

Harnessing Large Language Models for Cybersecurity Threat Hunting: A Log Analysis and Agentic Action Framework

Omar Salem Almutairi MidOcean University
Email: 251002148@st.midocean.eu.km

Abstract—With the rising content and complexity of unstructured log data growing by leaps and bounds, traditional rule-based systems may not suffice to carry out effective cyber threat hunting. This paper introduces a novel **Large Language Model (LLM) approach** for real-time threat detection that adjusts to the context, focusing specifically on the analysis of logs. We validate our approach on a synthetic but realistic chain of 100-plus intermingled log lines to simulate the entire attack lifecycle: Initial Access, Persistence, and Exfiltration through comparison with the gpt-oss-120b model. The model achieved perfect quantitative accuracy with **100% Detection Rate (Recall)** and **0% False Positive Rate (FPR)** by aggregating multiple failed login attempts into a single brute-force finding, along with being capable of accurately classifying benign system noise. Moreover, the model has demonstrated a key feature in the tool box for **Agentic Action** by providing a 100% syntactically correct and full remediation script across the network, process, and file-system layers. These findings demonstrate the effectiveness of LLMs as robust threat hunting agents in the scope of threat detection (context-aware agents) and action in preventing alert fatigue in an environment and support for immediate containment of threats.

Index Terms—Large Language Models (LLMs), Threat Hunting, Log Anomaly Detection, Zero-shot Learning, Agentic Action, MITRE ATT&CK Framework.

I. INTRODUCTION

TODAY’S cyber threat environments require a transition from reactive incident response to proactive threat hunting. The sources for most of this work (system, network and application logs) have ballooned to become non coherent and far too much of a jumble in many places. Conventional log analysis techniques that rely on static signatures and inflexible correlations can be overwhelmed by the large amount of data and cannot discriminate emerging or subtle attack patterns. This limitation gives rise to a major research gap which involves a smartly and computationally feasible system that is able to perceive the *context* and *narrative* of an attack concealed between different types of log entries, regardless of the format. Large Language Models (LLMs) possess the capability to process, interpret, and generate humanlike text and are particularly well positioned to contribute to this space, as semantic interpreters of raw log data. This paper presents and tests a new LLM based framework of log analysis and shows its utility in scenario of zero-shot threat hunting.

This work provides the following key contributions:

- 1) **Validation of zero-shot threat hunting:** Proving the capability of a monstrous LLM (gpt-oss-120b) to con-

duct multi-stage attack reconstruction simply based on log narrative analysis without any special modification.

- 2) **Noise Reduction and Aggregation:** Quantifying the model’s potential to reduce alert fatigue by aggregating data from multiple low level log events (e.g., > 30 failed logins) into one high-fidelity security finding (Brute Force).
- 3) **Agentic Action Capability:** Presenting an approach to introducing and evaluating the generation of immediate, contextually relevant and syntactically correct remediation scripts for containment and mitigation.

II. RELATED WORK

The development of AI used in log analysis has progressed from machine learning classifiers (SVMs, Isolation Forests) to deep learning techniques (such as recurrent neural networks and autoencoders, namely LogAnomaly, DeepLog). More recent work has investigated some more specific LLM frameworks—e.g., LogLLM [1], LogRAG [2]—that address improved semantic understanding and false positive reduction. Our work contrasts by concentrating on the **end-to-end zero-shot reasoning** capabilities of a large, general LLM towards more complex, narrative-focused threat hunting. This paper emphasizes, uniquely, the pivotal transition from detection (classification) to automated response generation (**Agentic Action**), a potential improvement from the detection-oriented state of the art.

III. THE PROPOSED METHOD: LLM THREAT HUNTING

The proposed approach uses a large scale LLM as an advanced reasoning engine for threat hunting. The key to its power is the application to help **contextualize, correlate, and generate** actionable intelligence under one and the same actionable action on actionable intelligence itself.

A. Core Architecture

The approach uses strong, high parameter LLM (gpt-oss-120b in this study), and is fed raw unstructured log data as a prompt, tailored according to the data. The prompt defines LLM operations, dataset, and a multi-step output, which requires the model to undertake a complex Chain-of-Thought processing to: 1) Filter out undesired noise and 2) Recognise and correlate malicious events 3) Reconceive the attack timeline 4) Generate remediation actions.

B. Generate Agentic Actions

The framework incorporates the **Agentic Action** ability, which is an aspect of this proposed environment, that is not possible in any other architecture. The model, after successfully performing a positive detection, was requested to synthesize knowledge of system utilities (e.g. `iptables`, `pskill`, `sed`) to produce a functional script for real-time threat containment. It enables the LLM to go from simply an analytical tool to an active response agent.

IV. EXPERIMENTAL METHODOLOGY

In this chapter the large language model (LLM), the design of the cyber-attack simulation, preparation of log data, and the metrics used to evaluate the model in threat hunting are reported.

A. Threat Hunting Model and Configuration

The log analysis and threat hunting model is the **gpt-oss-120b** model. The 120 billion parameter, LLM is an open-source, powerful framework and it has many features, including complex reasoning and zero shot analysis.

- **Model:** gpt-oss-120b.
- **Architecture:** A transformer-only decoder-only.
- **Deployment:** The model was run at a local scale on an individual cluster with N NVIDIA H100 GPUs with X GB of VRAM as well as on the V version of the vLLM inference engine to achieve the best throughput and latency.
- **Inference Mode:** Zero-Shot Task Execution was used.
- **Input Context:** A fixed context window was utilised to encapsulate the entire log chain within one prompt.

