



## **Project Proposal**

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

Prepared By

Name: OMAR SALEM ALMUTAIRI

ID No.: 251002148

Submitted in fulfillment of the requirements for the degree Master  
of Science in Artificial Intelligence

At

Collage of Informatics

Midocean University

UAE

Supervisor Name: Amjad Qatish

Collage of Informatics, Midocean University, Comoros

Email: Aubsalem558@gmail.com

## 1. Introduction and Background

The modern cybersecurity landscape is marked by increasingly sophisticated attack methods, a rising number of malicious activities, and highly dynamic threat actors. This changing environment requires a major shift from reactive security measures—which respond to incidents after they happen—to proactive strategies like threat hunting. Threat hunting actively seeks out potential threats that may have eluded initial security defenses. The goal is to spot malicious activities before they can inflict significant damage.

Traditional log analysis methods, once a cornerstone of cybersecurity, are becoming inadequate for this proactive approach. These conventional methods depend heavily on predefined rules and manual checks, making them time-consuming and prone to mistakes. The vast amount and complexity of log data produced by modern IT systems worsen these limitations, leaving traditional approaches largely ineffective at detecting new or evolving threats.

Large Language Models (LLMs) have the potential to transform log analysis in cybersecurity because they can process and interpret vast amounts of unstructured and semi-structured data. This ability allows LLMs to recognize subtle patterns, identify anomalies, and provide actionable insights that go beyond the limitations of traditional methods. By using their natural language processing capabilities, LLMs can shift from rigid, signature-based security to a more flexible, behavior-based anomaly detection system.

This project seeks to design, develop, and evaluate a threat hunting system that harnesses the transformative power of LLMs. The system will use prompt engineering with a readily available, open-source LLM to analyze security logs, identify unusual patterns, and uncover potential threats for cybersecurity analysts. This approach will focus on using agentic action to improve analysis, moving toward an intelligent system that can recognize subtle anomalies and suspicious behaviors that might otherwise go unnoticed.

## 2. Problem Statement

The fast-changing and increasingly complex nature of cyber threats poses a serious challenge to organizations. Current cybersecurity practices rely heavily on signature-based detection and manual log analysis, which are reactive and struggle to spot new, unknown, or highly evasive threats. This reactive stance causes several critical issues:

- **Delayed Threat Detection:** Traditional methods often find threats only after significant damage has already happened, resulting in substantial financial losses, data breaches, and harm to reputation.
- **Analyst Overload and Fatigue:** The vast volume and complexity of security logs overwhelm human analysts. This makes it nearly impossible for them to consistently

identify subtle signs of compromise within massive datasets. Missed threats can lead to analyst burnout.

- Inability to Identify Zero-Day and Advanced Persistent Threats (APTs): Signature-based systems do not work well against new, unseen threats (zero-days) and sophisticated, long-term attacks (APTs) that deliberately evade known patterns.
- Lack of Contextual Understanding: Current automated tools often fail to grasp the context and nuances of log data. This leads to a high rate of false positives and false negatives, which further burdens security teams.

Although Large Language Models (LLMs) have shown promise in understanding and generating natural language, their use in proactive cybersecurity threat hunting, particularly in generating agentic actions for sophisticated log analysis, remains largely unexplored. This research addresses a critical gap in current threat hunting methods that struggle to effectively identify subtle, evasive, and novel threats within vast security log datasets. Traditional analysis techniques have limitations, and LLMs offer untapped potential for generating intelligent, actionable insights beyond simple pattern matching. Specifically, there is a lack of systems that use LLMs not just for passive log analysis but for actively deriving and executing intelligent actions that refine the analysis process and pinpoint threats with greater accuracy and fewer false positives.

### **3. Motivation + Scope**

#### **Motivation**

The main motivation for this research comes from the urgent need to improve proactive threat detection in a security environment where traditional, signature-based defenses are increasingly ineffective against sophisticated attacks. Current methods are mainly reactive, which causes significant delays in identifying and mitigating threats. This increases the risk of major financial and reputational damage and puts great pressure on cybersecurity analysts, who must manually sift through huge amounts of complex log data.

