LLM Network Intrusion Detection System

INSyT (Innovative Network Security Technologies)
Taeyang Kim, Bronze Frazer, Isaac Peterson, Damon Tingey

Sandia National Laboratories
Ryan Holt, Mike Reed

Abstract

Current network intrusion detection systems (NIDS) struggle to keep pace with the sophistication and evolving nature of cyberattacks. Traditional signature-based and rule-based systems are often brittle and easily bypassed, while anomaly-based systems suffer from high false positives. This leaves an alarming gap in network security, exposing organizations to data breaches, financial losses, and reputational damage.

We propose the development of a full-stack Large Language Model (LLM) product designed to revolutionize threat analysis through the comprehensive examination of system and network logs as natural language. This solution aims to provide systems that can understand complex attack protocols, detect novel attacks, and adapt to evolving threats. Our system enhances security with proactive and comprehensive defense against a wider range of threats, improves efficiency through accurate threat detection and reduced false positives, and offers a future-proofed defense with continuous adaptation to evolving threats, ensuring long-term effectiveness and protection against emerging attack vectors.

1 Introduction

Network intrusion is a critical issue in today's digital landscape. In 2023, the average cost per data breach in the United States was $9,480,000 [1]. Organizations face significant financial losses, operational disruptions, and reputational

damage due to sophisticated cyberattacks that traditional NIDS struggle to detect effectively.

Current NIDS rely heavily on signature-based and rule-based methodologies, which are often inflexible and unable to detect novel or sophisticated attacks [2]. Anomaly-based systems, while more adaptive, tend to generate high false-positive rates, burdening security teams with excessive alerts [3].

To address these challenges, we propose a novel approach: leveraging advanced LLMs to analyze system and network logs as natural language, enabling a deeper understanding of complex attack patterns and the detection of previously unseen threats.

2 Related Work

Machine learning has been applied to NIDS with varying degrees of success. Previous work has explored the use of neural networks for intrusion detection [4], but they often lack the ability to interpret the semantic content of log data effectively. Recent advancements in Natural Language Processing (NLP) and LLMs, such as BERT [5], have shown promise in interpreting and classifying text data, which can be applied to system logs.

Our approach distinguishes itself by treating log data as a form of pseudo-natural language, allowing the LLM to learn patterns and correlations that traditional models may miss.

3 Methodology

3.1 System Overview

Our proposed system is a full-stack framework capable of classifying network intrusions using LLMs. The key components include:

- **Data Ingestion**: Collecting and preprocessing system and network logs.
- **LLM-Based Analysis**: Utilizing a fine-tuned BERT model to analyze logs.
- **Classification Engine**: Categorizing detected anomalies into specific attack types.

- **User Interface**: A React-based client interface for monitoring and responding to threats.

3.2 Data Preparation

3.2.1 Dataset Description

We curated a dataset consisting of system and network logs collected from various sources, initially containing 48 different types of network attacks. The dataset was highly imbalanced, with a majority of data points falling within two classes.

3.2.2 Data Profiling and Label Consolidation

To streamline the modeling process, we conducted in-depth data profiling to consolidate the attack types into six broader categories:

1. **Privilege Escalation**: Includes escalated commands and user changes.
2. **Scan**: Encompasses DNS scans, network scans, and service scans.
3. **Data Exfiltration**: Covers methods like DNSteal and exfiltration services.
4. **Remote Command Execution**: Involves attacker-initiated HTTP and VPN connections.
5. **Webshell Upload**: Pertains to webshell commands and uploads.
6. **Password Cracking**: Includes various password cracking attempts.

This consolidation reduced complexity and improved the model's ability to generalize.

3.2.3 Exploratory Data Analysis (EDA)

Our EDA revealed:

- A clear distinction in log line lengths between benign and malicious entries (Figure 1).
- Longer log lines tend to be associated with a broader range of attack types.

- Tokenization of log lines presented challenges due to the pseudo-natural language nature of logs.

