

**Cyber Security Threat Hunting Through Log Analysis Using LLM with Agentic
Action – Project Plane**

Student : OMAR SALEM ALMUTAIRI

Student ID: 251002148

MIDOCEAN UNIVERSITY

FOR MASTER OF AI PROGRAM

GRADUATION PROJECT 1

Problem Statement

Current research in applying Large Language Models (LLMs) to cybersecurity threat hunting, as exemplified by frameworks like LogLLM, LogRAG, and Audit-LLM, has made significant strides in automated threat detection and explanation generation. These agentic and hybrid systems excel at parsing complex logs, identifying anomalies with high accuracy, and providing faithful explanations for their findings.

However, their function largely terminates at the point of alerting. There is a critical "last-mile" problem where the process of translating these high-fidelity, explainable alerts into concrete, timely, and appropriate security responses remains a manual, human-driven task. This gap fails to leverage the full potential of agentic AI, leaving security analysts burdened with the cognitive load of interpreting alerts and then deciding on and executing the correct response actions (e.g., isolating a host, blocking an IP address, terminating a process). This manual intervention introduces latency, increases the risk of human error, and limits the scalability of incident response, ultimately slowing down the security team's ability to mitigate threats in real-time. The core problem is the lack of an integrated mechanism for these advanced detection agents to autonomously propose or execute safe and context-aware response actions.

Research Objective

To design, develop, and evaluate a novel framework that extends agentic AI systems for Threat hunting using Log analysis via LLM by enabling them to autonomously generate and recommend safe, context-aware, and actionable responses directly from the security alerts they produce.

Research Questions

1. How can a set of pre-defined security response actions (e.g., `block_ip`, `isolate_host`, `terminate_process`) be structured and integrated as a functional "toolkit" for an LLM-based security agent?
2. What reasoning mechanisms are required for an agentic AI to effectively analyze an alert's context, severity, and its own generated explanation to accurately select and parameterize the most appropriate response action from its toolkit?
3. How can the safety and effectiveness of the agent-proposed actions be evaluated in a simulated environment to measure their ability to mitigate threats while minimizing operational risk and avoiding negative impacts from false positives?

12-Week Implementation Plan: LLM-Powered Threat Hunting System (No Fine-tuning)

Project Title: Cyber Security Threat Hunting Through Log Analysis Using LLM with Agentic Action

Goal: To design, develop, and rigorously evaluate a threat hunting system that leverages prompt engineering with an off-the-shelf open-source LLM to analyze security logs, identify anomalous patterns, and surface potential threats for cybersecurity analysts.

Pre-Requisite: Comprehensive Literature Review (Completed)

Total Duration: 12 Weeks (Intensive Implementation & Documentation)

Month 1: Data Preparation & LLM Integration (Weeks 1-3)

Objective: Secure and prepare your security log data, select and integrate the open-source LLM, and establish the core prompting pipeline.

- **Week 1: Rapid Data Acquisition & Initial Preprocessing**
 - **Action:** Immediately acquire the specific security log datasets you identified during your literature review as most suitable (e.g., Windows Event Logs, Sysmon, network flow data from public sources like CICFlowMeter). Prioritize readily available, well-documented, and reasonably clean datasets.
 - **Action:** Begin initial parsing. Develop scripts (Python, PowerShell, etc.) to extract raw log entries into a structured, usable format (e.g., JSON lines, CSV) based on the log type. Focus only on the fields relevant to the threat scenarios you're targeting.
 - **Action:** Set up your primary development environment (e.g., Python, Pandas, Hugging Face Transformers, necessary LLM inference libraries like transformers, torch or tensorflow).
 - **Deliverable:** Acquired datasets, initial parsing scripts, and a basic structured data sample. Configured development environment.
- **Week 2: Advanced Data Preprocessing & LLM Selection**
 - **Action:** Refine your parsing scripts. Transform structured log entries into coherent text sequences optimized for LLM understanding. This involves creating "log narratives" or concise summaries. For example: "Windows Security Event 4624. Successful logon. User: 'admin', Source IP: '192.168.1.5', Logon Type: 'Interactive'. Description: A user successfully logged into the system."
 - **Action:** Formally select your open-source LLM. Consider models like Llama 2 (7B/13B versions), Mistral, or Falcon, prioritizing models known for good instruction following and manageable inference

requirements for your hardware. Crucially, ensure the model is suitable for local inference or accessible via an API.

- **Action:** Download/install the chosen LLM and its tokenizer. Test basic inference to ensure it runs on your system.
 - **Deliverable:** Robust data preprocessing pipeline, LLM-ready text sequences, chosen open-source LLM loaded and functional for inference.
 - **Week 3: Baseline Model & Initial Prompt Engineering**
 - **Action:** Finalize your train, validation, and test dataset splits. These splits will be used for evaluating your prompt engineering strategies, not for model training.
 - **Action:** Implement a simple baseline log analysis model for comparison. This could be a keyword search, a basic regex engine, or a simple rule-based system targeting your specific threats.
 - **Action:** Begin initial prompt engineering. Draft several different prompt structures to guide the LLM to identify anomalies or threats in your preprocessed log entries. Experiment with "few-shot learning" by including 1-3 examples of "normal" and "malicious" logs directly in your prompt.
 - **Deliverable:** Ready-to-use dataset splits. Working baseline model code. Initial prompt templates for the LLM.
-

Month 2: Prompt Optimization & System Prototyping (Weeks 4-7)

Objective: Iteratively refine your prompt engineering, develop evaluation metrics, and integrate the LLM into a functional threat hunting prototype.