We are specifically motivated by the transformative potential of Large Language Models (LLMs). We believe that their ability to understand context, identify subtle patterns, and interpret unstructured data can help overcome the drawbacks of traditional log analysis. An LLM-powered system can go beyond simple keyword searches and rule-based alerts, acting as an intelligent co-pilot to help analysts find threats more efficiently.

A key part of our motivation is to tackle the underexplored area of agentic action in LLM-based threat hunting. While some research has looked into using LLMs for log analysis and summarization, there is a significant gap in integrating them to perform intelligent, iterative actions that improve their analysis. Our research aims to build a system that not only understands the content of log data but also automates the process of threat investigation.

#### **Scope**

This project focuses on designing, developing, and evaluating a proof-of-concept threat hunting system that uses an open-source Large Language Model (LLM) and prompt engineering. The scope of this research is clearly defined as follows:

- **Log Data:** The system will be evaluated using publicly available, open-source cybersecurity log datasets (e.g., Windows Event Logs, Sysmon, network flow data). It will not be tested on private or proprietary data.
- **LLM & Approach:** The system will use a pre-trained, open-source LLM. This project will not include fine-tuning or training a custom LLM from scratch. The main method of interaction will be prompt engineering, with a focus on designing prompts that allow the LLM to perform agentic actions.
- **Functionality:** The system will identify unusual patterns and potential threats in the log data. A core part of the project is developing agents that can perform actions like:
  - **Data retrieval:** Fetching additional logs or context based on initial findings.
  - **Hypothesis testing:** Formulating and testing hypotheses about suspicious activity.
  - **Enrichment:** Adding context from external sources (e.g., threat intelligence feeds).
- **Evaluation:** The system's effectiveness will be assessed using a testing framework that measures its performance against set metrics, such as accuracy in detecting known threats, its ability to identify new anomalies, and the rate of false positives.
- **Limitations:** The project has a 12-week timeline, which limits its scope. The focus will be on creating a functional prototype rather than a production-ready system. The research will address limitations regarding generalizability, reliance on prompt quality, and computational limits. The system's performance will be compared to traditional methods but not necessarily to fully custom-trained models.

## 4. Contribution

This research makes several important contributions to cybersecurity threat hunting and the practical use of Large Language Models (LLMs). These contributions focus on overcoming the limitations of current methods and showcasing a new approach based on agents.

Our main contribution is the design and implementation of a threat hunting system that uses LLM-driven actions to improve log analysis. This goes beyond existing methods by enabling LLMs to perform a series of smart, iterative steps rather than just static analysis, such as summarization or pattern matching. This agent-based framework allows the system to:

- **Proactively Test Hypotheses:** The system can create hypotheses about potential threats based on initial log patterns and then automatically gather and analyze additional contextual logs to confirm or reject these hypotheses. This automates a key and time-consuming part of the human threat hunting process.
- **Generate Actionable Insights:** By linking its analysis to a set of defined actions, the LLM-powered system can deliver more precise and actionable findings for human analysts. This significantly cuts down on the noise and false positives typical in traditional alert systems.

- **Improve Efficiency and Scalability:** This research shows how an LLM can be used to intelligently prioritize log analysis tasks, reducing analyst workload and boosting the overall efficiency and scalability of threat hunting operations amid vast data volumes.

A secondary but equally important contribution is demonstrating the effectiveness of prompt engineering with a standard LLM for complex cybersecurity tasks. Instead of using expensive and time-consuming fine-tuning or training, this project shows that a powerful, accessible model can be effectively tailored for a specialized field through well-crafted prompts. This offers a cost-effective and practical solution for organizations wanting to use advanced AI technologies for security without needing extensive computational resources.

Lastly, this research presents a practical framework and a proof-of-concept prototype that can serve as a foundation for future work. By providing a working model, we set the stage for more exploration of LLM-powered agents in cybersecurity, including the potential for integration with other security tools and the analysis of more complicated, multi-modal data. The prototype and accompanying documentation will be a valuable resource for researchers and practitioners.