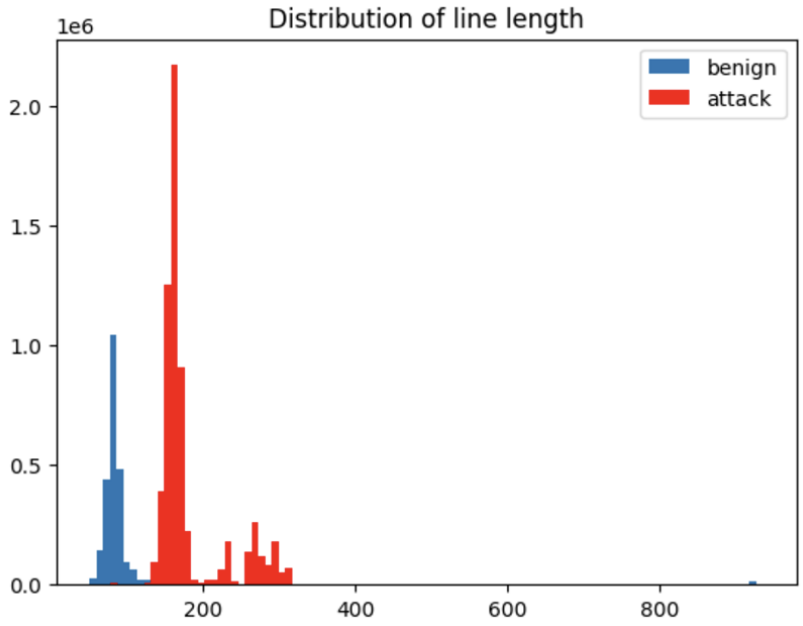

Figure 1: Distribution of Log Line Lengths by Classification

3.3 Model Development

3.3.1 BERT Tokenizer Customization

We utilized HuggingFace's implementation of DistilBERT [6] and customized the tokenizer to handle system log syntax. Recognizing that logs differ from natural language, we fine-tuned the tokenizer on our dataset to better capture meaningful patterns.

3.3.2 Model Training

The training process involved:

- Splitting the data into training, validation, and test sets.
- Employing oversampling and undersampling techniques to balance the dataset.

- Fine-tuning the BERT model using the customized tokenizer.
- Evaluating model performance using metrics such as accuracy, precision, recall, and F1-score.

3.3.3 Baseline Models

To establish a performance baseline, we trained simpler models, including logistic regression, random forest, and XGBoost classifiers, using engineered features like log line length and keyword presence.

4 Results

4.1 Model Performance

Our fine-tuned BERT model achieved the following results:

- **Accuracy**: 99.7%
- **Precision**: 99.6%
- **Recall**: 99.5%
- **F1-Score**: 99.6%

These results significantly outperformed the baseline models, which achieved approximately 85% accuracy on average.

4.2 Analysis

The high performance of the BERT model indicates its effectiveness in understanding and classifying complex log data. The model was able to learn intricate patterns and correlations within the logs that simpler models could not capture.

4.3 Challenges and Solutions

- **Data Imbalance**: Addressed through resampling techniques.
- **Tokenization of Log Data**: Solved by customizing the tokenizer and fine-tuning on our dataset.

- **Pseudolanguage Complexity**: Overcome by training BERT to recognize the syntax and semantics of log data.

5 Implementation

5.1 Full-Stack Framework

We developed a robust architecture comprising:

- **Backend**: Flask server managing API endpoints.
- **Database**: Redis for storing interim data and facilitating communication between components.
- **Frontend**: React-based interface for user interaction.
- **Model Integration**: Seamless connection between the frontend, backend, and the trained BERT model.

5.2 Deployment

The system was containerized using Docker and deployed on cloud platforms (e.g., AWS, Google Cloud) to ensure scalability and availability.

6 Discussion

6.1 Impact and Benefits

Our LLM-based NIDS offers:

- **Enhanced Security**: Proactive detection of a wider range of threats.
- **Improved Efficiency**: Reduced false positives, lowering operational costs.
- **Future-Proofing**: Adaptability to new and evolving attack vectors.

6.2 Cost Analysis

An initial investment of $10,000 was made for GPU resources. Compared to ongoing costs of leading NIDS contractors (up to $576 per day), our system provides a cost-effective alternative, recouping the initial investment in a relatively short time frame.

6.3 Limitations and Future Work

While our system shows promising results, challenges remain:

- **Continuous Learning**: Implementing mechanisms for ongoing model updates with new data.
- **Real-Time Processing**: Optimizing for low-latency detection in high-throughput environments.
- **Explainability**: Enhancing the model's ability to provide interpretable explanations for its detections.

7 Conclusion

We have developed a full-stack LLM network intrusion detection system that leverages advanced NLP techniques to analyze system and network logs as natural language. Our fine-tuned BERT model achieved 99.7% accuracy in classifying network intrusions into six major categories. The system offers significant improvements over traditional NIDS in terms of accuracy, adaptability, and cost-effectiveness.

This project demonstrates the potential of LLMs in cybersecurity applications and sets the groundwork for further advancements in intelligent threat detection systems.

Acknowledgments

We thank Sandia National Laboratories, particularly Ryan Holt and Mike Reed, for their support and collaboration. We also acknowledge the contributions of the INSyT team members: Bronze Frazer, Isaac Peterson, and Damon Tingey.

References

[1] IBM Security and Ponemon Institute. (2023). *Cost of a Data Breach Report 2023*. Retrieved from https://www.ibm.com/security/data-breach

[2] Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1-2), 18-28.

[3] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305-316).

[4] Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4), 1690-1700.

[5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT 2019* (pp. 4171–4186).

[6] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.