- **Week 4: Iterative Prompt Engineering (Round 1)**
 - **Action:** Systematically test your initial prompt templates on a small subset of your validation data. Analyze the LLM's responses.
 - **Action:** Refine prompts based on observations: Adjust wording, add more specific instructions, experiment with different "personas" for the LLM (e.g., "You are a cybersecurity analyst..."), or change the output format desired (e.g., JSON, simple bullet points).
 - **Deliverable:** Refined prompt templates, analysis of initial prompt performance.
- **Week 5: Iterative Prompt Engineering (Round 2) & Evaluation Metrics Definition**
 - **Action:** Continue refining prompts, focusing on reducing false positives and false negatives based on your validation set.

- **Action:** Define precise evaluation metrics relevant to threat hunting. Beyond typical ML metrics (Precision, Recall, F1-score), consider metrics like Mean Time to Detect (MTTD) simulation, False Alert Rate, and the quality/actionability of LLM-generated explanations.
- **Deliverable:** Optimized prompt templates, clear definitions of evaluation metrics.
- **Week 6: Core System Integration & Inference Pipeline**
 - **Action:** Build the core inference pipeline. This system should take a batch of preprocessed log entries, apply your best prompt template to each, send them to the LLM, and capture its responses.
 - **Action:** Implement a mechanism to parse and interpret the LLM's output (e.g., extracting "threat detected" flags, identifying specific IoCs, summarizing findings).
 - **Deliverable:** Functional LLM inference pipeline.
- **Week 7: Minimum Viable Product (MVP) & Initial Evaluation**
 - **Action:** Integrate the entire process: Log Ingestion -> Preprocessing -> Prompt Engineering -> LLM Inference -> Output Interpretation.
 - **Action:** Run your MVP on a small portion of your test dataset. Perform a preliminary evaluation using your defined metrics to ensure the system is working as expected end-to-end.
 - **Deliverable:** Working MVP of the LLM-powered threat hunting system. Initial performance insights on the test set.

Month 3: Evaluation, Analysis & Thesis Drafting (Weeks 8-11)

Objective: Rigorously evaluate your system, analyze its performance (especially the impact of prompt engineering), and rapidly draft the core chapters of your master's thesis.

- **Week 8: Comprehensive Quantitative Evaluation**
 - **Action:** Perform a thorough, large-scale evaluation of your LLM-based system on the entire unseen test dataset.
 - **Action:** Calculate and present all your defined key metrics: Precision, Recall, F1-score, False Positive Rate (FPR), False Negative Rate (FNR), and actionability scores if you devised them.
 - **Action:** Critically compare your LLM's performance (via prompt engineering) against your baseline model. Quantify the improvement and discuss where the LLM excels.
 - **Deliverable:** Detailed quantitative evaluation report with metrics, tables, and comparison to your baseline.

- **Week 9: Qualitative Analysis & Prompt Impact**
 - **Action:** Dive into qualitative analysis. Examine specific logs where the LLM performed exceptionally well and where it struggled (false positives, false negatives). Try to understand *why* based on the log content and your prompt design.
 - **Action:** Analyze the impact of different prompt engineering strategies. Document which elements of your prompts were most effective for different threat types or log characteristics.
 - **Deliverable:** Qualitative analysis report highlighting LLM strengths/weaknesses and prompt engineering insights.
- **Week 10: Thesis Drafting Sprint (Methodology & Results)**
 - **Action:** Begin aggressively drafting your thesis chapters. Focus immediately on Methodology (detailing your data acquisition, preprocessing, LLM selection, prompt engineering process, system implementation) and Results (presenting your quantitative and qualitative evaluation findings clearly with tables, figures, and charts).
 - **Deliverable:** Solid drafts of your Methodology and Results chapters.
- **Week 11: Thesis Drafting Sprint (Introduction, Discussion, Conclusion)**
 - **Action:** Draft the remaining thesis chapters:
 - **Introduction:** Revisit and refine your problem statement, clearly state your project's objectives, and highlight your contributions, particularly emphasizing the prompt engineering approach.
 - **Discussion:** Interpret your results in the context of your research questions. Discuss the broader implications of using off-the-shelf LLMs for threat hunting, acknowledge the limitations of your 12-week project (e.g., generalizability, reliance on prompt quality, compute limitations), and compare your work to fine-tuned or trained-from-scratch approaches.
 - **Conclusion & Future Work:** Summarize your main findings concisely and propose specific, actionable future research directions (e.g., exploring more advanced prompt techniques, integration with SIEMs, multi-modal log analysis).
 - **Deliverable:** A complete first draft of your entire thesis. Send this to your supervisor immediately for their critical review.

Week 12: Thesis Discussion & Finalization

Objective: Address supervisor feedback, finalize your thesis, and prepare for any oral defense or discussion.

- **Week 12: Feedback Integration, Final Polish & Presentation Prep**
 - **Action: Incorporate your supervisor's feedback on your thesis draft. This is the most crucial part of this week. Be prepared for substantial edits.**
 - **Action: Conduct a meticulous final proofread for grammar, spelling, formatting, and consistency. Ensure all citations and references are perfect.**
 - **Action: Develop your concise presentation slides for your thesis discussion/defense. Practice your presentation multiple times to ensure you can clearly articulate your project, methodology, key findings, contributions, and limitations within the allotted time.**
 - **Deliverable: Final, polished thesis document ready for submission. Complete presentation materials.**