## **5. Research Questions**

- **RQ1: System Construction & Baseline:** How can I design and build a functional agent-based system that uses an open-source LLM and prompt engineering to analyze a given set of security logs?
- **RQ2: Agentic Actions & Prompts:** What are the most effective prompts and agentic actions needed to enable the LLM to perform specific threat hunting tasks, retrieving additional log data, and enriching findings for a human analyst?
- **RQ3: Performance Evaluation:** What specific metrics (e.g., precision, recall, F1-score) should be used to measure the system's performance?

## **6. Related Work**

The use of Large Language Models (LLMs) in cybersecurity has become an exciting area of research, especially for improving log analysis and proactive threat hunting. This section gives an overview of key frameworks and methods that support current research. It highlights their strengths and weaknesses to show the unique contributions of this project.

### **A. Foundational Approaches in LLM-Based Log Analysis**

Current research on using LLMs for log analysis can be divided into several categories:

- **Anomaly Detection:** This method aims to find deviations from normal behavior in log data that might indicate a potential threat. Frameworks like LogLLM use pre-trained models such as BERT and Llama to pull semantic vectors from log messages and classify

log sequences. LogLLM has shown better performance in both semantic understanding and anomaly detection compared to standard deep learning methods. Another notable framework, LogRAG, is a semi-supervised model that applies Retrieval-Augmented Generation (RAG) principles to reduce parsing errors and uses a single-class classifier to lower false positives. This approach has achieved significant improvements in F1 Score on public datasets.

- **Log Parsing and Embedding:** These are crucial steps in preparing raw log data for analysis. LLMeLog is a new method that uses in-context learning to improve log events before fine-tuning a pre-trained BERT model. This technique has earned F1-scores over 99% on public log datasets, even with limited labeled training data. A hybrid approach has also been proposed, which uses LLMs to create regular expressions (REGEX) for accurate log parsing and solid event embedding.
- **System State Analysis and Fingerprinting:** This new strategy goes beyond traditional log analysis by converting raw data into a unique "fingerprint." This fingerprint captures the core state of a system. This method allows for real-time anomaly detection by comparing these fingerprints to quickly identify changes, removing the need to manually sift through large log files.

## **B. Multi-Agent Collaboration and its Relevance to this Project**

A key area of research is the use of multi-agent collaboration frameworks for complex threat hunting tasks. These systems coordinate multiple intelligent agents to complete a complicated task.

- **Audit-LLM** is a prime example of a multi-agent framework designed for detecting insider threats. Audit-LLM has three collaborating agents: the Decomposer, which breaks down tasks using Chain-of-Thought reasoning; the Tool Builder, which creates reusable tools; and the Executor, which generates the final conclusion. The framework also has an Evidence-based Multi-agent Debate (EMAD) mechanism to refine conclusions. Experiments show that Audit-LLM outperforms existing models and greatly improves the clarity of the explanations it produces, addressing the important need for interpretability and explainability (XAI).

While frameworks like Audit-LLM have set the stage for multi-agent systems, our project stands out by concentrating on LLM-driven actions using an easy-to-access open-source model through prompt engineering. Our aim is not just to apply an LLM to analysis but to enable it to actively and gradually improve its own investigation. For example, it can retrieve additional logs based on an initial finding, as described in our second research question (RQ2). This adds to the existing research by providing a more practical and reachable implementation model for using LLM capabilities in proactive threat hunting.

## **C. Acknowledged Challenges in the Literature**

**The reviewed works also point out significant challenges that this research needs to address. These include:**

- Scalability: The vast amount of modern log data presents a major computational challenge for LLMs, which require plenty of resources.
- False Positives: The probabilistic nature of LLMs can cause incorrect alerts, leading to "alert fatigue" for security teams.
- Vulnerability to Adversarial Attacks: LLMs can be vulnerable to prompt injection, data poisoning, and model inversion attacks, which can undermine their integrity and effectiveness.
- Need for Explainability (XAI): The "black box" nature of LLMs makes it hard for analysts to grasp their reasoning, which weakens trust and complicates effective validation of their outputs. Our focus on creating a system that provides actionable, step-by-step insights directly tackles this challenge.