B. Preparation / Simulation of Log Data

To thoroughly test the ability of this model to reconstruct a complex, multi-stage attack a synthetic, yet realistic log dataset was formed.

1) *Simulation of the Attack Lifecycle:* It produced the **100+ mixed log lines** of log data which simulated the entire cyber-attack lifecycle against the Linux server environment. The attack process following the *MITRE ATT&CK Framework* covered techniques in the following three steps:

- 1) **Initial Access (T1110 - Brute Force):** 30+ failed SSH logins from an external IP address (104.28.2.14).
- 2) **Persistence (T1053.003 - Cron Job, T1543.002 - Systemd Service):** A malicious cron job and persistence-focused systemd service was created.
- 3) **Exfiltration (T1048 - Exfiltration Over Other Channels):** Data stealing over other channels: scans using command-line utilities (`curl`, `nc`) to perform simulated data theft.

2) *Log Chain Design:* The 100+ log lines were mixed with **60% benign "noise"** (e.g. normal dhclient traffic and benign systemd start-ups) and **40% bad data**.

C. Evaluation Metrics

The model performance was assessed with a **Ground Truth** dataset where all malicious events were pre-classified.

TABLE I
EVALUATION METRICS

Metric	Definition
Detection Rate (Recall)	% of all real malicious events correctly identified.
False Positive Rate (FPR)	% of benign events mistakenly classified as malicious threat
MITRE Coverage	Percentage of three simulated MITRE techniques accurately

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Reconstruction of Cyber Attacks

Using the gpt-oss-120b model, we have reconstructed the attack timeline accurately and pulled discrete log entries into this coherent threat narrative across three phases:

Phase 1 - The Recognition of the Presence of Brute Forces The model caught over 30 failed SSH logins within a minute.

Log Evidence: Dec 5 10:00:01
server sshd[1501]: Failed password
for invalid user admin from
104.28.2.14.

Phase 2: Persistence and Privilege Escalation The model has correctly identified the production of hidden files and unauthorized services.

Log Evidence: Dec 5 10:05:01
server CRON[1600]: (root) CMD
(echo "/1 bash -c 'bash -i >&
/dev/tcp/10.1.1.50/9001 0>&1' " >>
/var/spool/cron/root)
LLM Analysis: Found "Creation of hidden binaries,
cron-backdoors, systemd service... Persistence &
privilege escalation."

Phase 3: C2 (Command and Control) The model found outbound malicious IPs 185.10.10.10 and 13.37.13.37.

LLM Analysis: "Mass network connections...
listeners on ports 1337, 4444... firewall disabling."

B. Model Performance and Metrics

A detailed quantitative analysis of this model response led to **perfect detection scores** on this data set.

TABLE II
QUANTITATIVE PERFORMANCE MEASURES

Metrics	Definition	Result
Detection Rate (Recall)	% of malicious events detected	100%
False Positive Rate (FPR)	% of benign events considered to be attacks	0%
MITRE Coverage	Mapping of techniques to ATT&CK framework	100%

The model efficiently classifies benign noise such as Dec 5 10:20:03 server dhclient[1903]: DHCPREQUEST and it achieved a zero false-positive rate (as visible in Table II).

C. Action and Analysis of Remediation Actions Generated

The model developed an entire remediation script that indicates the potential of "Agentic Action."

LLM Generated Action Plan Snippet:

```
# Block brute-force source
iptables -A INPUT -s 104.28.2.14 -j DROP

# Kill back-door processes you know how to kill
pskill -f "bash -i >& /dev/tcp/10.1.1.50/9001"
pskill -f "nc -l -p 1337"

# Remove malicious cron entries
sed -i '/evil_script.sh/d' /var/spool/cron/root
```

Evaluation of Actions:

- **Syntax Accuracy:** 100 percent. The commands (iptables, pskill, sed) are all valid in Linux code terms.
- **Completeness with Respecting Remediation:** 100 percent. This system included Network layer (blocking IPs), Process layer (killing of back doors), and File System layer (taking down cron jobs).

D. Discussion

The findings support the hypothesis that LLMs can be effective threat detectors.

- **Noise Reduction and Contextual Awareness:** There are regular systemd startups that could go untapped and rogue services that could be flagged in your system, showing an understanding of system context beyond keyword matching.
- **Aggregation:** The model aggregated over 30 SSH logs until it got a single key finding ("Brute Force Password Guessing") which helps mitigate alert fatigue for analysts.
- **Immediate Containment:** Generating command types for an immediate response allows for rapid containment and mitigation, potentially preventing data exfiltration in real-world environments.

VI. CONCLUSION

In the cyber threat hunting domain, it is a powerful and transformative opportunity that Large Language Models can support with log-analysis. By analysing log files in this way, AI will provide a better approach to identify anomalous activity and develop actionable threats. Thus, our evaluation of the gpt-oss-120b model validates its ability to process dynamic, multistage log chains zero-shot, with perfect quantitative detection results and no false positives in the simulated situation. In addition to detection, the ability of the model to generate a full and accurate **Agentic Action** remediation script demonstrates its capability for immediate response and containment, which is critical. However, further work needs to consider optimizing the latency of the framework for full deployment in a real-time application and testing the generalization capability across multiple enterprise log formats.

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-Centri
- [12] M. R. Hawthorne, S. L. Plotkin, and B. Douglas, *Fingerprints: Analysis and Understanding the Science*, 2nd ed. Routledge, 2021.
- [13] M. R. Hawthorne, S. L. Plotkin, and B. Douglas, *Fingerprints: Analysis and Understanding the Science*, 2nd ed. Routledge, 2021.