic for failed/successful SSH visualization, timestamp-based ETL deduplication, and /etl/run API integration.
- **ui/** – Updated index.html with an improved dark theme, optimized layout, larger chart display area, and "Fetch Latest Logs" button linked to backend ETL endpoint.
- **Folders db/ , etl, ui/css, ui/script** – Created proper folder structure and placed the relevant files inside the respective folder.

Under the documents folder,

- Project_UKa749_Report3.pdf – This progress report document.