## **7. Expected Results**

**Based on the research questions and the proposed methodology, this project is expected to yield the following results:**

### **1. A Functional Prototype**

- A fully working prototype of a threat hunting system that uses an open-source LLM.
- The system will take in public security logs and analyze them using the developed agentic action framework.
- The prototype will act as proof of concept, demonstrating the feasibility and potential of an LLM-driven, agent-based approach to proactive threat hunting.

### **2. Enhanced Detection Capabilities**

- The system should show better performance in spotting unusual patterns and subtle signs of compromise compared to a simple, rule-based log analysis script.
- It is expected to achieve higher accuracy in detecting threats and a lower rate of false positives, as the agentic actions will allow for deeper, contextual analysis to confirm initial findings.

### **3. Optimized Prompt Engineering Framework**

- A set of clear, validated prompts will be an important outcome of this project. These prompts will provide a practical guide for effectively working with an LLM to handle complex cybersecurity tasks.
- The project will outline a straightforward method for creating prompts that help the LLM carry out agentic actions, such as retrieving more information or doing follow-up queries, thus providing a repeatable framework for future research and applications.

#### **4. Quantitative Performance Metrics**

- The research will offer a detailed quantitative analysis of the system's performance using standard cybersecurity metrics.
- Expected results will include specific values for precision, recall, and F1-score, which will directly address the third research question by showing a clear comparison between the LLM-driven system and the established baseline.

#### **5. Identified Limitations and Future Directions**

- The final report will clearly outline the limitations of the prototype, including issues related to computational resources, the quality of the LLM used, and the scope of a 12-week project.
- The research will conclude with a set of actionable recommendations for future work. These will include possible paths for improvement, such as exploring better prompt engineering techniques, integrating with other security tools, and applying the agentic action framework to different types of security data.

### **8. Project Steps And Timelines**

This section explains the 12-week plan for the "Cyber Security Threat Hunting Through Log Analysis Using LLM with Agentic Action" project. The aim is to create, develop, and evaluate a threat hunting system that uses prompt engineering with an open-source LLM to analyze security logs. This system will identify unusual patterns and highlight potential threats for cybersecurity analysts. The project spans three months, with specific goals and deliverables for each month.

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

**Objective: Secure and prepare security log data, choose and incorporate an open-source LLM, and set up the main prompting pipeline.**

- **Week 1: Rapid Data Acquisition & Initial Preprocessing**

- Action: Acquire suitable security log datasets, such as Windows Event Logs, Sysmon, and network flow data from public sources like CICFlowMeter. Focus on datasets that are readily available, well-documented, and relatively clean.
- Action: Start initial parsing by developing scripts (using Python, PowerShell, etc.) to transform raw log entries into a structured format (like JSON lines or CSV) according to the log type. Include only fields relevant to specific threat scenarios.
- Action: Set up the primary development environment, including Python, Pandas, Hugging Face Transformers, and necessary LLM inference libraries like transformers, torch, or tensorflow.

- Deliverable: Acquired datasets, initial parsing scripts, a basic structured data sample, and a configured development environment.
- **Week 2: Advanced Data Preprocessing & LLM Selection**
  - Action: Improve parsing scripts to change structured log entries into clear text sequences that are optimized for LLM. Create "log narratives" or concise summaries, such as: "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: Choose an open-source LLM (e.g., Llama 2, Mistral, or Falcon) that is known for good instruction following and has manageable inference requirements. Confirm that the model is compatible for local inference or available via an API.
  - Action: Download and install the selected LLM and its tokenizer. Conduct basic tests to ensure it functions on the system.
  - Deliverable: A solid data preprocessing pipeline, LLM-ready text sequences, and the chosen open-source LLM that is loaded and ready for inference.
- **Week 3: Baseline Model & Initial Prompt Engineering**
  - Action: Finalize splits for train, validation, and test datasets. Use these splits to evaluate prompt engineering strategies rather than for model training.
  - Action: Create a simple baseline log analysis model for comparison, like a keyword search, a basic regex engine, or a simple rule-based system targeting specific threats.
  - Action: Start initial prompt engineering by drafting various prompt structures to guide the LLM in identifying anomalies or threats in the preprocessed log entries. Try "few-shot learning" by including 1-3 examples of "normal" and "malicious" logs in the prompt.
  - Deliverable: Ready-to-use dataset splits, working baseline model code, and initial prompt templates for the LLM.

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

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

- **Week 4: Iterative Prompt Engineering (Round 1)**
  - Action: Test the initial prompt templates on a small subset of validation data and analyze the LLM's responses.
  - Action: Adjust prompts based on observations by changing wording, adding specific instructions, trying different "personas" for the LLM (like "You are a cybersecurity analyst..."), or altering the desired output format (e.g., JSON, simple bullet points).

- Deliverable: Improved prompt templates and an analysis report on initial prompt performance.
- **Week 5: Iterative Prompt Engineering (Round 2) & Evaluation Metrics Definition**
  - Action: Keep refining prompts, focusing on reducing false positives and false negatives using the validation set.
  - Action: Establish precise evaluation metrics relevant to threat hunting. In addition to standard ML metrics (Precision, Recall, F1-score), include metrics like Mean Time to Detect (MTTD) simulation, False Alert Rate, and the effectiveness of LLM-generated explanations.
  - Deliverable: Optimized prompt templates and clear definitions of evaluation metrics.
- **Week 6: Core System Integration & Inference Pipeline**
  - Action: Create the core inference pipeline. This system should process a batch of preprocessed log entries, apply the best prompt template to each, send them to the LLM, and capture the responses.
  - Action: Implement a method to parse and interpret the LLM's output, such as extracting "threat detected" flags, identifying specific IoCs, and summarizing findings.
  - Deliverable: A functional LLM inference pipeline.
- **Week 7: Minimum Viable Product (MVP) & Initial Evaluation**
  - Action: Integrate the entire process: Log Ingestion, Preprocessing, Prompt Engineering, LLM Inference, and Output Interpretation.
  - Action: Test the MVP on a small portion of the test dataset. Conduct a preliminary evaluation using the defined metrics to ensure everything works as expected from start to finish.
  - Deliverable: A working MVP of the LLM-powered threat hunting system with initial performance insights from the test set.

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

**Objective: Evaluate the system thoroughly, analyze its performance (particularly the influence of prompt engineering), and quickly draft the main chapters of the master's thesis.**

- **Week 8: Comprehensive Quantitative Evaluation**
  - Action: Conduct a detailed, large-scale evaluation of the LLM-based system on the entire unseen test dataset.
  - Action: Calculate and present all key metrics: Precision, Recall, F1-score, False Positive Rate, False Negative Rate, and actionability scores (if applicable).
  - Action: Critically compare the LLM's performance (through prompt engineering) with the baseline model. Measure the improvement and highlight areas where the LLM excels.
  - Deliverable: A detailed quantitative evaluation report featuring metrics, tables, and a comparison to the baseline.
  
- **Week 9: Qualitative Analysis & Prompt Impact**
  - Action: Conduct qualitative analysis by examining specific logs where the LLM performed well and where it faced challenges (false positives and false negatives). Investigate the reasons based on the log content and prompt design.
  - Action: Assess the impact of different prompt engineering strategies. Document which prompt elements were most effective for various threat types or log features.
  - Deliverable: A qualitative analysis report that highlights the LLM's strengths and weaknesses along with insights from prompt engineering.
  
- **Week 10: Thesis Drafting Sprint (Methodology & Results)**
  - Action: Start drafting thesis chapters, focusing on Methodology (covering data acquisition, preprocessing, LLM selection, prompt engineering, and system implementation) and Results (showing quantitative and qualitative findings with tables, figures, and charts).
  - Deliverable: Strong drafts of the Methodology and Results chapters.
  
- **Week 11: Thesis Drafting Sprint (Introduction, Discussion, Conclusion)**
  - Action: Complete the remaining thesis chapters:
    - Introduction: Review and refine the problem statement, clearly explain project objectives, and emphasize contributions, especially the prompt engineering approach.
    - Discussion: Interpret results in light of research questions, discuss the broader implications of using off-the-shelf LLMs for threat hunting, acknowledge project limitations (like generalizability and reliance on prompt quality), and compare this work to fine-tuned or trained-from-scratch methods.

- Conclusion & Future Work: Summarize main findings succinctly and propose specific, practical future research directions, such as exploring advanced prompt techniques or multi-modal log analysis.
- Deliverable: A complete first draft of the entire thesis.

## **Week 12: Thesis Discussion & Finalization**

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

- Action: Incorporate feedback from the supervisor on the thesis draft and be ready for significant edits.
- Action: Conduct a thorough final proofread for grammar, spelling, formatting, and consistency. Confirm all citations and references are accurate.
- Action: Prepare concise presentation slides for the thesis discussion or defense. Practice the presentation several times to ensure clear communication of the project, methodology, findings, contributions, and limitations within the given time frame.
- Deliverable: A final, polished thesis document ready for submission along with complete presentation **materials**.

## 9. References

- [1] S. Akhtar, S. Khan, and S. Parkinson, "LLM-based event log analysis techniques: A survey," arXiv preprint arXiv:2502.00677, 2025.
- [2] L. Zhang, T. Jia, M. Jia, and Y. Li, "LogRAG: Semi-Supervised Log based Anomaly Detection with Retrieval-Augmented Generation," ResearchGate, 2025.
- [3] A. Fariha, V. Gharavian, M. Makrehchi, S. Rahnamayan, S. Alwidian, and A. Azim, "Log Anomaly Detection by Leveraging LLM-Based Parsing and Embedding with Attention Mechanism," ResearchGate, 2024.
- [4] S. Akhtar, S. Khan, and S. Parkinson, "LLM-based event log analysis techniques: A survey," arXiv preprint arXiv:2502.00677, 2025.
- [5] C. Song, L. Ma, J. Zheng, J. Liao, H. Kuang, and L. Yang, "Audit-LLM: Multi-Agent Collaboration for Log-based Insider Threat Detection," arXiv preprint arXiv:2408.08902, 2024.
- [6] M. R. Hawthorne, S. L. Plotkin, and B. Douglas, *Fingerprints: Analysis and Understanding the Science*, 2nd ed. Routledge, 2021.
- [7] E. Shayegani, M. A. A. Mamun, Y. Fu, P. Zaree, Y. Dong, and N. B. Abu Ghazaleh, "Survey of Vulnerabilities in Large Language Models Revealed by Adversarial Attacks," arXiv preprint arXiv:2310.10844, 2023.
- [8] Analytics Vidhya, "Exploring Ethics and Privacy in the World of Advanced Language Models," Analytics Vidhya Blog, 2023.
- [9] M. Q. Li and B. C. M. Fung, "Security Concerns for Large Language Models: A Survey," arXiv preprint arXiv:2501.04437, 2025.
- [10] E. Kate, "Explainable AI (XAI) in Cybersecurity: Bridging the Gap Between AI and Human Understanding," ResearchGate, 2025.
- [11] International Journal of Multidisciplinary Scientific Research and Development, "Human-in-the-Loop Intelligence: Advancing AI-Centric Cybersecurity for the Future," ResearchGate, 2023.
- [12] M. T. Alam, D. Bhusal, and N. Rastogi, "CTIBench: A Benchmark for Evaluating LLMs in Cyber Threat Intelligence," Neural Information Processing Systems, 2024. [13] Researcher.Life, "What is the cost-effectiveness of using AI in cybersecurity?," R Discovery- Researcher.Life, 2024